SIMULATION-BASED OPTIMIZATION OF A FOUR STAGE HYBRID FLOW SHOP WITH SEQUENCE-DEPENDENT SETUP TIMES AND AVAILABILITY CONSTRAINTS

Paul Aurich[a], Abdulrahman Nahhas[b], Tobias Reggelin[c], Marco Krist[d]

[a],[c] Department of Logistics and Material Handling Systems, Otto-von-Guericke University Magdeburg, Germany
[b] Magdeburg Research and Competence Cluster for Very Large Business Applications, Faculty of Computer Science, Otto von Guericke University Magdeburg, Germany
[d] Tectron GmbH, Germany

[a]paul.aurich@mail.de, [b]abdulrahman.nahhas@ovgu.de, [c]tobias.reggelin@ovgu.de
[d]marco.krist@thermofisher.com

ABSTRACT
In this paper, a deterministic four stage hybrid-flow-shop scheduling problem with sequence-dependent setup times of a printed circuit board assembly is discussed. Since the problem came from an industrial company, availability constraints have been taken into consideration. There are many reasons which can cause an unavailability of machines. In this article stochastic breakdowns and preventive maintenance were considered. Furthermore, deterministic breakdowns will be investigated, in order to compare deterministic and stochastic breakdowns in terms of the robustness and the stability of the solution and the required computational effort. The objective of the problem is to minimize the makespan, the total tardiness and the total setup time of the first stage. To generate an optimized production schedule, the metaheuristics simulated annealing, tabu search and differential evolution as well as a sequencing algorithm were combined with a discrete event simulation model. This paper is a continuation and extension of our previous work (Aurich et al. 2016; Nahhas et al. 2016).

Keywords: metaheuristic, simulation-based optimization, hybrid flow shop, breakdown, maintenance

1. INTRODUCTION & LITERATURE REVIEW
The development of digitization over the last decades and their connection with technical systems leads to the possibility to record the process data of a production system at any moment. This makes it possible to react faster to unforeseen events. In contrast to this, in most of the publications which were discussed in the field of scheduling, availability constraints were not taken into consideration. But real production systems are always subject to interruptions. Based on the data, which now can be detected, it is possible to define preventive maintenance times more precisely and notice breakdowns quickly. Therefore availability constraints should be taken into account.
Two types of processing cases are differed in the literature, when a machine becomes available again after an interruption. The first type according to (Lee 1996) is called *resumable*, in which the processing of a job can be continued without any loss in time. In opposite to that it is called *non-resumable*, if a restart of the processing from the beginning is necessary, which propagate a loss of time. (Saidy et al. 2008) surveyed a wide range of typical scheduling problems with the addition of resumable and non-resumable availability constraints.
Considering stochastic breakdowns, makes it difficult to create a mathematical model, because of the structural and functional complexity. However, simulation techniques are often used as an alternative solution. In addition, using simulation techniques the modeler is capable of mimicking the exact behavior of a considered system. According to (März et al. 2011), four possibilities of combining simulation with optimization techniques can be differentiated:

- Optimization is integrated into the simulation
- Simulation as evaluation function of optimization
- Simulation results as initial value of the optimization
- Optimization results for configuring the simulation

Since this article focusses on a hybrid flow shop (HFS) scheduling problem the literature review is concentrated on parallel machine (P), flow shop (F) and hybrid flow shop scheduling problems. A hybrid flow shop production environment consists of *k* stages in series. Each production stage comprises *j* parallel machines. Each job *i* should be processed in all stages and each job can be processed by any machine of a stage (Ruiz and Vázquez-Rodríguez 2010; Pinedo 2012).
The problem $F2|.|C_{max}$ is the only flow shop problem which can be solved in polynomial time. (Johnson 1954) developed the so called Johnson Rule to solve this problem. However, already the problem $F3|.|C_{max}$ is NP-hard in a strong sense (Garey and Johnson 1979). Also the problem $HFS2(P)|.|C_{max}$ is NP-hard in a strong sense, even if there is only one machine in the first stage und two parallel machines on the second stage, i.e. problem $HFS2(1, P2)|.|C_{max}$ studied by (Hoogeveen et al. 1996). Several other cases of the two stage HFS were

studied in the literature (Gupta 1988; Li 1997; Allaoui and Artiba 2006).

(Allaoui and Artiba 2004) considered different HFS with four stages, a maximum number of five machines at each stage and 50 jobs. Furthermore resumable and non-resumable availability constraints were taken into consideration. They investigated the impact of the initial schedule on simulated annealing with different objectives, such as minimization of the makespan, total completion time, mean flow time, mean waiting time, mean tardiness and maximum tardiness. To generate the initial solutions the SPT, LPT and EDD dispatching rules were used. (Gholami et al. 2009) developed a simulation-based optimization approach to solve several HFS problems with sequence-depend setup times and stochastic breakdowns. Resumable processing was considered. He adapted the random key genetic algorithm to build a schedule for the first production stage. To assign the jobs for the following stages the SPT cycling heuristic and a Johnson-Rule-based heuristic were implemented. The simulation is used as an evaluation function of the optimization. (Gholami et al. 2009) noticed that the first available machine rule would not be efficient, if sequence-dependent setup times have to be considered at all stages.

2. PROBLEM DESCRIPTION

The considered deterministic scheduling problem can be described as a four stage hybrid flow shop scheduling problem. In the classical hybrid flow shop all jobs have to be processed on one machine at each stage. In contrast to this the considered HFS is as special form, where all jobs have to be processed on the first and second stage, but only specific jobs have to be processed on the third or fourth stage or on each of them.

The machine environment consists of four stages (see Figure 1). The first stage contains four identical parallel surface mount device placement machines (SMD). These are the critical resources in the considered production system. The second stage accommodates five identical parallel automated optical inspection machines (AOI). The further stages contain only one machine each, a selective soldering machine (SS) in the third stage and a coating machine (CM) in the fourth stage. In the first and fourth stage jobs are scheduled with sequence-dependent major and minor setup times. (Tang 1990) and (Wittrock 1990) introduced the concept of major and minor setup times. They both investigated parallel machines scheduling problems, where several jobs could be grouped into different families depending on their part-types. In the first stage the setup time s_f depends on family-type f of a job. The jobs are cluster into families based on their raw materials. In the fourth stage the setup time depends on the coating-type c of job. The company uses two different coating-types. In the second and third stage jobs are scheduled with sequence-independent setup times.

The job nature and the considered restrictions can be described with the following assumptions:

- The number of jobs in a certain scheduling period is fixed.
- The processing time $p_{i,j,k}$ of each job i on machine j of stage k is known and fixed.
- The priority of a job represents it´s desired delivery date d_i.
- It is not allowed to process jobs from the same family-type on different machines of the first stage at the same time.
- The buffer size between two stages is unlimited.
- Preemption and splitting of jobs are not allowed.

Unlike most of the papers which deal with scheduling problems, this paper takes availability constraints into consideration. More specific preventive maintenance and stochastic breakdowns can affect a machine during or not during the processing of a job. When the machine becomes available after an interruption, the processing of a job continuous without any loss in time, consequently it is resumable rs.

The objective functions of the analyzed HFS are to minimize the makespan $C_{\max}$ (1), to minimize the total tardiness T (2) and to minimize the total setup time of the first stage $\sum s_f$.

$$C_{\max} = \max C_i \quad \forall \quad i = 1, \ ..., \ n \tag{1}$$

$$T = \sum_{i=1}^{n} T_i \ , \ T_i = \max(C_i - d_i, \ 0) \tag{2}$$

According to the classification of (Graham et al. 1979) the problem can be described with:

HSF4 (IP4, IP5, 1, 1)|s_f, s_c, rs|$C_{\max}$, T, $\sum s_f$.

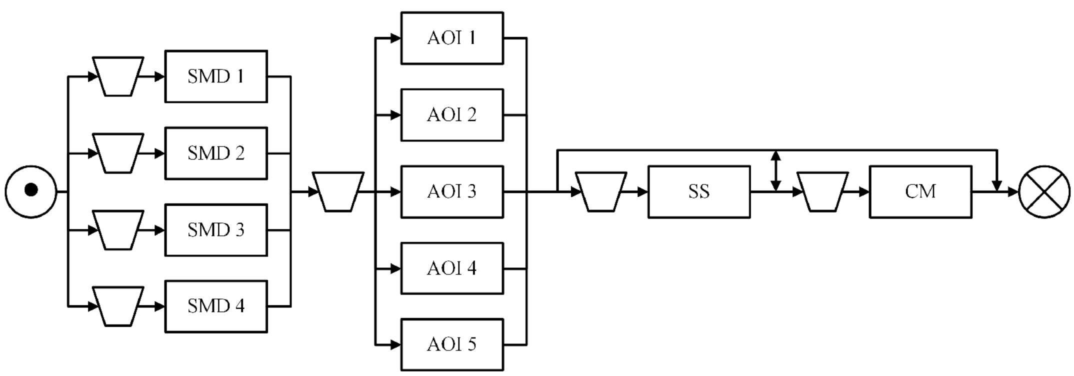

Figure 1: Production System

3. SOLUTION APPROACHES

Since the considered problem is NP-hard, it is not possible to develop a polynomial algorithm, which can provide an optimal solution in a reasonable time. In order to reduce the complexity of the problem the allocation and sequencing decisions of the first stage are separated from each other. On the following stages the jobs are assigned to the machines according to the first available machine (FAM) rule.

To solve the allocation problem of the first stage the metaheuristics simulated annealing, tabu search and differential evolution were implemented. To deal with the four independent single machine problems with sequence-dependent setup times in the first stage a sequencing algorithm was developed. The job sequence for the stages two to four is generated based on the earliest due date rule. The implementation of all metaheuristics, dispatching rules and the sequencing algorithm is done inside the simulation model in order to avoid an increasing computational time because of the data exchange between optimization and simulation tool. The simulation model was implemented in ExtendSim 9.1. The discrete event simulation approach is adapted to build the simulation model, in which each job is aggregated into a single object The combination of optimization and simulation model cannot be classified based on the classification of (März et al. 2011). This is because two methods of combination were used. On one side the optimization is integrated inside the simulation. On the other side the simulation is an evaluation function for the metaheuristics.

3.1. Initialization

Before the optimization takes place the user must insert a dataset into an excel document. Moreover, the user must decide which metaheuristic should be used and accordingly the control parameter must be setup. Then an initial allocation of families is executed in excel. This can be done randomly or with some easy sorting rules, for instance sorting the families based on the number of jobs with the same family-type or the total processing time of a family on the first stage. Hereafter all informations were send to the simulation model and a single simulation run is executed to measure the objective values of the initial allocation.

3.2. Functionality of the metaheuristics

Depending on the control parameters of the used metaheuristic a multi run simulation is setup. The execution of the metaheuristic takes place at the end of each simulation run. Based on the decision strategy of the metaheuristic it is decide if the current solution is used for the next iteration.

3.2.1. Simulated Annealing

Simulated Annealing (SA) is a nature inspired optimization technique, mimicking a thermodynamically cooling process. It was first introduced by (Kirkpatrick et al. 1983) and (Černý 1985) to solve the traveling salesman problem. The neighborhood search (NHS) of the adapted SA is a random single point operator NHS. This means that a randomly chosen family is randomly allocated to a new machine at the first stage. The decision strategy of the SA is divided into the following cases:

1. The new schedule dominates the old one in all objective values. The new solution is accepted and used as the next start solution.
2. The old schedule dominates the new one.
3. Neither the old schedule nor the new one dominates the other.

For cases two and three, the Boltzmann distribution is used to decide whether to accept a new solution or not. A weighted sum of the observed objective values was used since the Boltzmann distribution contains only one value.

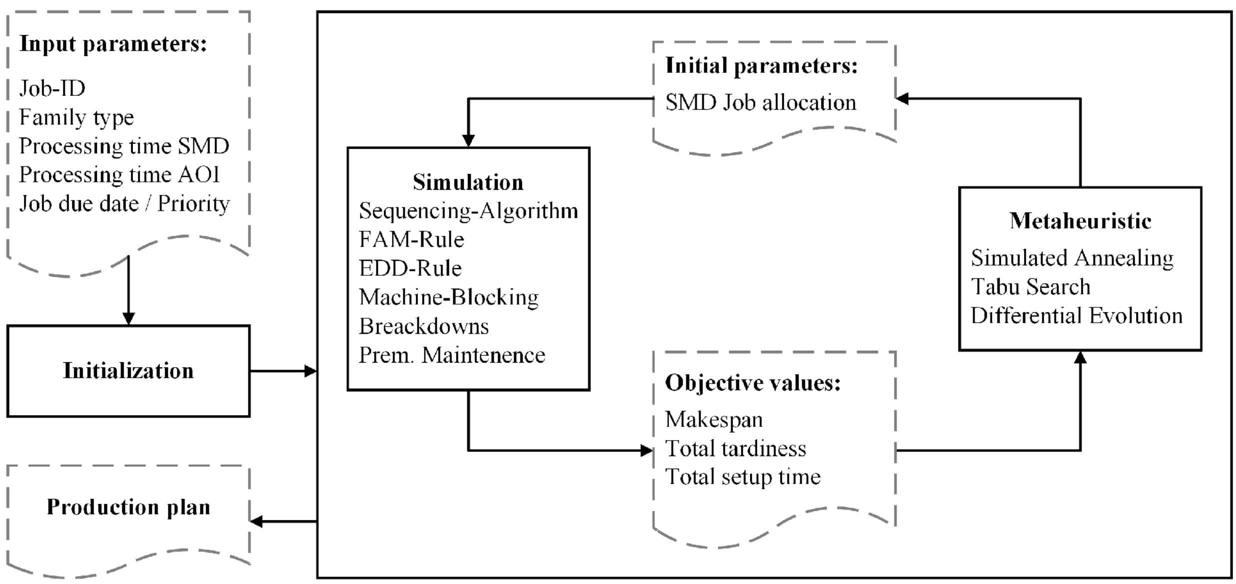

Figure 2: Metaheuristic Simulation-Based Optimization

3.2.2. Tabu Search

Tabu Search (TS) is a deterministic local search technique guided by a fixed or adaptive memory structure. It was developed by (Glover 1977), in order to solve combinatorial optimization problems. As decision strategy for the adapted TS the best neighbor strategy is used. The NHS is again a single point operator neighborhood search. This means that in each iteration each family is once allocated to each machine on the first stage. When the neighborhood search is finished the neighbors are compared to each other in order to find the best neighbor. If no neighbor exists that dominates all other neighbors in all objective values, the weighted sum is used to identify the best solution.

3.2.3. Differential Evolution

Differential Evolution is a stochastic population based optimization technique, introduced by (Storn and Price 1997). In contrast to other metaheuristics it is comparatively new.

For the NHS the user can decide between DE/best/1 and DE/rand/1. DE/best/1 means that the new individual is generated based on the best individual from the last generation and two randomly chosen ones. DE/rand/1 means that the individual is generated depending on three randomly chosen individuals from the previous generation. The selection strategy for the DE is a greedy selection between the current individual and its predecessor. If the current solution dominates the previous one or if the weighted sum outperforms the other, the new individual is chosen, else the predecessor is selected for the next generation.

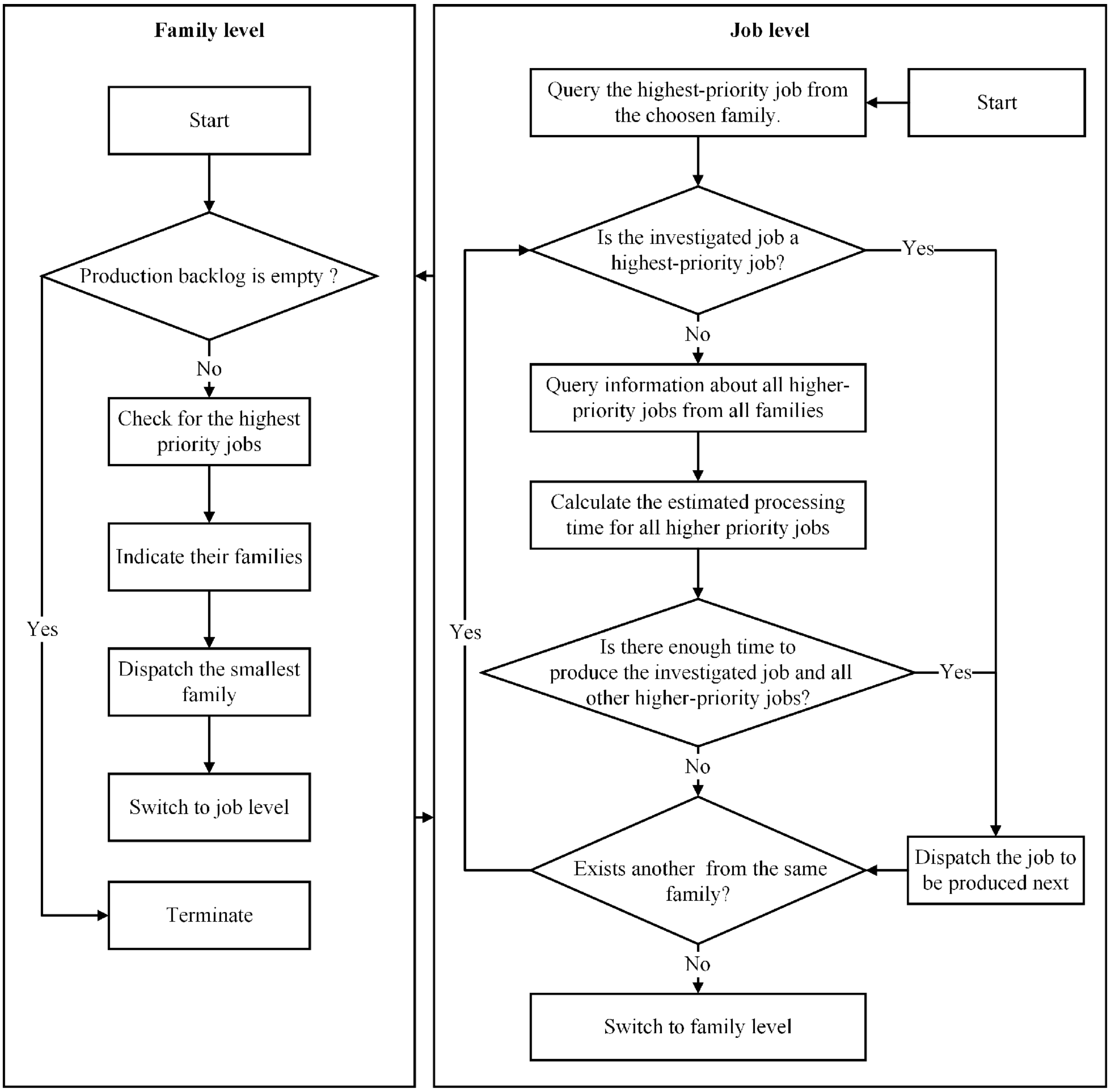

Figure 3: Sequencing Algorithm

3.3. Sequencing algorithm

The sequencing algorithm was developed in order to minimize the objective functions and furthermore to improve the machine utilization of the first stage and was derived from the one formally presented in (Aurich et al. 2016; Nahhas et al. 2016). To meet these conditions all jobs of a family should be processed successively to avoid major setups. But a strict successive processing of jobs from the same family would lead to delivery time violations of many jobs from other families. The algorithm has been developed to resolve this tension.

The behavior of the sequencing algorithm can be divided in two logical levels; a family level and a job level (see Figure 3). When the sequencing algorithm is initiated, it first executes the family level. Here, the smallest family which contains at least one of the highest priority jobs is chosen. The smallest family is the one with the least total processing time of jobs. The reason to choose the smallest family is a chance to completely produce all jobs of it before reaching a critical point. A critical point is met, when it is no longer possible to produce a job from the same family without violating the delivery date of jobs from other families. When the family level of the sequencing algorithm has chosen a family the job level is executed. On the job level the algorithm chooses jobs from the same family according to their priority, using the EDD rule. The sequencing algorithm keeps operating in the job level until all jobs of the family are produced or a critical point is met.

3.4. Machine Blocking

The machine blocking is an extension of the methodologies which were discussed in sections before. The goal of this extension is to avoid a delay of jobs. It is implemented on the stages two, three and four. When a job leaves stage *k*, it is checked if a job is processed in stage *k-1*, which will be delayed, when it has to wait in stage *k*. If this case occurs the free machine of stage *k* is blocked for the job. The blocking only takes place when all machines of a stage are full except one machine and all jobs in the queue of stage *k* are not critical. Critical means that a job must be processed as fast as possible to meet its due date. More precisely the maximum blocking time is up to 120 minutes and the job a machine is blocked for, has to be critical.

4. DESIGN OF EXPERIMENTS

The datasets which were used for the experiments are real records from the company's production pool. Each dataset represents a production schedule of three weeks. The characteristics of those datasets are shown in Table 1. In general the dataset contain a relatively large amount of jobs, with heterogeneous processing times.

In order to compare the impact of the deterministic and the stochastic breakdowns, several runs with all datasets and solution methodologies were executed. A deterministic breakdown happens at the end of a day for 135 minutes. This represents a 45 minute breakdown in each of the three shifts of a day. The behavior of the stochastic breakdowns can be described with the time between failure (TBF) and time to repair (TTR) scheme. The TBF is based on a normal distribution with a mean of 480 minutes and a standard deviation of 120 minutes. For the TTR a mean of 45 minutes and a standard deviation of 15 minutes were used. That is about 135 minutes per day, which is similar to the deterministic behavior. Thus, the two breakdown variants are approximately comparable. The use of the normal distribution follows from an analysis of the machine breakdowns. In the stochastic case the objectives of each schedule are the mean values of teen simulation-runs. To check the quality of an optimized schedule, 200 stochastic simulation runs were executed.

For the SA the following ranges of control parameter values were used: temperature $T \in \{15,30\}$, linear cooling rate $\alpha \in \{0.1\}$ and step size $n \in \{10,25\}$. The low α and the relative high n lead to a slow cooling and thus prevent a large amount of entropy.

The control parameters of the TS were set with the following ranges: tabu list length $TL \in \{5,20\}$ and number of iterations $i \in \{30,75\}$.

The experiments with the DE were executed with the following ranges of parameter values: number of generations $G \in \{25,100\}$, population size $NP \in \{20,50\}$, crossover rate $CR \in \{0.05\}$, mutation factor $F \in \{0.3\}$.

Table 1: Input Datasets

	Dataset 1	Dataset 2	Dataset 3	Dataset 4
Number of jobs	181	179	194	170
Number of families	21	23	24	29
SMD: process time interval | mean	7 - 2912 | 306	6 - 1376 | 366	3 - 1151 | 228	3 - 1957 | 326
AOI: process time interval | mean	9 - 3747 | 381	8 - 1605 | 456	4 - 1495 | 286	4 - 2173 | 403
SS: process time interval | mean	50 - 3250 | 845	133 - 2808 | 1083	11 - 3641 | 648	79 - 2804 | 1133
SS: proportion of jobs	15%	11%	18%	12%
C: process time interval | mean	137 - 8973 | 1322	67 - 2870 | 1044	70 -2302 | 786	70 -2302 | 1076
C: proportion of jobs	9%	9%	13%	11%

5. COMPUTATIONAL RESULTS

Table 1 and Figure 4 show the computational results of the applied approaches. One of the research questions was, whether metaheuristics are able to optimize the job allocation, while the sequencing is done by a special algorithm and which metaheuristic performs best. Overall the computational results of the applied metaheuristics are very similar. The formulation of a statement, which metaheuristic outperforms another in terms of the objective functions, is very difficult for this scheduling problem and these datasets.

Table 2: Computational Results of the different solution approaches

		Makespan (minutes)	Total Tardiness (minutes)	Penalties (number)	Major-Setups (number)	Computational Time (hours)
Dataset 1						
deterministic Breakdowns	SA	29618	352	0,432	18,395	0,73
	TS	29561	250	0,267	19,86	0,45
	DE	29946	291	0,384	18	0,28
stochastic Breakdowns	SA	29921	323	0,458	19,075	9,03
	TS	29703	341	0,465	20	8,53
	DE	29744	404	0,53	23,21	9,03
Dataset 2						
deterministic Breakdowns	SA	27851	0	0,005	24,635	0,54
	TS	28174	2	0,05	22,56	0,65
	DE	27894	68	0,63	24,475	0,73
stochastic Breakdowns	SA	27913	3	0,025	24,015	5,42
	TS	28235	0	0,005	22,94	9,75
	DE	27949	1	0,01	24,1	9,03
Dataset 3						
deterministic Breakdowns	SA	27983	3	0,055	27,755	0,68
	TS	28462	64	0,665	25	0,98
	DE	27886	6	0,01	25	0,73
stochastic Breakdowns	SA	28148	1	0,06	25,14	13,55
	TS	27483	27	0,22	25,25	9,75
	DE	28385	3	0,065	25	9,03
Dataset 4						
deterministic Breakdowns	SA	28543	10	0,325	31,93	0,73
	TS	28464	4	0,135	32	1,18
	DE	28952	7	0,255	30,285	0,68
stochastic Breakdowns	SA	28775	1	0,045	31	6,78
	TS	28821	2	0,08	31	11,78
	DE	28534	1	0,085	34	7,22

The TS slightly outperforms the other approaches in the deterministic case in terms of total tardiness and makespan. When stochastic breakdowns took place, no solution approach dominates over all datasets. In summary, all methods are suitable for solving the considered problem. But, for example, an increasing number families would raise the complexity of the problem, especially the computational time of the TS would increase rapidly, because of the neighborhood search structure, cf. (Nahhas et al. 2016). A short tabu list leads to better results for the TS, this also caused by the small number of families. Furthermore another strategy for the generation of the start population of the DE should be tested. The current random generation seems to be improvable.
The results for the makespan are very similar over all approaches, regardless of whether deterministic or stochastic breakdowns are considered. The evaluation of the provided Gantt chart shows that third and fourth production stage are highly loaded. Therefore they must be regarded as a kind of bottleneck, consequently the stages three and four have a main impact on the makespan. The sooner a job first arrives at one of these stages, rather at stage four, the smaller is the makespan. The first and second stages are often in an idle state, while the last jobs are produced at stage three or stage four. For this reason the company should maybe rethink the selection of objective functions. For example the total completion time could be a good substitute. On the other hand is the considered environment a dynamic system and a period of three weeks will be optimized. Since an optimization run will take place at the end of each week, more jobs could be added, which doesn´t need to be processed on stage three or four.
Another research questions was if it is necessary to consider stochastic breakdowns or whether deterministic breakdowns also lead to stable production schedules. The experiments of both procedures showed very similar results in terms of makespan. In regard to total tardiness and total major setup times it seems that the deterministic procedure tends to better results. The computational time increases tenfold at stochastic procedure. It could be shown that the considering of deterministic breakdowns led to stable results, while the computational effort rapidly decreases. This procedure suits very well for the consideration of machine breakdowns.

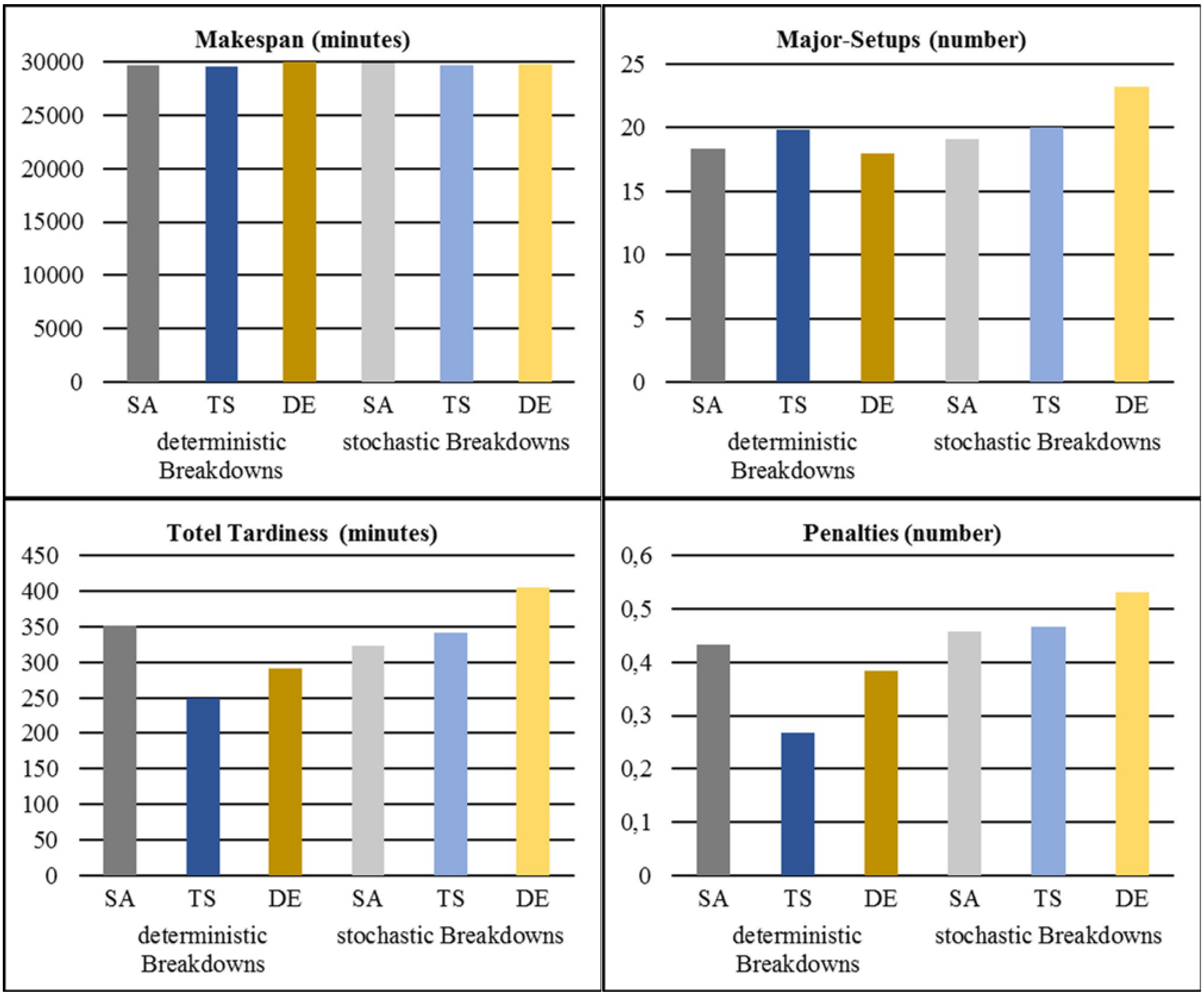

Figure 4: Computational Results of the different solution approaches for the first dataset

6. CONCLUSION

In this paper, a four stage hybrid flow shop scheduling problem was solved with a simulation-based optimization approach. Sequence-dependent setup times, stochastic and deterministic machine breakdowns and preemptive maintenance were considered. The allocation problem of the first production stage was optimized with the metaheuristics simulated annealing, tabu search and differential evolution. The sequencing algorithm proposed by (Nahhas et al. 2016) was implemented to solve the sequencing problem of the machines on the first stage. For the following stages the first available machine rule was used for the allocation and the earliest due date rule was applied for the sequencing. The results of several experiments showed, that a deterministic representation of breakdowns suffices to reach stable production schedules.

REFERENCES

Allaoui H., Artiba A., 2004. Integrating simulation and optimization to schedule a hybrid flow shop with maintenance constraints. Computers & Industrial Engineering, 47 (4), 431–450.

Allaoui H., Artiba A., 2006. Scheduling two-stage hybrid flow shop with availability constraints. Computers & Operations Research, 33 (5), 1399–1419.

Aurich P., Nahhas A., Reggelin T.; Tolujew, Juri, 2016. Simulation-based optimization for solving a hybrid flow shop scheduling problem: Proceedings of the 2016 Winter Simulation Conference. Winter Simulation Conference 2016, 2809–2819, 11.-14. December. Arlington, VA.

Černý V., 1985. Thermodynamical approach to the traveling salesman problem. Journal of Optimization Theory and Applications, 45 (1), 41–51.

Garey M. R. and Johnson D. S., 1979. Computers and intractability. A guide to the theory of NP-completeness. New York: Freeman.

Gholami M., Zandieh M., Alem-Tabriz A., 2009. Scheduling hybrid flow shop with sequence-dependent setup times and machines with random breakdowns. The International Journal of Advanced Manufacturing Technology, 42 (1-2), 189–201.

Glover F., 1977. Heuristics for integer programming using surrogate constraints. Decision sciences, 8 (1), 156–166.

Graham, R. L.; Lawler, E. L.; Lenstra, J. K.; Kan, A.H.G.Rinnooy, 1979. Optimization and Approximation in Deterministic Sequencing and Scheduling: a Survey. In Hammer P. L., Johnson E. L., Korte B. H. (Eds.): Discrete Optimization II, Proceedings of the Advanced Research Institute on Discrete Optimization and Systems Applications of the Systems Science Panel of NATO and of the Discrete Optimization Symposium co-sponsored by IBM Canada and SIAM Banff, Aha. and Vancouver (Annals of Discrete Mathematics, 5), vol. 5. North Holland: Elsevier, 287–326.

Gupta J. N. D., 1988. Two-Stage, Hybrid Flowshop Scheduling Problem. Journal of the Operational Research Society, 39 (4), 359–364.

Hoogeveen J. A., Lenstra J. K., Veltman B., 1996. Preemptive scheduling in a two-stage multiprocessor flow shop is NP-hard. European Journal of Operational Research, 89 (1), 172–175.

Johnson S. M., 1954. Optimal two- and three-stage production schedules with setup times included. Naval Research Logistics, 1 (1), 61–68.

Kirkpatrick S., Gelatt C. D., JR, Vecchi M. P., 1983. Optimization by simulated annealing. Science (New York, N.Y.), 220 (4598), 671–680.

Lee C.-Y., 1996. Machine scheduling with an availability constraint. Journal of Global Optimization, 9 (3-4), 395–416.

Li S., 1997. A hybrid two-stage flowshop with part family, batch production, major and minor set-ups. European Journal of Operational Research, 102 (1), 142–156.

März L., Krug W., Rose O., Weigert G., 2011. Simulation und Optimierung in Produktion und Logistik. Praxisorientierter Leitfaden mit Fallbeispielen. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg.

Nahhas A., Aurich P., Reggelin T.; Tolujew, Juri, 2016. Heuristic and metaheuristic simulation-based optimization for solving a hybrid flow shop scheduling problem: Proceedings of the International Conference on Modeling and Applied Simulation 2016. The 15th International Conference on Modeling and applied Simulation, 95–103, 26.-28. September. Cyprus.

Pinedo M. L., 2012. Scheduling. Theory, Algorithms, and Systems. Boston, MA: Springer US.

Ruiz R., Vázquez-Rodríguez J. A., 2010. The hybrid flow shop scheduling problem. European Journal of Operational Research, 205 (1), 1–18.

Saidy D., Reza H., Taghi Taghavi-Fard M., 2008. Study of scheduling problems with machine availability constraint. Journal of Industrial and Systems Engineering, 1 (4), 360–383.

Storn R., Price K., 1997. Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. Journal of Global Optimization, 11 (4), 341–359.

Tang C. S., 1990. Scheduling batches on parallel machines with major and minor set-ups. European Journal of Operational Research, 46 (1), 28–37.

Wittrock R., 1990. Scheduling parallel machines with major and minor setup times. International Journal of Flexible Manufacturing Systems, 2 (4).

AUTHORS BIOGRAPHY

PAUL AURICH is a master student of Industrial Engineering with specialization in production systems at the Otto von Guericke University Magdeburg. His main work interests are scheduling problems and simulation based optimization solution strategies as well as automated modelling. Paul Aurich holds a bachelor degree in Industrial Engineering with specializing in logistics from Otto von Guericke University Magdeburg. His email address is paul.aurich@mail.de.

ABDULRAHMAN NAHHAS is a PhD-student of Business Informatics at the Otto von Guericke University Magdeburg. His main work interests are scheduling problems and simulation based optimization solution strategies. Abdulrahman Nahhas holds a master degree in Business Informatics from Otto von Guericke University Magdeburg. His email address is abdulrahman.nahhas@hotmail.com.

TOBIAS REGGELIN is a project manager at the Fraunhofer Institute for Factory Operation and Automation IFF and Otto von Guericke University Magdeburg. His main research and work interests are logistics system modeling and simulation and the development and conduction of logistics management games. Tobias Reggelin received a doctoral degree in engineering from Otto von University Magdeburg. Furthermore, he holds a diploma degree in industrial engineering in logistics from Otto von Guericke University Magdeburg and a master’s degree in Engineering Management from Rose-Hulman Institute of Technology in Terre Haute, IN. His email address is tobias.reggelin@ovgu.de.

MARCO KRIST is a logistic manager of the company Thermo Fisher Scientific. His main work interests are optimization of production systems and supply chain planning. Marco Krist holds a diploma degree in industrial engineering in logistics from Otto von Guericke University Magdeburg. His email address is marco.krist@thermofisher.com.