METAHEURISTIC AND HYBRID SIMULATION-BASED OPTIMIZATION FOR SOLVING SCHEDULING PROBLEMS WITH MAJOR AND MINOR SETUP TIMES

A. Nahhas[(a)], P. Aurich [(b)], S. Bosse [(c)], T. Reggelin [(d)], K. Turowski [(e)]

[(a), (c), (e)] Magdeburg Research and Competence Cluster for Very Large Business Applications, Faculty of computer science, Otto von Guericke University Magdeburg, Germany
[(b), (d)] Department of Logistics and Material Handling Systems, Otto von Guericke University Magdeburg, Germany

[(a)] abdulrahman.nahhas@ovgu.de, [(b)] paul.aurich@mail.de, [(c)] sascha.bosse@ovgu.de
[(d)] tobias.reggelin@ovgu.de, [(e)] klaus.turowski@ovgu.de

ABSTRACT
This work has been motivated by an industrial case study in the field of printed circuit board's assembly production. Two- and four-stage Hybrid Flow Shop (HFS) scheduling problems with family major and minor sequence-dependent setup times are investigated. The majority of HFS scheduling problems are NP-hard optimization problems. Therefore, in this work, a metaheuristic and two hybrid simulation based optimization approaches will be presented to solve the problems and present a decision-making support tool for setting scheduling policies. Hybrid solution approaches that combine Genetic Algorithms (GA) with a heuristic are presented to solve the problems and compared to the GA. The optimization approaches are integrated into a discrete-event simulation model, which contributes as well as evaluates the quality of the obtained solutions. The formulated optimization problems are based on multi-objective measures to take into consideration the optimization of the system utilization through minimizing the makespan and the total number of major setup times as well as the customer satisfaction through minimizing the total tardiness. The presented solution techniques are evaluated based on real data, which are supported by the enterprise.

Keywords: Simulation-based Optimization, Hybrid Flow Shop Scheduling problem, Genetic Algorithms, Meta-heuristics

1. INTRODUCTION

Scheduling problems have been intensively investigated in the last four decades in different fields of academia due to their essential role in different manufacturing and service sectors as well as the challenging and complexity nature of their formulation (Ruiz and Vázquez-Rodríguez, 2010). In spite of the operative nature of scheduling tasks, they have a critical impact on most of the strategic decision making processes in an enterprise (Pinedo, 2012). This work has been motivated by an industrial case study in the field of printed circuit board's assembly production. The investigated problems are classified under Hybrid Flow Shop (HFS) scheduling problems. The HFS scheduling problems constitute a major class of scheduling problems, which is recently heavily addressed since the majority of assembly industrial production system are classified under HFS production systems (Ribas et al., 2010; Ruiz and Vázquez-Rodríguez, 2010). An HFS production environment consists of k production stages in series. Each production stage comprises m identical parallel machines. Each job j has to be processed on each production stage on one of the identical machines (Pinedo, 2012).

The majority of HFS scheduling problems are NP-hard optimization problems (Lenstra et al., 1977). Therefore, in this work, a metaheuristic and a hybrid simulation based optimization approaches will be presented to solve the problems and support decision-making processes regarding setting scheduling policies in the investigated system. Hybrid solution approaches that combine GA with a heuristic are presented to solve the problems and compared to the GA. The majority of the previous contributions in the field of scheduling problems strive to optimize the makespan (C_{max}), while very few target problems with multi-objectives function (Ribas et al., 2010). Therefore, the formulated optimization problems are based on multi-objective measures to take into consideration the optimization of the system utilization through minimizing the makespan and the total number of major setup times as well as the customer satisfaction through minimizing the total tardiness.

The presented solution techniques are evaluated based on real data, which are supported by the enterprise. The solution approaches are integrated into a simulation model to deliver a decision support tool for setting scheduling policies. This research aims to investigate the performance of the hybrid solution approach against Genetic Algorithms (GA) to solve the two-stage problem and then analyze the impact of expanding the complexity to solve the four-stage problem in terms of optimizing the objective measures and the required computational effort to obtain the solution. In the course of this paper, a literature review is presented in the second section to outline the relevance of the investigated problem and the often adapted solution methodologies. The definition of the problems and the description of the investigated system are demonstrated in the third section. The fourth section comprises the presented hybrid approaches and implementation of the GA. Followed, in the fifth section, the computational results of the evaluation are demonstrated. Finally, the paper is closed with conclusions and further investigation scopes.

2. SYSTEM DESCRIPTION AND PROBLEM FORMULATION

According to Graham et al. (1979), scheduling problems are formally described based on three fields problem description α | β | γ. In the first field α, the machine environment and configuration is illustrated. The job characteristics and the restrictions are presented in the second field β. The objective functions are then described in the third field γ.

2.1. Machine environment and configuration

The investigated problems are derived from the same production system. The four-stage problem is an expansion of the two-stage one. The system is classified as a Hybrid Flow Shop (HFS) production system. The first production stage contains four identical Surface Mounting Devices (SMD) machines. The second stage comprises five identical Automated Optical Inspection (AOI) machines. All jobs must be processed on the first and second production stages. Some jobs undergo the third and/or the fourth processing stages. The third stage contains a single Selective Soldering (SS) machine. Similarly, on the fourth stage, a Conformal Coating (CC) machine is available.

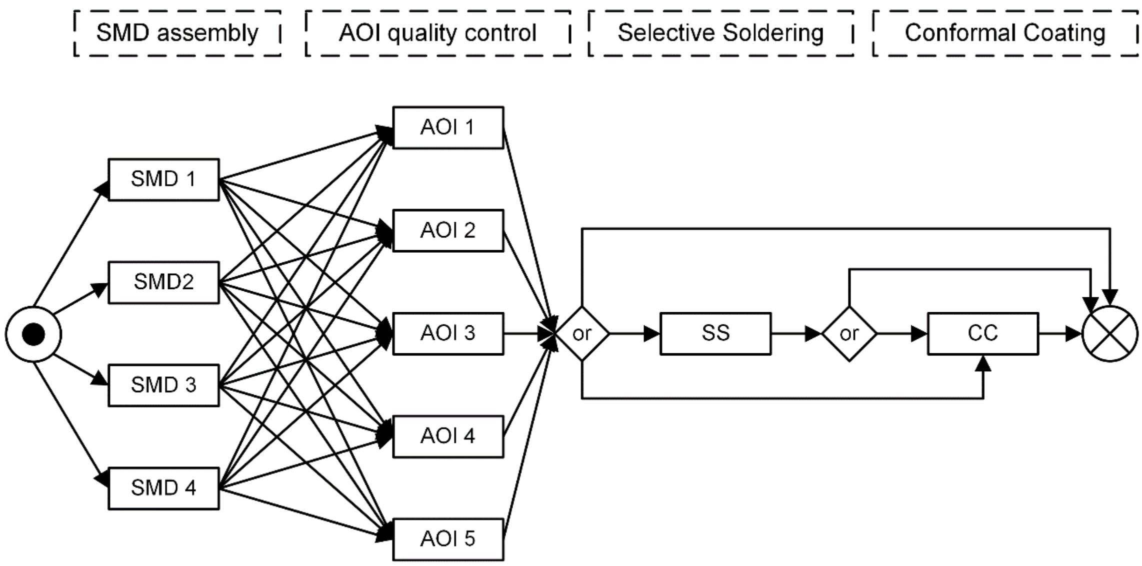

Figure 1: The investigated production system.

2.2. Job characteristics

Jobs of the analyzed HFS scheduling problems can be characterized as follows:

- The number of jobs in a certain time period and the number of products per job are known and fix.
- Part types are very heterogeneous.
- The family type of a job depends on the used raw materials.
- The processing time $p_{j,m,s}$ of each job *j* on the machine *m* of stage *s* is known and fix.
- The priority of a job represents the delivery date to the customer.
- The major setup time $s_{g,h}$ is the required time to configure a machine that was processing jobs from family *g* to process jobs from family *h*.
- Machine breakdowns are aggregated and subtracted from the production capacity.
- Buffer size between production stages is unlimited.

In the first production stage (SMD), jobs are scheduled with family major and minor setup times on the machines. In the second production stage (AOI), jobs are scheduled incurring sequence-independent setup times on the machines. The concept of major and minor setup time was introduced by Wittrock (1990) as well as by Kut C. So, (1990) to describe sequence-dependency. Jobs which share common raw materials are grouped into families. A minor setup time is inquired if the machine switches from one part type to another inside the same family. On the other hand, a major setup time is encountered, if the machine switches from one part type to another from a different family.

2.3. Objective functions

Accomplishing a balance between production system efficiency and the job's due date is a trade-off decision. For this reason, tardiness has been frequently used as a major supplementary performance criterion along with the makespan (Lenstra et al., 1977). The objective functions of the analyzed HFS problems are to minimize the makespan $C_{\max}$, the total tardiness T and the total number of major setup times to process all jobs as demonstrated in. The makespan is the necessary time to complete all released jobs (Wittrock, 1990) as demonstrated in $C_{\max} = \max C_i \ \forall \ i = 1, \ ..., \ n$

(**1**. To minimize $C_{\max}$ it is important to minimize the number of major setups. Tardiness is the difference between the completion time of a job C_j and its due date d_j as shown in $T = \sum_{i=1}^{n} T_i \ , \ T_i = \max(C_i - d_i, \ 0)$

(**2**.

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

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

3. LITERATURE REVIEW

Generally speaking, the solution methodologies that are often used to deal with scheduling problems can be classified according to their solution quality and method of implementation into two main groups. The first group contains analytical methods and exact optimization techniques, while the second group consists of heuristics and improvement approaches. The exact approaches usually guarantee optimal solutions or bounded optimal ones using some approximation scheme (Baker and Trietsch, 2009). Dynamic programming (Held and Karp, 1961) and branch and bound (Kis and Pesch, 2005) are typical exact optimization approaches, which are often employed to deal with scheduling problems. Those approaches are usually adapted to solve small- and moderate-size scheduling problem. One of the prior works in the flow shop is scheduling is the contribution of Johnson (1954). The author investigated a two-stage Flow shop (F) scheduling problem $F(1,1) \,||\, C_{\max}$ and presented an exact algorithm to minimize the makespan in polynomial time.

Kis and Pesch (2005) presented a comprehensive review of the adapted exact approaches for solving HFS scheduling problems, in which the branch and bound optimization technique was the dominant exact solution approach. Although exact approaches grantee optimal solution, they are computationally expensive when the problem domain gets more complex. In addition, the conducted implementations of exact approaches in the field of operation research maintain often a scientific nature of problem formulation, which usually involves many simplifications and rough assumptions to reduce the complexity of a considered problem. This fact propagated a gap between the research conducted on scheduling theory and scheduling activities in practice (Maccarthy and Liu, 1993, p. 59).

In the industry, scheduling policies are often managed based on some intuitive rules and procedures. Academia classified those procedures under the so-called Priority Dispatching Rules (PDRs). They constitute the simplest form of heuristic procedures. These rules are widely used in practice to manage production planning. By definition, PDRs are some simple constructive procedures used to prioritize a set of released jobs for scheduling and production (Hunsucker and Shah, 1994). The Earliest-Due-Date (EDD) and the Shortest-Processing-Time (SPT) are typical examples of PDRs, which are often used to set a sequencing strategy to prioritize and measure the significant degree of jobs (Andersson et al., 2008). Hunsucker and Shah (1994) presented a profound analysis on PDRs and their performance. They analyzed six PDRs on different flow shop scheduling problems $\mathrm{HFS_m} \,||\, C_{\max}, F_{\max}, \bar{F}$ that are subject to the minimization of various objective functions such as the makespan, the mean flow time and the maximum flow time ($F_{\max}$).

However, when the quality of the obtained solution become more essential, PDRs are not anymore applicable. This statement can be explained through observing an inverse correlation between their solution quality and the complexity of an investigated problem. Therefore, PDRs have been recently investigated in combination with metaheuristic approaches as for instance genetic algorithms, in which a simulation study is conducted to evaluate the quality of the obtained solution as presented by Geiger et al. (2006). A Similar concept has been investigated by Andersson et al. (2008), who conducted a simulation study, in which genetic algorithms are employed to identify the appropriate combination of priority dispatching rules for solving scheduling problem. The encoding strategy of the metaheuristic in such implementation usually targets the problem indirectly, through passing the different rules to the optimization techniques instead of approaching the problem directly. More sophisticated methods have been also applied to construct PDRs using some machine learning mechanisms as for instance neural networks (Wang et al., 2005).

Improvement heuristics and metaheuristics can be anticipated as a middle ground between PDRs and exact methods in terms of the solution quality and the computational effort required to solve scheduling problems. Improvement heuristics are conceptually more sophisticated heuristic procedures than the constructive ones since the construction of a production schedule is the first step in the internal functionality of an improvement heuristic. Thereafter, based on the initial constructed schedule, an improvement heuristic seeks to conduct single or several changes on the schedule, which yield to a better investigated objective function (Pinedo, 2012, p. 382). A particular class of improvement procedures is local search algorithms. A local search algorithm functions in a similar manner to an improvement procedure, except that the modification procedures on an investigated solution must be well defined and identically conducted on all candidate solutions (Pirlot, 1996).

Local search algorithms attempt to find the optimum of an investigated region or 'neighborhood' of the solution space. Two solution candidates are neighbors if conducting a predefined change on a schedule generates the other neighbor schedule. The modification process is iteratively conducted on the investigated solutions until the optimum of that region in the solution space is found or a breaking criterion is met (Orlin et al., 2004, p. 588). The modification procedure in this sense defines the complexity of an improvement algorithm since it shapes the size of the investigated neighborhood in the solution space. Wittrock (1990) investigated a simplified form of the considered problems. He presented an improvement heuristic approach based on binary search tree to solve the identical parallel machines problem (*P_m*), which is a stage of a hybrid flow shop production system $\mathrm{P_m} \mid s_{g,h} \mid C_{\max}$. The problem was investigated under major and minor family sequence-dependent setup time ($s_{g,h}$), which is subject to minimize the makespan. The author proved that the complexity of the problem is NP-hard. The performance of the presented approach was finally compared against a lower bound on the problem, which was demonstrated by the author.

Gupta, (1988) addressed a sophisticated form of the identical parallel machine problem $\mathrm{HFS_2(P_m,1)} \parallel C_{\max}$, in which the second stage with a single machine has been taken into consideration to form an HFS problem. The problem is NP-hard in a strong sense. Gupta treated the problem by dealing with the sequencing part and the allocation part independently. He assumed that only two-stage flow shop scheduling problem has to be solved with the objective function to minimize the makespan $F(1,1) \parallel C_{\max}$. He adapted the algorithm, which was presented by Johnson (1954) to deal with the sequencing part of the problem since this algorithm solves the problem optimally. Independently, he presented a heuristic to deal with the allocation part of the problem on the first stage with the objective function to minimize the total idle time on the second stage. He reported near optimal results of the makespan with three to five percent deviation from his calculated lower bound in almost all generated problems.

Voß (1993) addressed a similar problem to the one investigated by Gupta, in which sequence-dependent setup times is considered $\mathrm{HFS_2(P_m,1)} \mid s_{j,k} \mid C_{\max}$. The author adopted the same solution strategy presented by Gupta to solve the problem. His contribution lied in integrating different improvement procedures to the solution strategy presented by Gupta and a new setup method. He also applied Tabu Search to obtain a local optimum from the solution space of the generated initial solution. The author reported improved results for all problem instances in comparison to the ones generated by Gupta. Li (1997) also treated a two-stage hybrid flow shop scheduling problem. However, his problem formulation was literally opposite in term of machines to the one addressed by Voß and Gupta, in which only one machine on the first stage and parallel identical machines on the second stage have been taken into consideration $\mathrm{HFS_2(1, P_m)} \mid s_{j,k} \mid C_{\max}$. The problem was investigated under family major and minor sequence-dependent setup time constraint using different heuristic methods.

Although improvement and among them local search algorithms point out good results, their conceptual design and functionality are based on searching only in the neighborhood of an initial solution to achieve better ones. This implies that the optimum in this region of the solution space will be identified (Ross, 2005, pp. 529-530). In essence, a rough assumption has to be made to define the modification strategy as mentioned earlier that automatically discard many feasible solutions, which might lead to finding a global optimum for a problem. This major drawback gave a solid motivation to academia to address the problem of the local optimum. As a result, many sophisticated optimization methodologies under the name metaheuristics have been presented to solve very difficult combinatorial optimization problems (Glover and Kochenberger, 2003). Metaheuristics are guided local search algorithms, which consist of two main fundamental elements: A local search algorithm and an overall optimization or control strategy. The control strategy is used to guide and control the local search algorithm (Ross, 2005, p. 530). Simulated Annealing (SA) (Kirkpatrick et al., 1983), Tabu Search (TS) (Glover, 1989), Genetic Algorithms (GA) (Goldberg, 1989; Holland, 1975) are some examples of often adapted metaheuristics to solve HFS scheduling problems (Aurich et al., 2016; Mirsanei et al., 2011; Nahhas et al., 2016; Reeves, 1995; Ruiz and Maroto, 2006).

Reeves (1995) presented one of the first implementation of genetic algorithms to solve scheduling problems. The author addressed the flow shop scheduling problems with the objective function to minimize the makespan $\mathrm{F_m} \mid permu \mid C_{\max}$. The problem is NP-hard. The author investigated a special form of flow shop, in which the permutation of schedule is maintained after the sequencing process on the first machine. A permutation flow shop is a special case of the flow shop except that after building the sequence on the first, the First-In-First-Out (FIFO) discipline is used to further dispatch jobs from the in-production queues. The computational results comprised a comparison between the presented a GA and SA approaches and a local search algorithm. For all investigated problem instances, both GA and SA outperformed the local search algorithm. Similarly, Zheng and Wang (2003) presented a hybrid implementation to deal the same problem. The authors adopted GA to address the problems and further incorporated the heuristic presented by Nawaz et al. (1983), which is known with the name NEH heuristic to generate the initial population before triggering the GA. Their main idea is to size the advantage of a high-quality initial population of solution candidates, which helps GA to systematically search in the regions of high-quality solutions by the beginning of the first iteration.

One of the first contributions addressing HFS problems using GA was presented by Serifoğlu and Ulusoy (2004). The authors presented a comprehensive analysis of their implementation with the objective function to minimize the makespan $HFS_m \parallel C_{max}$. The problem is NP-hard with a simple reduction on the problem treated by Gupta (1988), in which a single machine one the second stage has been taken into consideration. The authors encoded the problem based on the permutation of jobs on the first production stage. The genetic algorithms are employed to solve the sequencing problem in the first stage, whereby Last-In-First-Out (LIFO) discipline has been applied to dispatch jobs on the other production stages. The experiments were conducted on a benchmark datasets, in which up to 100 jobs have been taken into consideration. Very similarly implementation was presented by Oĝuz and Ercan (2005) to deal with the same problem.

Ruiz and Vázquez-Rodríguez (2010) presented a comprehensive literature review, which involved the analysis of over two hundred contributions in the HFS research between the early seventies and 2010. The presented results showed that over sixty percent of the contributions targeted the minimization of the makespan as an objective function. A similar review of the HFS research has been presented by Ribas et al., (2010). The authors, however, restricted their review timeline between the middle nineties and 2010. The findings of the review also stressed on the unrealistic problem formulations. The authors also explicitly pointed out the lack of contributions, in which the investigated problems are formulated based on real-world experiences with real data used for the evaluation of the solution approaches.

Nahhas et al. (2016) presented one of the few contributions, which involved the minimization of the makespan as well as the total tardiness. The authors adapted TS and SA metaheuristics and a heuristic approach named ISBO to deal with a two-stage hybrid flow shop scheduling problem with identical parallel machines on each stage $HFS_2(P4, P5) \mid s_{g,h} \mid C_{max}, \sum T_j$. The problem was investigated under family major and minor setup times constraint on the first production stage. The authors evaluated their solution approaches based on an industrial use case using real data. This work will extend the conducted analysis and present two hybrid solution approaches that combine the ISBO with GA to solve the two-stage problem and then analyze the performance of the approaches and the impact of the complexity for solving the four-stage problem $HFS_4(P4, P5, 1, 1) \mid s_{g,h} \mid C_{max}, \sum T_j$.

4. SOLUTION APPROACHES

The problem to minimize the makespan of a two-stage flow shop is NP-hard (Gupta, 1988). Similarly, the four-stage problem is also NP-hard with a simple reduction on the two-stage one. The development of a polynomial algorithm, which guarantees an optimal solution in a reasonable computational time, is unlikely possible. Thus, dealing with the allocation and the sequencing parts of the problem independently can be a key element to obtain near optimal solution in reasonable computational time. This implies that first a new solution for the allocation is obtained through the GA or a heuristic. Consequently, four single machine problems with family major and minor setup times emerge on the first production stage and five single machine problems with sequence-independent setup times arise on the second production stage. Finally, a single machine problem on the third and the fourth stages. For dealing with the sequencing part of the problem, the sequencing algorithm presented by Nahhas et al. (2016) is adapted. For solving the allocation part of the problem, two hybrid approaches are presented and compared to GA. The first hybrid solution strategy presented is the Improved Integrated Simulation Based Optimization (IISBO). The second solution strategy involves another combination of the ISBO and GA (ISBO & GA). Finally, those approaches will be compared to GA.

4.1. Improved Integrated Simulation Based Optimization (IISBO)

This solution approach is based on the ISBO solution strategy. The conducted analysis in Nahhas et al., (2016) revealed a considerable potential of the ISBO in terms of minimizing the makespan and the total number of major setup times in comparison to the Tabu Search (TS) and Simulated Annealing (SA). In the context of this work, some sensitive parameters of the ISBO are identified and further passed to the Genetic Algorithms (GA). Thus to present a hybrid approach that size the advantages of the metaheuristic (high-quality solution, robustness, etc.) and propose quick solutions likewise the heuristic. This implies that the GA deals with the investigated problems indirectly through optimizing the performance of the heuristic.

Briefly, the ISBO is a constructive heuristic, in which the production schedule is obtained in a single simulation run using an integrated sequencing and allocation algorithms in a simulation model. The ISBO conceptual design is based on a periodical balancing of the production load of the high priority jobs between the available machines during the simulation. The reallocation processes are conducted based on a static predefined reallocation event-list by the end of each simulated working day during the scheduling period. However, this static behavior might lead in some instances to violations in the delivery dates of jobs, if an inappropriate reallocation process is triggered. Furthermore, the production load is balanced between the machines taking into consideration a static balancing indicator of four working days, which might not be optimum for all cases. The allocation algorithm reallocates the families based on the balancing indicator, in which the next four highest priorities are roughly equally distributed between the machines to avoid unnecessary major setups and sustain a balance in the production load between the machines. Accordingly, in spite of the outperformance of the ISBO in terms of minimizing the makespan and the total number of major setup times, some violations in the delivery date of jobs have been encountered. The IISBO is a combination of GA and the ISBO, in which the GA are employed to optimize the reallocation event-list and the balancing period as demonstrated in the conceptual design of the IISBO in Figure 2. Those parameters are encoded in a genetic representation to optimize the performance of the ISBO using the GA.

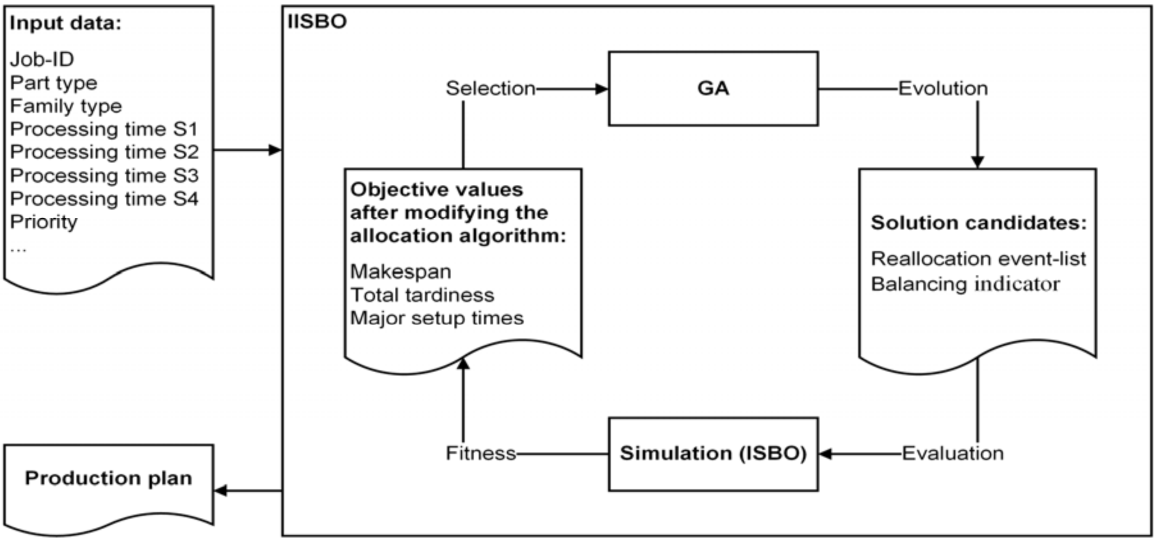

Figure 2: The conceptual design of the IISBO.

The optimization process starts by randomly generating a population of different possible sets of the parameter by the GA for the ISBO. This implies that a genome of the GA is a set of the required parameters of the ISBO. Then the solution candidates or the different parameter sets are passed to the ISBO simulation model to be evaluated, in which the sequencing algorithm presented by Nahhas et al. (2016) is implemented to build the sequence of jobs on each SMD machine. For the dispatching on the Automatic Optical Inspection (AOI) processing stage and the Selective Soldering (SS) processing stage, the Earliest Due Date (EDD) priority dispatching rule is used to optimize the total tardiness taking into consideration the required processing time of jobs on the third and fourth processing stages. Moreover, the sequencing algorithm is used to dispatch jobs on the Conformal Coating (CC) processing stage. The allocation algorithm access the current solution candidate of the GA to trigger the reallocation process during the simulation. In addition, the families are balanced between the machines based on the balancing indicator, which is also generated by the GA.

After evaluating the solution candidates based on the objective function and assigning fitness values, tournament selection strategy is used to select parents for evolving a new generation of solution candidates. For a comprehensive discussion about tournament selection strategy for GA, one can refer to the contribution of Miller and Goldberg (1995). One of the main advantages of this selection strategy is the opportunity to select considerably low-quality solution candidates, which contribute in maintaining a higher diversity among the solution candidates to avoid being trapped in local optimum caused by false convergence. From the selected solution candidates, a uniform crossover operator is used to generate the genes of the offspring solution candidates. Thereafter, based on a random mutation rate the genes of some individuals are mutated to maintain diversity in the population of solution candidates. Moreover, elitism strategy is implemented in this GA. Elitism strategy simply ensures that the best solution candidate so far found in the search process will survive to the next generation (Konak et al., 2006, p. 1001). After creating the new generation, the solution candidates are evaluated using the simulation model to assign fitness values before starting the selection process again, if the GA did not converge or the maximum number of generations is reached. The convergence function is based on calculating the relative difference between the best and the worst solution candidates in the population using the mean and/or the median of their fitness values.

4.2. Combination of GA and ISBO (GA & ISBO)

The second hybrid approach is also a combination between the same GA and the ISBO. However, the GA deals with the problems directly in this solution approach. The encoding strategy of the GA is targeting the allocation part of the problem on the first processing stage (SMD). This implies that a genome of the GA represents the allocation of families to the SMD machines. The sequencing part of the problem is treated using the same sequencing algorithm identically to the previous approach. However, for evaluating the solution candidates of the GA, the ISBO simulation model is used, in which the allocation of the GA is then manipulated during the simulation through the allocation algorithm of the ISBO. In this solution strategy, the GA propose in essence the initial allocation for the ISBO before starting the simulation. A conceptual representation of the solution strategy is represented in Figure 3.

The third solution approach is based on GA. The solution candidates of the GA are encoded to deal with the allocation part of the problem on the SMD processing stage. The rest of the logic is identical to the other solution approaches. In addition, the solution candidates are evaluated using a normal simulation model that describes the behavior of the considered production system.

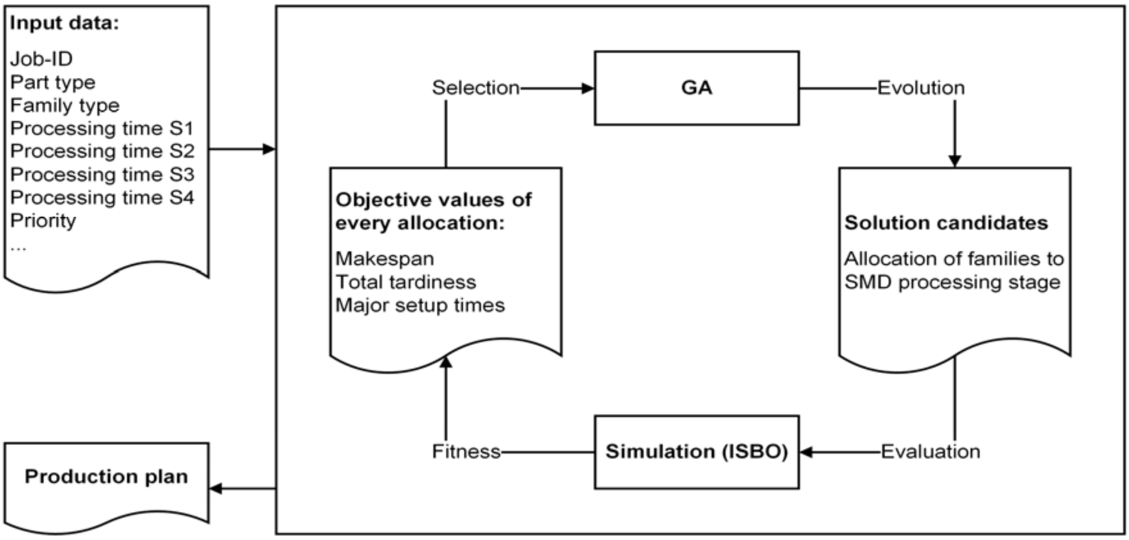

Figure 3: The conceptual design of the (GA & ISBO).

5. COMPUTATIONAL RESULTS

The presented approaches are evaluated by solving four problem instances of the two- and the four-stage problems. The experiments are designed to investigate the quality of the obtained schedules based on the objective functions as well as the required computational effort to obtain the solutions. The solution approaches are implemented and integrated into a simulation model that is built using the ExtendSim 9.1 simulation package. A descriptive information of the used datasets is presented in Table 1. The processing times are given in minutes.

Table 1: Descriptive information of the input data.

	Number of jobs	Number of families	Accumulated SMD processing time	Accumulated AOI processing time	Accumulated SS processing time	Accumulated CC processing time
Dataset 1	164	41	54,685	72,528	-	-
Dataset 2	170	37	62,345	88,702	-	-
Dataset 3	175	36	61,274	74,738	-	-
Dataset 4	143	35	56,250	79,294	-	-
Dataset 5	181	21	55,344	68,952	22,803	21,155
Dataset 6	179	23	65,470	81,601	21,661	17,749
Dataset 7	194	24	44,270	55,576	22,040	20,439
Dataset 8	170	29	55,457	68,585	23,791	20,442

In order to facilitate the comparison between the solution approaches, a convergence function is used to break the optimization process in all approaches if they converge to 99 %. The convergence function is based on calculating the relative difference between the best and the worst solution candidates of the GA population using the mean of their fitness values. To overcome the stochastic nature of the presented approaches, ten optimization runs have been recorded for solving each problem instance for all approaches. The population size used in the IISBO, the (GA&ISBO) and the GA are 15, 25 and 50 respectively. The applied maximum number of generation is 1000. The used mutation rate is 0.4. Those values have been empirically obtained based on an intensive analysis. A weighted sum approach has been adopted to formulate the objective function. The makespan, the total number of major setup times and the total tardiness are weighted with 0.4, 0.2 and 0.4 respectively. The considered scheduling period is four weeks with three shifts operating model. We reduced

the overall production capacity in 10 % to considered machine breakdowns in an aggregated form.
The computational results for solving the problems are presented in Figure 4. The experiments are conducted on a computer with the following characteristics: CPU 4 x 2.6 GHz, RAM 8 GB and Windows operating system. As demonstrated, a clear domination by a single solution technique cannot be concluded. However, the IISBO and the (GA&ISBO) share a clear domination over the GA and the ISBO in terms of minimizing the makespan and the total number of major setup times. More precisely, the (GA&ISBO) deliver the best makespan on three occasions for solving the two-stage problem, whereby the IISBO also reports the best makespan in three occasions for solving the four-stage problem. However, the IISBO clearly outperforms the other hybrid approach in terms of the required computational effort for solving all problem instances. In the same context, both approaches deliver optimized production schedules in considerably less computational time in comparison to the GA. Some conclusions on the total tardiness can be drawn. Clearly, the ISBO performs worst in comparison to the other solution approaches since seven penalties are indicated in all datasets. On the contrast, the GA deliver optimized production schedules for all datasets without reporting violations in the delivery dates.

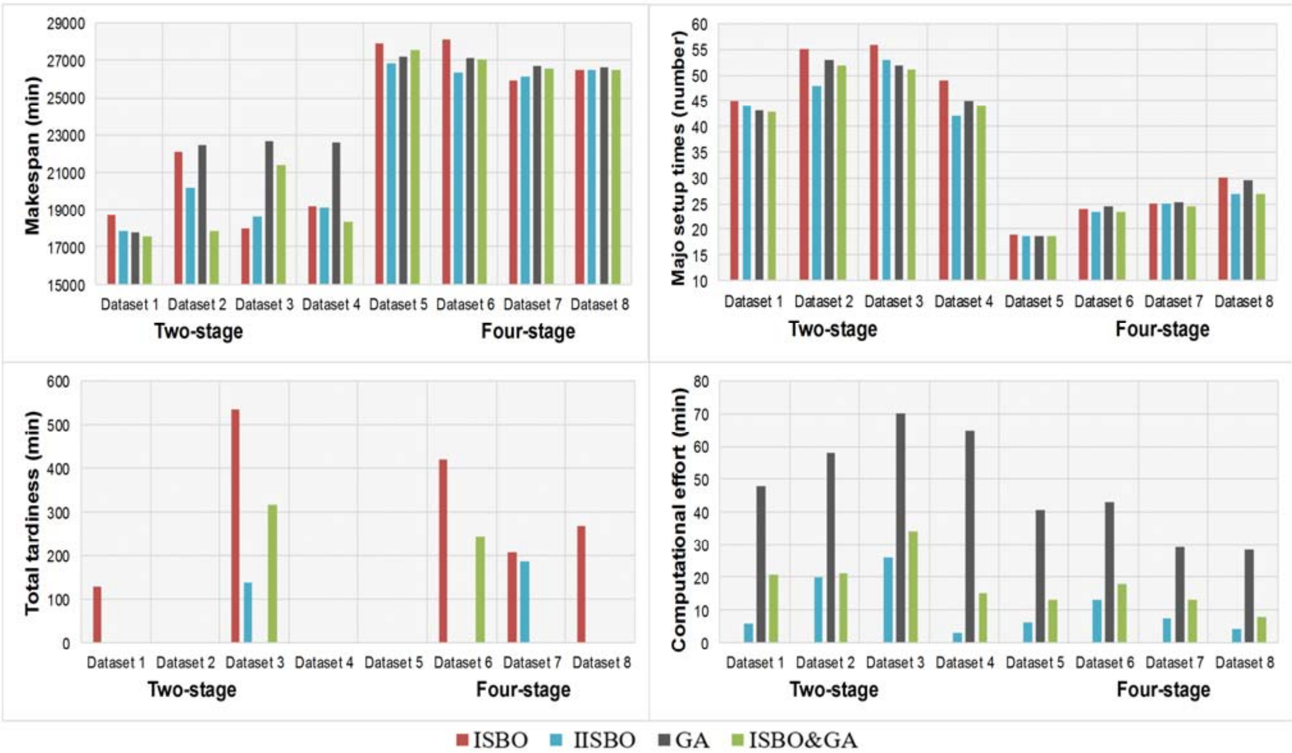

Figure 4: The computational results of the solution approaches for solving the problems.

6. CONCLUSION AND FUTURE WORK

In this paper, two hybrid solution approaches: improved integrated simulation-based optimization (IISBO) and the combination of the GA and ISBO (GA&ISBIO) are evaluated against the genetic algorithm (GA) and the (ISBO) for solving hybrid flow shop (HFS) scheduling problems. The performance of the IISBO and the hybrid approach is pretty similar in minimizing the makespan, the number of major setup times and the total tardiness. The IISBO outperforms the hybrid approach in terms of the required computational time to obtain the solutions. In average, the IISBO obtains the solutions in roughly 50 % of the required computational time of the hybrid approach. Furthermore, the hybrid approach clearly dominates the GA in optimizing the objective values as well as in the required computational time. In addition, the designed improvement strategy for the ISBO did not only overcome the drawback of the total tardiness but also reports significant improvements in terms of optimizing the makespan and the total number of major setup times.
The performance of the GA is strongly impacted by the random initialization of the solution candidates at the beginning of the optimization, which requires GA longer computational time to overcome the very low-quality initial population. This can be explained by a strongly unbalanced allocation of families on the SMD processing stage, which accordingly leads to an increase in the makespan and number of major setup times. Therefore, developing a simple heuristic or adapting some introduced heuristic in the literature might be a matter of further research to generate the start population of the GA to overcome the computational effort drawback.
We conducted ten optimization experiments on four problem instances of the two and the four stages

problems, which allows us to conclude that the quality of presented solution approaches is stable for solving many problem instances. The increase of the complexity for solving the four-stage problem leads to a decrease in the optimization margin, which the solution approaches strive to achieve. This statement can be supported by observing the performance of the solution approaches in the last four datasets and specifically for minimizing the makespan and the number of major setup time objective values. We can notice that the difference in the performance of the solution approaches in the first four datasets witnessed stronger deviation, whereby in the last four datasets, slight deviations can be noticed. In addition, the required computational time did not increase for solving the last four problem instances with the increase of the complexity in comparison to the first four problem instances. In fact, it rather decreases due to the less number of considered families in the four-stage problem instances.

REFERENCES

Andersson, M., A. H. C. Ng, and H. Grimm. 2008. "Simulation Optimization for Industrial Scheduling Using Hybrid Genetic Representation." In *Proceedings of the 40th Conference on Winter Simulation,* 2004–2011: Winter Simulation Conference.

Aurich, P., A. Nahhas, T. Reggelin, and J. Tolujew. 2016. "Simulation-based Optimization for Solving a Hybrid Flow Shop Scheduling Problem." In *Proceedings of the 2016 Winter Simulation Conference,* 2809–2819, Piscataway, NJ, USA: IEEE Press.

Baker, K. R., and D. Trietsch. 2009, Principles of Sequencing and Scheduling, Wiley-Blackwell: Oxford.

Geiger, C. D., R. Uzsoy, and H. Aytuğ. 2006. "Rapid Modeling and Discovery of Priority Dispatching Rules: An Autonomous Learning Approach." Journal of Scheduling, 9 (1), 7–34.

Glover, F. 1989. "Tabu search—part I." ORSA Journal on computing, 1 3, 190–206.

2003, Handbook of Metaheuristics, Springer Science & Business Media. Glover, F., and G. A. Kochenberger, eds.

Goldberg, D. E. 1989, Genetic algorithms in search, optimization, and machine learning, Addison-Wesley: Reading, Mass., Wokingham.

Graham, R. L., E. L. Lawler, J. K. Lenstra, and Rinnooy Kan, A. H. G. 1979. "Optimization and Approximation in Deterministic Sequencing and Scheduling: a Survey." In *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,* edited by E. J. P.L. Hammer and B.H. Korte, 287–326: Elsevier.

Gupta, J. N. D. 1988. "Two-Stage, Hybrid Flowshop Scheduling Problem." The Journal of the Operational Research Society, 39 4, 359.

Held, M., and R. M. Karp. 1961. "A Dynamic Programming Approach to Sequencing Problems." In *Proceedings of the 1961 16th ACM National Meeting,* 196–210, New York, NY, USA: ACM.

Holland, J. H. 1975, Adaptation in natural and artificial systems. *An introductory analysis with applications to biology, control, and artificial intelligence / by John H.Holland*, University of Michigan Press: Ann Arbor, Mich.

Hunsucker, J. L., and J. R. Shah. 1994. "Comparative performance analysis of priority rules in a constrained flow shop with multiple processors environment." European Journal of Operational Research, 72 (1), 102–114.

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

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

Kis, T., and E. Pesch. 2005. "A review of exact solution methods for the non-preemptive multiprocessor flowshop problem." European Journal of Operational Research, 164 3, 592–608.

Konak, A., D. W. Coit, and A. E. Smith. 2006. "Multi-objective optimization using genetic algorithms: A tutorial." Reliability Engineering & System Safety, 91 9, 992–1007.

Kut C. So. 1990. "Some Heuristics for Scheduling Jobs on Parallel Machines with Setups." Management Science, 36 4, 467–475.

Lenstra, J. K., A. Rinnooy Kan, and P. Brucker. 1977. "Complexity of Machine Scheduling Problems." In *Studies in Integer Programming,* edited by P.L. Hammer, E.L. Johnson, B.H. Korte, and G.L. Nemhauser, 343–362: Elsevier.

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.

Maccarthy, B. L., and J. Liu. 1993. "Addressing the gap in scheduling research: a review of optimization and heuristic methods in production scheduling." International Journal of Production Research, 31 (1), 59–79.

Miller, B. L., and D. E. Goldberg. 1995. "Genetic algorithms, tournament selection, and the effects of noise." Complex systems, 9 3, 193–212.

Mirsanei, H. S., M. Zandieh, M. J. Moayed, and M. R. Khabbazi. 2011. "A simulated annealing algorithm approach to hybrid flow shop scheduling with sequence-dependent setup times." Journal of Intelligent Manufacturing, 22 6, 965–978.

Nahhas, A., P. Aurich, T. Reggelin, and J. Tolujew. 2016. "Heuristic and Metaheuristic Simulation-Based Optimization for Solving a Hybrid Flow Shop Scheduling Problem." In *The 15th*

International Conference on Modeling and Applied Simulation, edited by A. G. Bruzzone, et al., 95–103, RENDE (CS), ITALY.

Nawaz, M., E. E. Enscore, and I. Ham. 1983. "A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem." Omega, 11 (1), 91–95.

Oĝuz, C., and M. F. Ercan. 2005. "A Genetic Algorithm for Hybrid Flow-shop Scheduling with Multiprocessor Tasks." Journal of Scheduling, 8 4, 323–351.

Orlin, J. B., A. P. Punnen, and A. S. Schulz. 2004. "Approximate Local Search in Combinatorial Optimization." In *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms,* 587–596, Philadelphia, PA, USA: Society for Industrial and Applied Mathematics.

Pinedo, M. L. 2012, Scheduling: Theory, Algorithms, and Systems, Springer New York.

Pirlot, M. 1996. "General local search methods." European Journal of Operational Research, 92 3, 493–511.

Reeves, C. R. 1995. "A genetic algorithm for flowshop sequencing." Computers & Operations Research, 22 (1), 5–13.

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.

Ross, P. 2005. "Hyper-Heuristics." In *Search Methodologies,* edited by E. K. Burke and G. Kendall, 529–556, Boston, MA: Springer US.

Ruiz, R., and C. Maroto. 2006. "A genetic algorithm for hybrid flowshops with sequence dependent setup times and machine eligibility." European Journal of Operational Research, 169 3, 781–800.

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

şerifoğlu, F. S., and G. Ulusoy. 2004. "Multiprocessor task scheduling in multistage hybrid flow-shops: a genetic algorithm approach." Journal of the Operational Research Society, 55 5, 504–512.

Voß, S. 1993. "The Two — Stage Hybrid — Flowshop Scheduling Problem with Sequence — Dependent Setup Times." In *Operations Research in Production Planning and Control,* edited by G. Fandel, et al., 336–352: Springer Berlin Heidelberg.

Wang, K.-J., J. C. Chen, and Y.-S. Lin. 2005. "A hybrid knowledge discovery model using decision tree and neural network for selecting dispatching rules of a semiconductor final testing factory." Production Planning & Control, 16 7, 665–680.

Wittrock, R. J. 1990. "Scheduling parallel machines with major and minor setup times." International Journal of Flexible Manufacturing Systems, 2 4, 329–341.

Zheng, D.-Z., and L. Wang. 2003. "An Effective Hybrid Heuristic for Flow Shop Scheduling." The International Journal of Advanced Manufacturing Technology, 21 (1), 38–44.

AUTHORS BIOGRAPHY

ABDULRAHMAN NAHHAS is a research associate in the Very Large Business Application Lab at the Otto von Guericke University Magdeburg. His main research and work interests are IT-system landscape engineering, scheduling problems and simulation-based optimization solution strategies. Abdulrahman Nahhas holds a bachelor and a master degree in Business Informatics from Otto von Guericke University Magdeburg. His email address is abdulrahman.nahhas@ovgu.de.

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. 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.

SASCHA BOSSE is a research coordinator and a project manager at the Very Large Business Application Lab. His professional field of interests includes IT service management, modeling, simulation, and optimization. Sascha Bosse received a doctoral degree in engineering from the Otto von University Magdeburg. Furthermore, he holds a bachelor and a master degree in Business Informatics from Otto von Guericke University Magdeburg. His email address is sascha.bosse@ovgu.de

TOBIAS REGGELIN is a project manager at the Fraunhofer Institute for Factory Operation and Automation 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.

Klaus Turowski is heading the Chair of Business Informatics (AG WI) at the Otto von Guericke University Magdeburg, the Very Large Business Applications Lab and the world's largest SAP University Competence Center. His professional field of interests includes IT service management, IT system landscape engineering, SAP systems, and optimization. Klaus Turowski achieved his doctorate at the Institute for Business Informatics at the University of Münster

and habilitated in Business Informatics at the Faculty of computer science at the Otto von Guericke University. His email address is klaus.turowski@ovgu.de.