Proceedings of the 2017 Winter Simulation Conference
W. K. V. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page, eds.

TOWARDS ADAPTIVE SIMULATION-BASED OPTIMIZATION TO SELECT INDIVIDUAL DISPATCHING RULES FOR PRODUCTION CONTROL

Mirko Kück
Eike Broda
Michael Freitag

BIBA - Bremer Institut für Produktion und Logistik
GmbH at the University of Bremen
Hochschulring 20
28359 Bremen, GERMANY
Faculty of Production Engineering
University of Bremen
Badgasteiner Straße 1
28359 Bremen, GERMANY

Torsten Hildebrandt

SimPlan AG, Niederlassung Nord
Hochschulring 20
28359 Bremen, GERMANY

Enzo M. Frazzon

Industrial and Systems Engineering Department
Federal University of Santa Catarina
Campus UFSC
88040-970, Florianópolis, SC, BRAZIL

ABSTRACT

Due to the increasing complexity of contemporary production scheduling problems, it is generally not possible to calculate nearly optimal production schedules in an acceptable amount of time. Hence, normally, dispatching rules are used to determine the job sequences. However, the selection of suitable dispatching rules is not a trivial task and depends on the relevant key performance indicators. Moreover, the suitability of dispatching rules changes over time because of the stochastic and dynamic nature of manufacturing systems. This paper proposes an adaptive simulation-based optimization approach to select individual dispatching rules for production control. The paper's contribution is two-fold. First, it shows that the proposed approach improves the performance compared to benchmark approaches in a manufacturing scenario from semiconductor industry. Second, in order to be able to react quickly to dynamic changes, it proposes strategies for maintaining information from previously calculated solutions after a change, such as a machine breakdown, occurred.

1 INTRODUCTION

In order to achieve a high performance, manufacturing companies have to accomplish a well-founded production scheduling and control. However, since production processes become more and more complex, these tasks demand the use of sophisticated methods. Since most production scheduling problems are NP-hard optimization problems, optimal scheduling solutions often cannot be computed or only in extremely long computation times (Papadimitriou 2003, Pinedo 2016). Therefore, in literature, it is suggested to use meta-heuristics to calculate nearly optimal solutions instead of exact mathematical optimization models. But despite of the computational power of today's computers, the necessary time to compute solutions for complex scheduling problems by meta-heuristics is still too high to be acceptable in practical applications. Thus, shop floor control is mostly performed based on dispatching rules (Rajendran and Holthaus 1999, Pickardt and Branke 2012). However, the choice of suitable dispatching rules is not a trivial task and depends strongly on the layout of a manufacturing system as well as the relevant key performance indicators. In addition, previous research has shown that the suitability of dispatching rules can change over time since manufacturing systems are subjected to stochastic and dynamic influences, such as rush orders or machine breakdowns (Kück et al. 2016). Hence, this paper proposes an adaptive simulation-based optimization (ASBO) approach to select suitable dispatching rules for each machine of a manufacturing system with different strategies to react to system changes. The approach applies a genetic algorithm to generate possible solutions, i. e. sets of dispatching rules, and evaluates the qualities of the proposed solutions in a discrete-event simulation model. The approach is applied to a manufacturing scenario from semiconductor industry: the FAB6 model from the publically available MIMAC (Measurement and Improvement of Manufacturing Capacities) testbed (Fowler and Robinson 1995, Feigin, Fowler, and Leachman 1996). The experimental results show that the proposed ASBO approach improves the performance regarding average production cycle times in comparison to a previously proposed approach selecting one dispatching rule for all machines (Kück et al. 2016). In addition to this, the paper proposes different strategies to use information from previously calculated solutions after changes of the manufacturing system, e. g. machine breakdowns, occurred.

The remainder of this paper is structured as follows. First, the state of the art is detailed, covering: traditional approaches for production scheduling and control as well as simulation-based optimization approaches applied to scheduling and control of manufacturing systems. Subsequently, the proposed adaptive simulation-based optimization approach is described. Then, the experimental setup and the experimental results are detailed. The paper closes with a conclusion and suggestions for future research.

2 LITERATURE REVIEW

2.1 Traditional Approaches for Production Scheduling and Control

Production scheduling can be defined as the task of assigning a number of jobs to the available resources of a production system (Pinedo 2016). Since most scheduling problems, especially those emerging from real-world scenarios, belong to the class of NP-hard optimization problems, optimal scheduling solutions often cannot be computed or only in extremely long computation times. This is the reason for the extensive use of heuristic methods instead of exact mathematical optimization models. Heuristic methods cannot guarantee optimal solutions but are often able to generate near-optimal solutions in relatively short computation times (Papadimitriou 2003). A sophisticated approach is to compute a production schedule by a meta-heuristic method, such as a genetic algorithm or particle swarm optimization. Meta-heuristics are able to compute solutions for larger instances of combinatorial problems than exact mathematical optimization models. However, they also feature limitations like the dependence on the choice of several parameters and the possibility to converge towards local extrema (Jungwattanakit et al. 2008). Moreover, the computation of solutions for very large problems also takes too much time in general. Thus, often no complete schedule is computed in advance but the job sequence is determined according to dispatching rules. In this case, each job in the queue of a machine is related with a priority value according to some predefined criteria, for example the time that is left until its due date (Pickardt and Branke 2012, Rajendran and Holthaus 1999). Whenever a suitable machine is available, the job with the highest priority is chosen for the next production step. Dispatching rules are characterized by a low effort for application and implementation and are therefore often used. They generate quite tight schedules and prevent idle times of machines. However, the choice of suitable dispatching rules is not trivial and depends highly on the application scenario.

While there are few approaches to describe the influence of dispatching rules on average production lead times analytically (Hübl, Jodlbauer, and Altendorfer 2013), an often applied approach for evaluating the performance of calculated production schedules or sets of dispatching rules is developing a simulation model representing the real manufacturing system and using this simulation model for evaluation regarding different key performance indicators. In general, modeling and simulation is an approach for deriving experience-based solutions in order to deal with real-world complex systems (Banks et al. 2000, Law 2015). Simulation enables the decision maker to evaluate several control policies and various replications of a simulation can be carried out to evaluate the robustness of a considered design. However, simulation is only able to evaluate predefined parameter configurations of the simulation model but there is no standard approach for finding appropriate configurations. Hence, this paper combines simulation and meta-heuristics within a simulation-based optimization approach.

2.2 Simulation-Based Optimization

As described above, exact optimization approaches as well as meta-heuristic approaches are often prohibitive for production scheduling since the problem complexity is too high in general. Therefore, these methods can usually only be applied to production scheduling problems if the complex system can be modelled by a simplifying abstraction. In contrast, simulation is a powerful tool for the analysis and evaluation of complex and stochastic systems. However, it cannot provide an efficient optimization of these systems with respect to one or more key performance indicators. Thus, both individual approaches are limited in taking optimal decisions for complex and stochastic systems such as manufacturing systems. A promising approach with the aim of combining the strengths of both is the so-called simulation-based optimization (SBO). This approach evaluates different system configurations through simulation and uses a meta-heuristic to determine nearly optimal configurations of parameters for the simulation (Fu 2002, Krug et al. 2002). Since the simulation model represents the real system in detail, it is not always necessary to express all relations of parameters explicitly in an optimization model. Hence, SBO is a promising approach for solving complex problems. SBO approaches allow for prescriptive analytics by relating input settings and goal performance (Shao, Shin and Jain 2014). Regarding the increase in current systems' complexity, Juan et al. (2015) argued for the relevance of extending meta-heuristics based on simulation, so that they are capable of properly solving stochastic combinatorial optimization problems.

Recently, SBO and related approaches have been successfully applied to different problems in the production and logistics context. In order to deal with a huge search space, multi-objective, and high-variability problems, Lee et al. (2008) combined evolutionary algorithms and simulation for performance estimation. Laroque et al. (2012) developed a fast converging procedure combining particle swarm optimization and genetic algorithms to find suitable parameter configurations in a material flow simulation concerning layouts of the production system of an automotive supplier. Pathak et al. 2014 used SBO with a particle swarm optimization to address manufacturing flow problems of a heavy equipment manufacturer. Ziarnetzky and Mönch (2016) applied an SBO approach for integrated production planning and capacity expansion decisions in a simplified semiconductor supply chain. Aurich et al. 2016 proposed an SBO approach for solving a hybrid flow shop scheduling problem with the objective to minimize the makespan and the total tardiness. Freitag and Hildebrandt (2016) used an SBO procedure with genetic programming to develop specified dispatching rules for the scheduling and control of a complex manufacturing scenario. Vieira et al. (2017) applied a hybrid approach using a genetic algorithm to compute production schedules for small scheduling problems and subsequently evaluating the schedule robustness through discrete-event simulation. Kück et al. (2016) proposed an SBO approach for choosing an aggregate dispatching rule in a semiconductor manufacturing problem. The authors showed that the use of real-time data coupled with a simulation model influenced the choice of better dispatching rules for improving system performance regarding the average production cycle times. The paper at hand extends these results by applying a genetic algorithm to enable the selection of individual dispatching rules for each machine as well as evaluating different strategies to use previously computed solutions after the production system has changed and new solutions have to be calculated. The next section details the adaptive simulation-based optimization approach developed in this paper.

3 ADAPTIVE SIMULATION-BASED OPTIMIZATION APPROACH

This paper proposes an adaptive simulation-based optimization (ASBO) approach consisting of a genetic algorithm (GA) and a simulation model. The GA is used to propose possible solutions, i. e. possible sets of individual dispatching rules for each machine of a job-shop production. The simulation model is used to evaluate the performance of a solution regarding logistic key performance indicators (KPIs). In the following, the ASBO procedure as well as the GA procedure are described in detail.

Figure 1 shows a flow chart of the ASBO procedure proposed in this paper. First, the necessary information about the production system to be optimized as well as the necessary parameters and criteria of the GA are initialized. In order to apply the GA for finding appropriate sets of dispatching rules for the job-shop production scenario, an optimization criterion (fitness function), a termination criterion (number of generations g) as well as the following parameters have to be initialized: population size n, mutation rate m, crossover rate c, elitism count e, and solution length (number of machines k in the job-shop production scenario). Afterwards, an initial population of n feasible solutions is generated randomly or by using available information about possible good solutions. Subsequently, the qualities of all n solutions of the population are evaluated by simulating the production system with the set of dispatching rules proposed in each solution and regarding the achieved KPIs. Based on these KPIs, the fitness of each solution is calculated by the fitness function. Afterwards, the solutions are sorted by their fitness values and a ranking of the n solutions of the population is conducted. If the termination criterion (number of generations g) is fulfilled the best solution of the ranking is taken as output solution. If the termination criterion is not fulfilled a new population is generated by the GA. The input values to the GA are the population $p0$ from the last current iteration of the ASBO procedure, the population size n, the mutation rate m, the crossover rate c, and the elitism count e. The GA procedure consists of two phases: the crossover phase and the mutation phase. The crossover phase starts with initializing a new empty population $p1$ of size n. Afterwards, all n individual solutions of population $p0$ are selected sequentially and the following procedure is conducted. If a selected solution $s(i)$ is an elitism solution, which means that it is one of the e best solutions of population $p0$, it is also added to population $p1$. If a selected solution $s(i)$ is not one of the best solutions, it serves as the first parent for a new solution. Then a second parent solution is selected from population $p0$ and a new solution $s(i)^*$ is created by conducting a crossover according to the given crossover rate c. Subsequently, solution $s(i)^*$ is added to population $p1$. The crossover phase is finished when all n solutions of population $p0$ have been regarded, so that population $p1$ consists of n solutions. Population $p1$ is the output of the crossover phase. Afterwards, the mutation phase starts with initializing a new empty population $p2$ of size n. Similar to the crossover phase, all n individual solutions of population $p1$ are selected sequentially and the following procedure is conducted. If a selected solution $s(i)$ is an elitism solution in population $p1$ it is added to population $p2$. If a selected solution $s(i)$ is not one of the best solutions, a new solution $s(i)^*$ is created by conducting a mutation according to the given mutation rate m and subsequently, solution $s(i)^*$ is added to population $p2$. The mutation phase is finished when all n solutions of population $p1$ have been regarded. The output of the mutation phase, which is also the output of the whole GA procedure, is the new population $p2$ consisting of n feasible solutions. Subsequently, this output population is used in the ASBO procedure.

Since manufacturing systems are complex stochastic and dynamic systems, their configurations can change over time. In this case, the corresponding planning simulation model representing the real manufacturing system has to be adapted to the new system configuration (Figure 1). In order to prevent long computation times, the ASBO approach applies different strategies to maintain information from previously calculated solutions before the system change had occurred. As a step towards adaptivity, in addition to the approach of generating a completely random initial population of solutions, two further strategies are proposed and evaluated in this paper: adding the best aggregate choice of one dispatching rule for all machines to a random initial population or taking the best population of solutions that was found before the system change occurred.

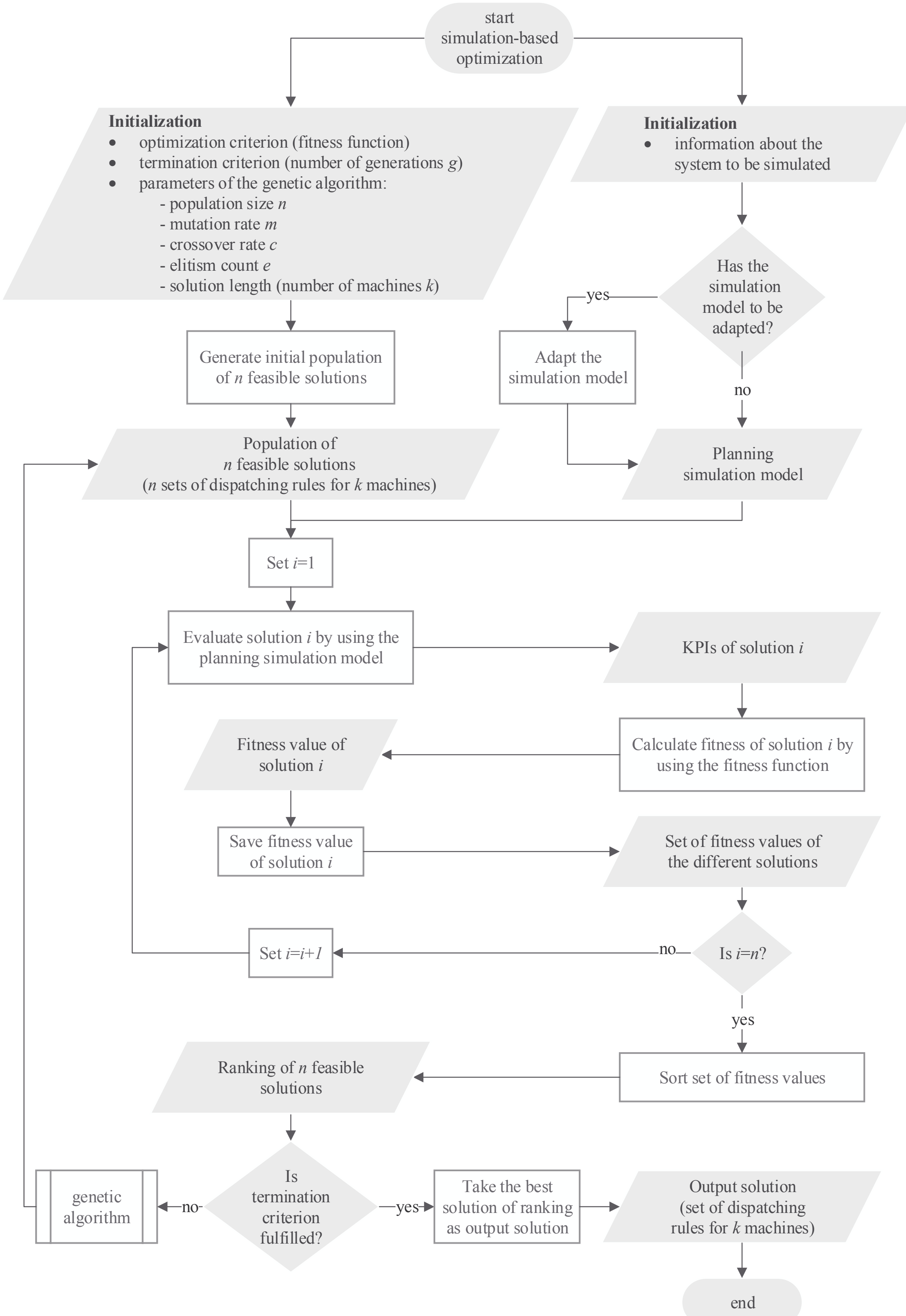

Figure 1: Flow chart of the adaptive simulation-based optimization procedure.

4 EXPERIMENTAL SETUP

4.1 Manufacturing Scenario

Similar to previous research presented in (Kück et al. 2016), the FAB6 model from the publically available MIMAC (Measurement and Improvement of Manufacturing Capacities) testbed (Fowler and Robinson 1995; Feigin, Fowler, and Leachman 1996) is considered as the manufacturing scenario. The model is derived from a real semiconductor manufacturing facility with the following characteristics: nine process flows/products having between 234 and 355 operations, 104 tool groups with a total number of 223 machines including batch machines with sequence-dependent setup times, random machine downtimes (failures and maintenance). The simulation of the system was conducted for a total duration of 18 months, ignoring data from the first 6 months to focus on the system's steady state behavior. About 3800 jobs/lots are started during the 18-month period used for a simulation run. To implement this model the jasima simulation library (jasima - an efficient Java Simulator for Manufacturing and Logistics; http://jasima.net) was used. As stochastic influences in the FAB6 model, the duration and time between downtimes were given using an exponential distribution. All other model parameters, especially job arrivals, were deterministic. Production orders are generated according to a make to order policy with immediate release. The product mix and bottleneck utilization resemble the settings from Zhou and Rose (2011).

In the experiments, for each machine, one the following standard dispatching rules (Haupt 1989) could be selected: FIFO (first in (queue) first out), EDD (earliest due date first), FASFS (first arrival in system first served), CR (critical ratio), ODD (operational due date), SPT (shortest processing time first) and MOD (modified operational due date). To distinguish between jobs with the same priority, the FASFS rule was used as a tiebreaker. While EDD assigns priorities according to the overall due dates of jobs, ODD calculates intermediary due dates for all individual operations of a job and selects the job with the smallest intermediary due date for the current operation. MOD computes the maximum of the operational due date and the sum of the current time plus the operation processing time for each job and selects the job with the smallest value. As some machines require setup times, all dispatching rules were used with a setup-avoidance strategy improving cycle times considerably. The batch machines used the "largest batch first" procedure forming a batch as large as possible for each batch family. Then the largest of these batches was started. Ties were resolved by selecting the batch family containing the job with the highest priority.

4.2 Configuration of the Optimization Algorithm

The optimization algorithm as outlined in Section 3 used the following parameter values of the genetic algorithm. The population size was n=100 possible individuals (solutions). Each individual had a chromosome length of 104, where each position (gene) represents the selected dispatching rule for one of the 104 machines. A generational schema with an elitism count e=10 was used. The genetic operators were roulette-wheel selection, point mutation (checking for every gene whether a mutation should happen, if true – a new random dispatching rule was chosen) with mutation rate m=0.1 as well as uniform crossover (each gene of the offspring had a 50% chance of coming from either its first parent or its second) with crossover rate c=0.9. The optimization criterion was to minimize the average cycle time. Each optimization was run ten times to be able to assess the average performance of the algorithm. The performance results presented in this paper are the averages over 30 independent replications for each of the investigated settings. In order to save computational time, a single replication was used to evaluate candidate solutions during the optimization run but the seed was changed after each generation to avoid overfitting a solution to a particular random seed value.

4.3 Experiments

In this paper, two different scenarios are considered: a scenario without machine failures (scenario "full") as well as a scenario with long machine failures so that only three of the five machines within tool group ”LTS-2”are available (scenario “failure”). For both scenarios, two different experiments were conducted. First, the ASBO approach described in Section 3 was applied to select an individual dispatching rule for each of the 104 machines of the manufacturing scenario. To assess the quality of the calculated solutions, they are compared to the benchmark solutions of selecting the same dispatching rule for all machines, which were previously presented in (Kück et al. 2016). In a second experiment, different strategies for initializing the start population of the ASBO procedure are compared, using information from previously calculated solutions after the production system has changed, i. e. either more or less machines are available. Each individual simulation run took about 2.57 seconds to be evaluated. In order to derive a good solution by the ASBO approach, 200 generations of 100 individual solutions each were computed in about 107 minutes of computation time. However, as Section 4.2 shows, good solutions can already be achieved within 50 generations, taking about 27 minutes of computation time. This short computation time shows the great potential of the ASBO approach to derive high-quality solutions in real-time applications.

5 EXPERIMENTAL RESULTS

5.1 Benchmark Results

As already presented in (Kück et al. 2016), Table 1 shows the benchmarks of choosing one aggregate dispatching rule for all machines. In the case of no machine failures, the MOD rule achieves the lowest average cycle time per job of 24.92 days. If machine failures are included into the simulation, obviously, the average cycle times increase for all selections of dispatching rules. However, the MOD rule is not the best rule any more, but only the third best of the seven possible choices of dispatching rules. Now, the CR rule and the ODD rule perform better. The ODD rule achieves the best average cycle time, saving 20.5 hours for each job on average compared to the MOD rule. This is a cycle time reduction of 3.1%, or a reduction of 5.9% of its reducible components, i. e., the sum of setup and waiting times.

Table 1: Average cycle times per job (in days) for the seven different dispatching rules and different numbers of available machines in machine group “LTS 2”. Also shown: ranks of the dispatching rules for the respective scenario. Values in brackets show twice the standard error across the 30 independent replications. (according to (Kück et al. 2016)).

Aggregate dispatching rule for all machines	Average cycle times [days] / Rank for the case	
	without machine failures (scenario “full”)	with machine failures (scenario “failure”)
MOD	24.92 (±0.08) / 1	27.98 (±0.61) / 3
ODD	24.93 (±0.08) / 2	27.12 (±0.14) / 1
CR	25.69 (±0.07) / 3	27.29 (±0.09) / 2
FCFS	26.06 (±0.11) / 4	29.40 (±0.18) / 4
SPT	26.66 (±0.12) / 5	32.99 (±0.31) / 7
EDD	27.87 (±0.13) / 6	31.99 (±0.19) / 5
FASFS	28.05 (±0.16) / 7	32.14 (±0.19) / 6

5.2 Individual Selection of Dispatching Rules with Different Initialization Strategies

Table 2 summarizes the results of the experiments described in Section 4.3, showing the average cycle times achieved over ten independent optimization runs conducted for each setting of the initial population. For both scenarios with and without machine failures, the average cycle times per job of the average best initial solution, the average best solution after 200 generations as well as the best overall solution after the ten optimization runs of 200 generations each are shown. Moreover, all these values are compared for three different initialization strategies of the start population: complete random initialization, random initialization with adding the best benchmark solution for the particular scenario, and initializing the best population that was calculated in the respective other scenario before a system change, i. e. a breakdown or repair of machines, occurred. In addition, Table 2 also shows the average cycle times of the best benchmark solutions of the two scenarios. As can be seen in the row “Best individual solution after 200 generations” for both, the “full” scenario (left column), as well as for the “failure” scenario (right column), the optimization algorithm is able to find improved solutions compared to the benchmark solutions. The maximum possible improvement in the “full” scenario is a reduction of average cycle time of almost 6 hours per job compared to the benchmark solution of using the MOD rule for all machines (24.68 days vs. 24.92 days). For the scenario with failures, improvements over the best benchmark solution, i. e. using the ODD rule on all machines, seems to be much harder to achieve. The best solution found here reduces the average cycle time per job by about 1.5h (27.06 days vs. 27.12 days).

Table 2: Average cycle times per job (in days) of the selected sets of dispatching rules after running the ASBO procedure as well as the best benchmark solutions. The left column shows the results of using different initialization strategies for the scenario without machine failures and the right column shows the corresponding results for the scenario with machine failures. Values in brackets show twice the standard error across independent replications.

	Average cycle times [days] in the scenario “full” for an initialization with			Average cycle times [days] in the scenario “failure” for an initialization with		
	random population	random population with best benchmark solution	best population for scenario “failure”	random population	random population with best benchmark solution	best population for scenario “full”
Average best initial solution	25.19 (±0.06)	24.93 (±0.00)	24.87 (±0.03)	27.42 (±0.09)	27.44 (±0.24)	27.27 (±0.16)
Average best solution after 200 generations	24.78 (±0.05)	24.83 (±0.04)	24.80 (±0.05)	27.21 (±0.12)	27.32 (±0.21)	27.40 (±0.21)
Best individual solution after 200 generations	24.68 (±0.08)	24.73 (±0.08)	24.72 (±0.08)	27.02 (±0.13)	27.06 (±0.15)	27.08 (±0.15)
Best benchmark solution	24.92 (±0.08)			27.12 (±0.14)		

Dynamic optimization as performed in the ASBO approach presented in Section 3 aims at providing better solutions in a shorter amount of time by, e. g., reusing results from previous optimization runs or by incorporating additional knowledge about the problem. In order to investigate a step towards this direction, this paper compares the convergence behavior of the optimization algorithm starting with a random population without any previous knowledge to two ways of adding additional knowledge. The first strategy adds the best benchmark solution (same rule on each machine) to the initial population. Figure 2 shows the convergence curve of the average solution quality of this strategy compared to the random initialization strategy in the “full” scenario. As can be seen, adding the benchmark solution significantly improves the quality of initial solutions. In the long run, however (at about generation 60), the performance of the runs with the random initial population starts to get better, achieving a slightly better average result at the end of the optimization run. This is an indication of premature convergence occurring in the case of the biased initial population. Initial bias helps to find good solutions quickly but finally the optimization stops at a local optimum and is not able to find further improvements. A similar result is obtained for the “failure scenario. As already mentioned before, in this setting, it seems to be considerably harder to find good solutions. In general, the optimization process is much more volatile, as shown by the rugged convergence curves (Figure 3) and the values of the standard errors (Table 2). The initialization strategy of using the best population from the "full" scenario leads to an improved quality in the first generations. From about generation 40, the optimization process seems to be stuck without being able to converge to better solutions (in fact, solutions even tend to get worse towards the end). Using the random population shows no such sign of premature convergence. There is a trend towards improved solutions as expected. Summarizing both scenarios, it can be seen that the two proposed initialization strategies lead to significant improvements regarding the average best initial solutions compared to random initialization. In the long run, random initialization achieves slightly better solutions, however, not statistically significant better solutions.

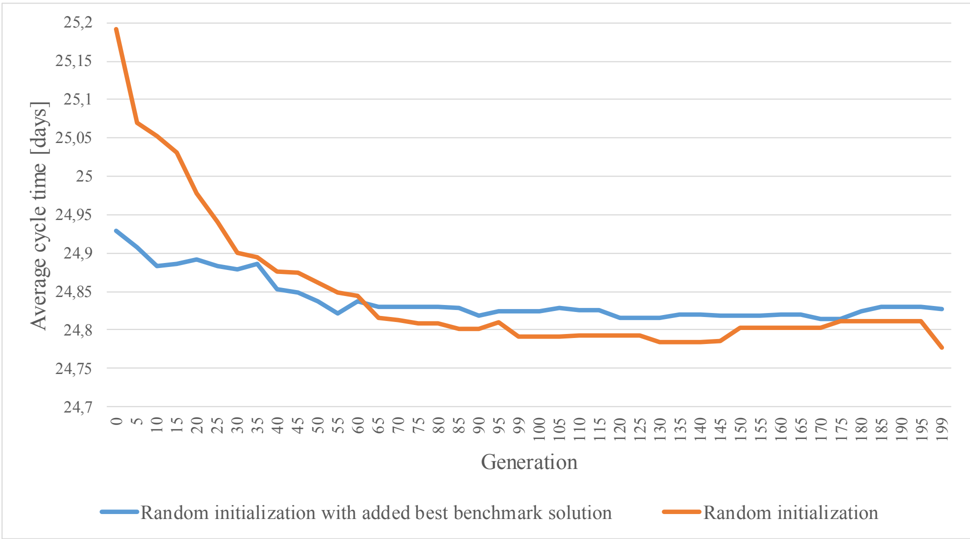

Figure 2: Convergence curve for optimizing the scenario "full" starting with random initial population vs. starting with random initial population and adding the best benchmark solution. Results are averaged over 10 optimization runs conducted for each method of setting the initial population.

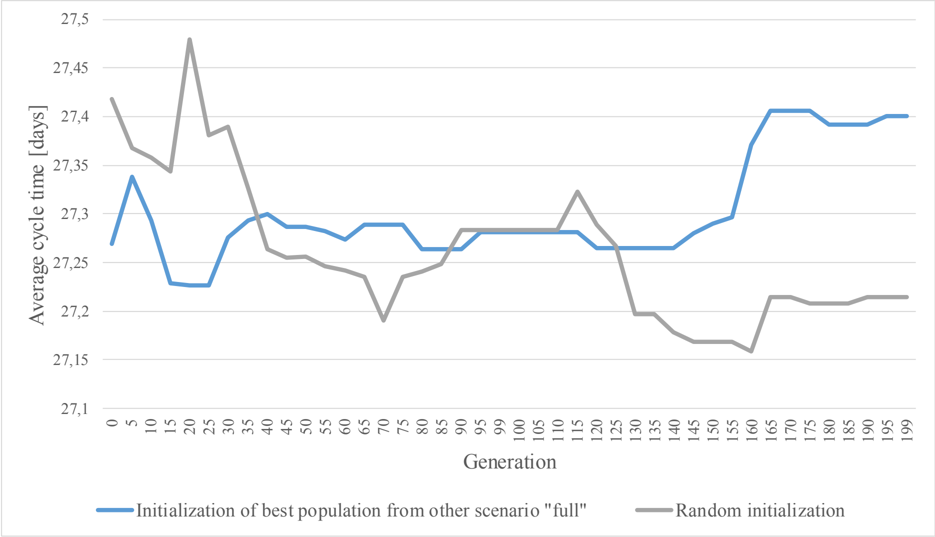

Figure 3: Convergence curve for optimizing the scenario "failure" starting with random initial population vs. reusing the last population of scenario "full". Results are averaged over 10 optimization runs conducted for each method of setting the initial population.

Despite the use of elitism in the optimization algorithm, solutions can get worse during optimization. As stated previously, a single replication during the optimization run is used in order save computational time (but seed values are changed from generation to generation to prevent overfitting to a particular problem instance). The data for the convergence curves base on fully evaluating the best solution (i. e., what the optimization thought would be best) at a certain generation, however. Therefore, each data point in the graphs bases on averaging over ten optimization runs, where each solution was evaluated with 30 replications, independent of those used during the optimization process. The noise and uncertainty evaluating a solution can mislead the optimization algorithm and cause performance to even decrease. This seems to be the case especially when reusing the old population for the new scenario of machine failures (blue curve in Figure 3). All in all, the applied ASBO procedure selects appropriate sets of dispatching rules. However, future work will focus on improved initialization strategies achieving a better convergence behavior.

6 CONCLUSION AND OUTLOOK

This paper proposed an adaptive simulation-based optimization approach to select suitable dispatching rules for production control in complex manufacturing systems. An application to a scenario from semiconductor industry showed that the approach achieves improved average cycle times compared to the benchmark of choosing a single dispatching rule to be used on all machines. In a second experiment, this paper proposed and compared different strategies to use information from previously calculated solutions after a change of the production system, i. e. a machine breakdown, occurred. The experiments showed that the applied simple strategies lead to improved solution quality at the beginning of a new optimization run. However, in the end, starting with a random population achieved slightly better, albeit not significantly better, solutions. This indicates that the simple strategies applied in this paper get stuck in local optima without being able to leave in order to reach a global optimum. Because of this result, future research should focus on more sophisticated strategies to maintain information from previously calculated solutions to be able to react to changes of the production system quickly, e. g. in the direction of Branke (2001) or Nguyen, Wang, and Branke (2012). Moreover, future research will deal with the questions on how often and in which cases a simulation model in the adaptive simulation-based optimization approach should be updated to reflect the current state of the real manufacturing system. Investigating the performance for different application scenarios will indicate the robustness of the proposed approach.

ACKNOWLEDGMENTS

This work is funded by the German Research Foundation (DFG) under reference number FR 3658/1-1 and also by CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil) under reference number 99999.006033/2015-06, in the scope of the BRAGECRIM program.

The work of Torsten Hildebrandt was partly supported by the ZIM project "SimChain", funded by the German Federal Ministry for Economic Affairs and Energy.

REFERENCES

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*, edited by T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, 2809-2819. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Banks, J., J. S. Carson, B. L. Nelson, and D. M. Nicol. 2000. *Discrete-Event System Simulation.* 3rd ed. Upper Saddle River, New Jersey: Prentice-Hall, Inc.

Branke, J. 2001. *Evolutionary Optimization in Dynamic Environments*. Norwell, MA, USA: Kluwer Academic Publishers.

Feigin, G., J. Fowler, and R. Leachman. 1996. MASM Test Data Sets. Accessed May 6, 2016. http://masmlab.engineering.asu.edu/ftp.htm.

Fowler, J., and J. Robinson. 1995. "Measurement and Improvement of Manufacturing Capacities (MIMAC): Final Report." Technical Report 95062861A-TR, SEMATECH, Austin, TX.

Freitag, M., and T. Hildebrandt. 2016. "Automatic Design of Scheduling Rules for Complex Manufacturing Systems by Multi-Objective Simulation-Based Optimization." *CIRP-Annals - Manufacturing Technology* 65: 433-436.

Fu, M. C. 2002. "Optimization for Simulation: Theory vs. Practice." *INFORMS Journal on Computing* 14:192-215.

Haupt, R. 1989. "A Survey of Priority Rule-Based Scheduling." *OR Spektrum* 11:3–16.

Hübl, A., H. Jodlbauer, and K. Altendorfer. 2013 "Influence of dispatching rules on average production lead time for multi-stage production systems." *International Journal of Production Economics* 144:479-484.

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.

Jungwattanakit, J., M. Reodecha, P. Chaovalitwongse, and F. Werner. 2008. "Algorithms for Flexible Flow Shop Problems with Unrelated Parallel Machines, Setup Times, and Dual Criteria." *The International Journal of Advanced Manufacturing Technology* 37:354-370.

Krug, W., T. Wiedemann, J. Liebelt, and B. Baumbach. 2002. "Simulation and Optimization in Manufacturing, Organization and Logistics." In *Proceedings 14th European Simulation Symposium*, edited by A. Verbraeck, and W. Krug, 7 pages. SCS Europe BVBA.

Kück, M., J. Ehm, T. Hildebrandt, M. Freitag, and E. M. Frazzon. 2016. „Potential of data-driven simulation-based optimization for adaptive scheduling and control of dynamic manufacturing systems." In *Proceedings of the 2016 Winter Simulation Conference*, edited by T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, 2820-2831. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Laroque, C., A. Klaas, J.-H. Fischer, and M. Kuntze. 2012. "Fast Converging, Automated Experiment Runs for Material Flow Simulations Using Distributed Computing and Combined Metaheuristics." In *Proceedings of the 2012 Winter Simulation Conference*, edited by C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A.M. Uhrmacher, 2887-2898. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Law, A. M. 2015. *Simulation modeling and analysis*. 5th ed. New York, NY: McGraw-Hill Education.

Lee, L. H., E. P. Chew, S. Teng, and Y. Chen. 2008. "Multi-Objective Simulation-Based Evolutionary Algorithm for an Aircraft Spare Parts Allocation Problem." *European Journal of Operational Research* 189:476-491.

Nguyen, T. T., S. Yang, and J. Branke. 2012. "Evolutionary dynamic optimization: A survey of the state of the art." *Swarm and Evolutionary Computation* 6:1-24.

Papadimitriou, C. H. 2003. "Computational Complexity." In *Encyclopedia of Computer Science*, 260-265. Chichester: John Wiley and Sons Ltd.

Phatak, S., J. Venkateswaran, G. Pandey, S. Sabnis, and A. Pingle. 2014. "Simulation based optimization using PSO in manufacturing flow problems: a case study." In *Proceedings of the 2014 Winter Simulation Conference*, edited by A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, 2136-2146. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Pickardt, C. W., and J. Branke. 2012. "Setup-oriented dispatching rules–a survey." *International Journal of Production Research* 50:5823-5842.

Pinedo, M. L. 2016. *Scheduling: Theory, Algorithms, and Systems*. 5th ed. Heidelberg, Germany: Springer International Publishing.

Rajendran, C., and O. Holthaus. 1999. “A comparative study of dispatching rules in dynamic flowshops and jobshops.” *European Journal of Operational Research* 116:156-170.

Shao, G., S.-J. Shin, and S. Jain. 2014. "Data Analytics Using Simulation for Smart Manufacturing." In *Proceedings of the 2014 Winter Simulation Conference*, edited by A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, 2192-2203. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Vieira, G. E., M. Kück, E. M. Frazzon, and M. Freitag. 2017. “Evaluating the Robustness of Production Schedules using Discrete-Event Simulation.” In *Proceedings of the IFAC 2017 World Congress*. Accepted for publication.

Ziarnetzky, T., and L. Mönch. 2016. “Simulation-based optimization for integrated production planning and capacity expansion decisions.” In *Proceedings of the 2016 Winter Simulation Conference*, edited by T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, 2992-3003. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Zhou, Z., and O. Rose. 2011. “A Composite Rule Combining Due Date Control and WIP Balance in a Wafer Fab.” In *Proceedings of the 2011 Winter Simulation Conference*, edited by S. Jain, R.R. Creasey, J. Himmelspach, K.P. White, and M. Fu, 2085-2092. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

AUTHOR BIOGRAPHIES

MIRKO KÜCK is research scientist at BIBA – Bremer Institut für Produktion und Logistik at the University of Bremen, Germany. His research interests lie in simulation modeling, time series forecasting as well as data mining and machine learning applications in production and logistics. He currently finishes his PhD thesis on the topic of applying meta-learning to select time series forecasting methods for customer demand prediction and production planning. His email address is kue@biba.uni-bremen.de.

EIKE BRODA is a master student at the University of Bremen, Germany. He currently finishes his master thesis regarding an adaptive simulation-based optimization approach for production scheduling. His email address is brd@biba.uni-bremen.de.

MICHAEL FREITAG is full professor of Planning and Control of Production and Logistics Systems in the Faculty of Production Engineering of the University of Bremen. Moreover, he is director of BIBA - Bremer Institut für Produktion und Logistik GmbH at the University of Bremen, Germany. He holds a PhD degree in Production Engineering from the University of Bremen. His primary research interests lie in production and operations management, logistics and supply chain management with a particular focus on simulation modeling and simulation-based optimization. His email address is fre@biba.uni-bremen.de.

TORSTEN HILDEBRANDT is branch manager at SimPlan AG, Germany, one of the largest tool-vendor-independent service providers in the field of simulation. His research interests include simulation-based optimization, heuristic optimization algorithms, as well as planning and control of production and logistic systems in general. He is the main author of the discrete-event simulation library jasima (JAva SImulator for MAnufacturing and logistics, http://jasima.net). His email address is Torsten.Hildebrandt@simplan.de.

ENZO M. FRAZZON is professor in the Industrial and Systems Engineering Department at the Federal University of Santa Catarina in Florianópolis, Brazil. He holds a PhD degree in Production Engineering from the University of Bremen. His research interests include theory and applications of production and logistics systems, with a focus on simulation and optimization. His email address is enzo.frazzon@ufsc.br.