*Proceedings of the 2019 Winter Simulation Conference*
*N. Mustafee, K.-H.G. Bae, S. Lazarova-Molnar, M. Rabe, C. Szabo, P. Haas, and Y-J. Son, eds.*

# NESTED-SIMULATION-BASED APPROACH FOR REAL-TIME DISPATCHING IN JOB SHOPS

Shufang Xie
Tao Zhang
Oliver Rose

Universität der Bundeswehr München
Werner-Heisenberg-Weg 39
Neubiberg, 85577, GERMANY

## ABSTRACT

Real-time dispatching in job shops can be interpreted as a sequential decision-making process in which decisions are made in ordinal stages. In this paper, first we use a branching tree with a time axis to depict the problem, and then propose a nested-simulation-based approach to solve the problem. When a dispatching decision is requested by the manufacturing environment, the simulation will be used to predict the possible future situation and to evaluate all job alternatives. During each alternative simulation process, once a new decision point emerges, the simulation approach will be utilized again, which is called nested alternative simulation in which a simple base rule is adopted for making decisions instead of the simulation. The proposed approach has been superior to some simple rules with respect to the average cycle time and total weighted tardiness.

## 1 INTRODUCTION

In the manufacturing world, we need to decide which machines need to be allocated to which jobs and in what sequence so that some objectives are satisfied, i.e., job shop scheduling problem. In a real-time production environment, as soon as a machine becomes available, we need to decide which job should be selected to be processed first from the waiting queue in front of this machine. During the whole manufacturing process, we have to make a huge number of such dispatching decisions sequentially. The decisions made previously determine the later decisions, and the later decisions influence the performance of the earlier decisions meanwhile, resulting in a sequential decision-making problem.

Dispatching rules are the common way to make decisions online, which have the advantage of being simple and easy to apply. Unfortunately, no simple dispatching rule can perform well across every performance criterion because they are usually based on a single objective function and involve only one model parameter, such as processing time (SPT - shortest processing time) or due date (EDD - Earliest Due Date). In this case, some researchers (Hübel et al.2013; Rajendran et al.1999; Xiong et al.2017) use simulation to evaluate different dispatching rules at each decision point and the dispatching rule with the best performance will be chosen and employed in practice; other researchers (Aydin et al. 2000) use reinforcement learning to train agent so the agent can select the most appropriate priority rule according to the shop conditions in real time. Their work has made the best use of priority rule, however, simple dispatching rules cannot adapt to the changing situation in the manufacturing line (Zhang and Rose 2013) where unexpected events, such as the arrival of new tasks or machine breakdowns, may occur. Many artificial intelligence techniques are used; Adibi et al. (2010) used a trained artificial neural network (ANN) to update parameters of a metaheuristic method at any rescheduling point in a dynamic job shop scheduling problem according to the problem condition. Banu et al. (2015) present a research review of artificial intelligence solution strategies for the job shop scheduling problem.

A large class of sequential decision-making problems under uncertainty can be modeled as a Markov decision process (Littman 1996). Malikppoulos et al. (2009) consider the problem of deriving a control policy for a dynamic system with unknown dynamics in real time, formulated as sequential decision-making under uncertainty. The evolution of the system is modeled as a controlled Markov chain. A new state-space representation model and a learning mechanism are proposed that can be used to improve system performance over time. Yih, Y. and A. Thesen (1991) formulated the scheduling problem as semi-Markov decision problems and used a non-intrusive 'Knowledge acquisition' method to reduce the size of the state space. Transition and reward functions are two essential elements in the Markov decision process (MDP) model. Due to the complexity of the manufacturing line, it is arduous to find a transition function for the MDP model. In addition, the simplified transition function used in the literature lost sight of many key pieces of information in the real-time manufacturing line, such as processing time variations, machine breakdowns, rework and so on. With respect to the reward function which is compatible with the scheduling objectives, accurate validation also requires lots of work.

In this paper, we propose a simulation-based approach which is an efficient approach to scheduling problem and can simulate a long period of time in an actual manufacturing line within a reasonably short computation time. We employ the same basic idea as that of Zhang and Rose (2014) in which once we need to make a dispatching decision, the simulation will be used to evaluate each alternative which means each job waiting in front of the machine. For each job the simulation runs once in deterministic models and several times in stochastic models, the selection of jobs is dependent on the evaluation on the basis of the simulation results. In this study, we will improve the idea and use nested simulation to make decisions in the alternative simulation. The remainder of this paper is structured as follows: Section 2 presents the real-time dispatching problem and sequential decision-making problem; in Section 3, the nested-simulation-based approach is described in detail; the adopted simulation model is briefly introduced in Section 4, along with the experiment results and discussion in Section 5. Finally, based on our findings, this paper is concluded in Section 6.

## 2 PROBLEM STATEMENT

As shown in the picture below (at left in Figure 1), the real-time dispatching decides which job in the queue will be processed first when a machine becomes free. We have to answer these questions again and again during the manufacturing process. It is hard to evaluate how good the answer is to one single question because we cannot measure the contribution of one single decision making in the whole sequential decision-making problem. We can only evaluate it afterward in combination with a certain amount of answers made later and tell how good the answers are. Usually, the evaluation can only be carried out after all questions are answered. Thus, there is a delay from the time that we made the decision to the time that we evaluate the decisions. We can see that collective performance and delayed evaluation are two key characteristics of real-time dispatching. Those characteristics make real-time dispatching very difficult, especially in complex manufacturing systems.

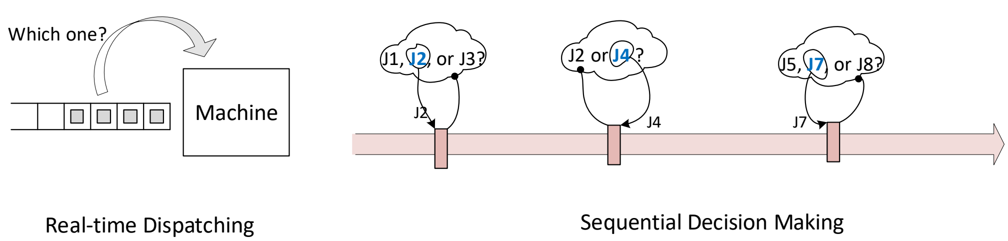


Figure 1: Real-time dispatching and sequential decision making.

If we put all these dispatching questions into a sequence in chronological order (shown at right in Figure 1), a sequential decision-making problem is created. For sequential decisions, the earlier decision determines the later decision while the later decision influences the performance of the earlier decision (Littman 1996). The goal of the sequential decision-making is that all decisions made will result in a good performance of the manufacturing system. Due to the randomness of manufacturing processes, the sequence is also random. Not only the decision-making time and location are random, but also the content of the decision making. Though this conversion of the dispatching into the sequential decision making opens a totally new door to solutions of dispatching problems, as we mentioned before, the algorithms developed in this direction are still at the very beginning phase and lack rigorous theoretical support.

In this study, we are still going to focus on the original dispatching problem and try to tackle the difficulties raised by the two characteristics: collective performance and delayed evaluation. Specifically, the difficulties are that when we evaluate one decision, we must know a certain amount of decisions we will make later and evaluate these decisions together before they are taken in the manufacturing system. This means we must be able to predict the future and do an if-then-analysis in future. Because simulation can predict the future and do if-then-analysis very well for the complex systems, we utilize the simulation and propose a nested-simulation-based approach to solve the problem.

## 3 NESTED-SIMULATION-BASED APPROACH

In this section, we will first give a general idea of our approach, along with a simple example. Afterward, the nested-simulation-based approach will be presented in detail.

### 3.1 Basic Idea

In practical online manufacturing environments, once some machine changes into an idle state and multiple jobs are waiting to become processed on this machine, we need to decide which job is the best suitable candidate at this point so that the overall performance can meet the predefined target more closely. Therefore, in order to get optimal decisions at each decision point, we need to evaluate all the candidate jobs.

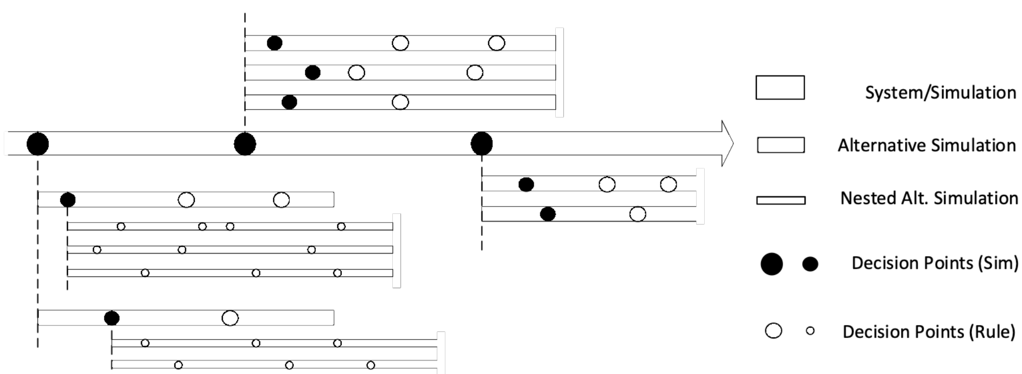


Figure 2: Nested-simulation-based dispatching approach.

In our research, we use a simulation-based approach (see in Figure 2) to substitute the simple priority rules which are commonly adopted by many researchers and practitioners. At every decision point (big black circles) during the production process, we run a simulation for each individual candidate job respectively. This simulation is named alternative simulation (medium blocks). The alternative simulation will predict future decision-making problems and make decisions for these problems. When the alternative simulation ends, all these decisions can be evaluated together according to the simulation results. The candidate jobs will be selected based on these evaluations.

During the alternative simulation process, we conduct new simulation runs for the first decision point only (medium black circles). We name the simulation in the alternative simulation process nested alternative (small blocks). For the subsequent decision points (big white circles) in the alternative simulation, a simple base rule is adopted. The reasons why we only consider the first decision point are twofold: one reason is taking into account the long run time requirements of reduplicated simulation; the other one is considering the degree of decision influence, which means how much the performance of the current decision is influenced by the next direct decision. The influence from the later decisions reduces as the distance (time between decision points) increases. Due to the time consumption, we cannot start new simulation runs for all the decision points (small white circles) in the nested alternative simulation process. Therefore, for those decision points, instead, we use a simple base rule which is chosen based on the objective.

### 3.2 Branching Tree with a Time Axis

To depict the sequential relationships of all dispatching questions and to make the approach easier to understand, we use a branching tree which is quite suitable considering the hierarchy attribute to demonstrate the approach. For the sake of the decision points' time, we combine the branching tree with a time axis shown in Figure 3.

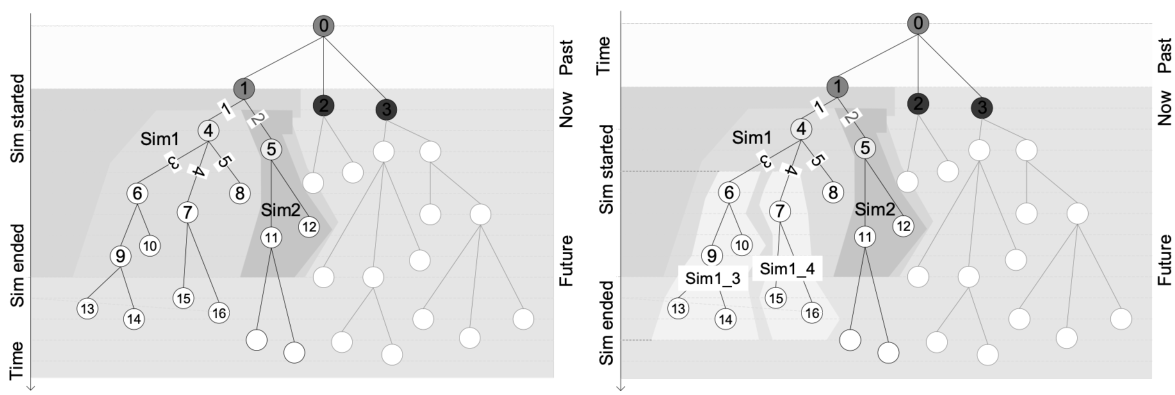


Figure 3: A branching tree with a time axis.

In this manner, a node denotes a decision point and a branch represents the selection of one job from all the waiting jobs. The corresponding value of one node on the time axis is the time when the decision is made. A path from the root node to the last leaf node is related to an active schedule. For the real complex job shop dispatching problem, as time goes on, it will be quite difficult to enumerate all the paths and then evaluate them due to the exponential explosion.

To make our approach easier to understand, we use an example to show how it works. In Figure 3, we assume that the current time is at decision point 1; job 1 and job 2 are two candidates. In order to decide which one is more appropriate, the alternative simulation Sim1 and Sim2 will start separately from this decision point and terminate at a given time. The Sim1 will simulate the future situation while job 1 is selected; the Sim2 handles the selection of job 2. The results of Sim1 and Sim2 will be used to evaluate the selection of job 1 and job 2. Obviously, in both Sim1 and Sim2, there are lots of decision points again. For the first decision points, we utilize a nested alternative simulation model to dispatch jobs, which is the biggest difference from the research (Zhang and Rose 2014) where they specify a base-rule to dispatch jobs instead of simulation. As is shown in the right picture of Figure 3, both Sim1_3 and Sim1_4 are the nested alternative simulation during the alternative simulation process of Sim1. Sim1_3 and Sim1_4 will simulate the future state of selecting job 3 and job 4 respectively. For the other decision points in the alternative simulation and for all decision points in Sim1_3 and Sim1_4, a simple base rule is adopted.

We assume that the selection of job 1 performs better, so job 1 is chosen at this decision point. When time advances to decision point 4, shown in Figure 4, we must select one job from three possible choices, thus alternative simulation Sim5, Sim6, and Sim7 will start. For the first decision points in Sim5, Sim6, and Sim7 nested alternative simulation is applied again. The remaining decisions will be made in the same way.

In summary, in order to evaluate the candidate job choices, alternative simulation is utilized to simulate the future states when one choice is made, and finally, the job will be selected according to the evaluation of the alternative simulation results. Nested alternative simulations are carried out for the first decision point during the alternative simulation process to substitute the simple base rule.

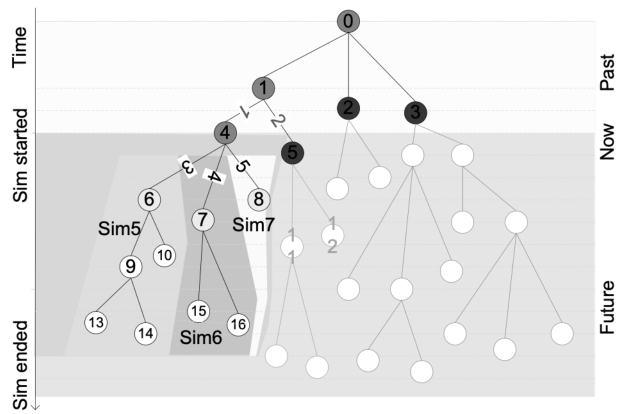


Figure 4: The branching tree after time advances.

### 3.3 Four-tuple and Key Issues

In this part, we give a more detailed description of our simulation approach for one decision point. A four-tuple is adopted to illustrate our method, M=(E, A, S, V). A represents alternatives, possible job choices in other words; S stands for the simulation which is called alternative simulation in our paper; V means the evaluation; E is the environment information at the decision point, namely job shop situation which includes conditions of all online jobs and machines. A machine's condition can be described by its state and the time this state lasts. In addition to its state and the time duration of the state, a job's condition encompasses its position and progress. The position describes which machine, buffer or transport line the job can be found on and the progress indicates at which step the job arrives. Furthermore, scheduled events, e.g., starting maintenance on one machine, are also concerned. These events will be directly put into the event list of the simulation and will occur during the simulation run.

Before we start the simulations, the simulation model is initialized with the environment information at the decision point. After the simulation initialization, in order to evaluate one candidate, a related "starting processing event" is put into the event list and the event occurring time is set to the current time. In this way, the simulation starts and will forecast what will happen in the following processes. For the simulation and the evaluation, there are several key problems which need to be fixed: 1) for each alternative job choice, how many times and how long the simulations should run; 2) how to evaluate the selection according to the simulation results.

The simulation run times for each candidate depends on the simulation model. If the simulation model is deterministic, the simulation just runs once. Otherwise, the simulation needs to run many times for stochastic models. Usually, the more the simulation runs, the better results we can obtain. Nonetheless, for the real-time dispatching in the job shops, the decisions should be made as soon as possible in order to maximize the utilization because the machines are free when we make the decision. On the other hand, for the sake of makespan, the decision should also be made immediately. The number of times can be calculated according to the time allowance in the real system. For the simulation duration problem, theoretically, the simulation ends when the effect of the current decision has disappeared, but it is hard to determine this time. In our research, the period of the simulation is decided by the number of completed jobs n after the decision point, as in Equation (1),

$$n = m + WIP / 2 \tag{1}$$

where m represents the number of completed jobs from the decision point to the time when all the candidate jobs at the decision point are totally finished. WIP is the work in process level at the decision point.

As to the evaluation problem, it is unquestionably that the evaluation criterion should be created on the basis of the specific dispatching objective. A priority value related to certain dispatching target is calculated for each job according to its simulation results. The job with the highest priority value will be selected. Here, we consider two formulas under two common objectives. When the objective is to minimize the cycle time, the priority value can be computed from Equation (2),

$$prio = 1 / \sum_{p \in P} (\omega_p \sum_{j \in J_p} c_j / n_{J_p}) \tag{2}$$

where $p$ denotes a product; $P$ is the set of products. $\omega_p$ is the weight of product $p$. $j$ is a job. $J_p$ is the set of completed jobs whose product type is $p$. $c_j$ is job $j$'s cycle time. $n_{J_p}$ is the number of jobs in the set $J_p$. When the objective is to minimize total weighted tardiness, the priority value can be computed from Equation (3),

$$prio = 1 / \sum_{p \in P} (\omega_p \sum_{j \in J_p} \max(C_j - d_j, 0) / n_{J_p}) \tag{3}$$

where $C_j$ is job $j$'s completion time and $d_j$ is job $j$'s due date. If the simulation runs many times, the average value of the priorities is adopted.

## 4 SIMULATION MODEL AND SIMULATOR

The simulation-based approach proposed above is applied to a sample simulation model. In this paper, we mainly focus on the application of simulation rather than modeling, so the simulation model used has not yet been mentioned until now. It is well known that the performance of the simulation approach highly depends on the built simulation model; hence in the following paragraph, some key requirements of our simulation model and simulator will be listed. For the details of the simulation model, please refer to our paper (Zhang and Rose 2012).

In order to simulate the manufacturing more precisely, uncertain events are considered in our simulation model. The processing time of jobs can be deterministic or stochastic. Machines need setup and break down randomly; the repairing time of the machines can be deterministic or random as well, and the preventive maintenance is also considered for the machines. Some common dispatching rules and release policies are included in our simulation model. For the simulator, it has been designed flexibly to load different kinds of job shop models and enable us to operate the event list in addition. Besides, the simulator can start from any initial states and start another new simulation run in a running simulation. Moreover, our simulator can run many times under the same initialization and terminate in the end condition which is specified according to some requirements. Finally, the simulator can output the results in a certain format for evaluation.

## 5 EXPERIMENT

In order to evaluate our nested-simulation-based dispatching approach, theoretically, we should test it online in a real manufactural environment. However, at present, it is impossible and also impractical for us to connect to the real system. Therefore, we create a simulation model to replace the real system. This simulation model is called environment simulation model in our paper which is always running like a real system. Once a decision needs to be made, the environment simulation will pause and wait for the decision. In the meanwhile, the decision-making procedure will start. The alternative simulation is initialized according to the current state of the environment simulation. When the decision-making process is finished, the environment simulation model will take this decision and continue. The performance of the simulation-based dispatching approach is gained from the results of the environment simulation.

In our environment simulation model, there are 6 machine groups, 24 machines and 4 kinds of products (Pa, Pb, Pc, and Pd) with 4 process flows and 4 variant outputs. There are no batch processing machines; interval times of releasing jobs follow the normal distribution; both sequence-dependent and independent setups are needed; the interval between two breakdowns is subject to the exponential distribution, and so is the repairing time. The objective of the dispatching is to minimize the cycle and the total weighted tardiness separately.

For experimental evaluation, we compare our nested-simulation-based approach (Nested SIM) with simple dispatching rules including first in first out (FIFO), operational due date (ODD), shortest processing time (SPT), and also SPT+ which is an improved version of SPT, and last with SIM which is the simulation based dispatching approach from our former research (Zhang and Rose 2014). In the SPT+ method, we set a maximal waiting time and use it to compare with each job's actual waiting time. If the actual waiting time of one job is longer, this job will be dispatched first so as to avoid the situation where the job with the long processing time has to wait too long. In order to testify the performance of the simulation approach fairly, the same base rule is used when it is needed in both the simulation-based approach and nested-simulation-based approach. The experiment results are shown in Table 1 with reference to average cycle time and Table 2 regarding total weighted tardiness.

Table 1: Average cycle time of jobs grouped by product type.

| Rule | Pa | Pb | Pc | Pd | Sum. | Time per Decision (s) |
|---|---|---|---|---|---|---|
| FIFO | 8.78 | 8.28 | 9.51 | 8.8 | 8.85 | 0.01 |
| SPT | 24.34 | 6.09 | 12.94 | 5.02 | 13.80 | 0.01 |
| SPT+ | 10.25 | 9.83 | 9.07 | 7.76 | 9.49 | 0.01 |
| SIM | 6.55 | 5.68 | 7.87 | 5.25 | 6.50 | 3.01 |
| Nested SIM | 6.12 | 5.32 | 6.98 | 5.22 | 6.02 | 93.24 |

To the best of our knowledge, SPT is an optimal rule for minimizing the cycle time in the single machine problem. However, in our experiments, from the results in Table 1, we can see that SPT performs worse than any other approaches for Pa and Pc. Because of the SPT+, the cycle time descends for Pa and Pc, but not for Pb and Pd. The reason is probably that both Pa and Pc have a longer raw processing time than Pb and Pd. Both simulation-based approaches perform better than the single rule, and nested-simulation-based approach gets the best results. Nonetheless, nested-simulation-based approach consumes the most time compared to all other four methods.

As it is presented in Table 2, considering the objective of the total weighted tardiness, the nested-simulation-based approach emerges to be the best approach again. Its performance is found to be significantly better than that of the other approaches, except for the run time.

In general, it can be seen that nested-simulation-based approach shows better results than all other tested methods for both objectives, but with more time consumption. The time consumption is due to the complication of the simulation model. The optimal performance is on account of the simulation technique which makes use of all possible information about the real manufacturing, the more simulation time is used, the better results can be achieved.

Table 2: Total weighted tardiness of jobs grouped by product type.

| Rule | Pa | Pb | Pc | Pd | Sum. | Time per Decision (s) |
|---|---|---|---|---|---|---|
| FIFO | 1.25 | 0.28 | 1.34 | 0.32 | 0.70 | 0.01 |
| ODD | 0.03 | 0.00 | 0.08 | 0.05 | 0.04 | 0.01 |
| SIM | 0.04 | 0.00 | 0.07 | 0.02 | 0.04 | 3.39 |
| Nested SIM | 0.02 | 0.00 | 0.07 | 0.00 | 0.03 | 101.02 |

## 6 CONCLUSION

In this article, the nested-simulation-based method is proposed to deal with the real-time dispatching problem online in the manufactural environment. Different from the general simulation-based dispatching method in which the simulation is used to evaluate the simple dispatching rules offline and the rule with the best simulation result will be chosen, in our method the simulation runs for each alternative job, the job with the best outcome will be selected and all the decisions are made online. During the simulation process for each alternative job, nested simulations are carried out for the first decision points and the based rule is used for the other decision points. In the nested simulation process, we use the base rule to dispatch job. Through plenty of experiments, our approach has demonstrated that it is superior to any other pure simple rule, even better than the method which also uses the simulation-based approach in our former research. The better performance is attributed to the utilization of the simulation which has the ability to imitate all the possibilities in the real production environment. The flaw of our approach is the long run time, even though sometimes it is reasonable to consume a little bit more time to achieve better results. However, considering our approach should be utilized for online decision making in real factories, it is absolutely necessary to optimize the run time of our approach within the acceptable range. Consequently, in the future, we are going to find ways to speed up our simulation so that the proposed method can perform more efficiently.

## REFERENCES

Adibi, M. A., M. Zandieh and M. Amiri. 2010. “Multi-objective Scheduling of Dynamic Job Shop Using Variable Neighborhood Search”. *Expert Systems with Applications* 37:282-287.

Aydin, M. and E. Öztemel. 2000. “Dynamic Job-shop Scheduling Using Reinforcement Learning Agents”. *Robotics and Autonomous Systems* 33(2-3):169-178.

Banu, C. And S. Bulkan. 2015. “A research Survey: Review of AI Solution Strategies of Job Shop Scheduling Problem”. *Journal of Intelligent Manufacturing* 26:961-973.

Littman, M. L.. 1996. *Algorithms for Sequential Decision Making*, Rhode Island: Brown University.

Malikppoulos, A. A., P. Y. Papalambros, and D. N. Assanis. 2009. “A Real-time Computational Learning Model for Sequential Decision-making Problems under Uncertainty”. *Journal of Dynamic Systems, Measurement, and Control* 131(041010):1-8.

Rajendran, C. and O. Holthau. 1999. “A Comparative Study of Dispatching Rules in Dynamic Flow Shops and Job Shops”. *European Journal of Operational Research* 116(1):156-170.

Xiong, H., H. Fan, G. Jiang, and G. Li. 2017. “A Simulation-based Study of Dispatching rules in a Dynamic Job Shop Scheduling Problem with Batch Release and Extended Technical Precedence Constraints”. *European Journal of Operational Research* 257(1):13-24.

Yih, Y. and A. Thesen. 1991. “Semi-Markov Decision Models for Real-time Scheduling”. *International Journal of Production Research* 29(11):2331-2346.
Zhang, T. and O. Rose. 2014. “Simulation-based Dispatching in Job Shops”. In *Proceeding of the Symposium Simulationstechnik*, September 3th-5th, Berlin, Germany, 255-258.
Zhang, T. and O. Rose. 2012. “Developing an Agent-oriented Parallel Simulator for Production Processes”. In *Proceedings of the 2012 Winter Simulation Conference*, edited by C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A.M. Uhrmacher. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
Zhang, T. and O. Rose. 2013. “Intelligent Dispatching in a Dynamic Stochastic Job Shops”. In *Proceedings of the 2013 Winter Simulation Conference*, edited by R. Pasupathy, S.-H. Kim, A. Tolk, R. Hill, and M. E. Kuhl, 2622-2632. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

## AUTHOR BIOGRAPHIES

**SHUFANG XIE** is a Research Assistant and Ph.D. student at Universität der Bundeswehr at the Chair of Modeling and Simulation. Her focus is on simulation-based scheduling and optimization of production systems. She has received her M.S. degree in Metallurgical Engineering from Chongqing University, China. Her email address is shufang.xie@unibw.de.

**TAO ZHANG** is a Research Assistant at the Universität der Bundeswehr München, and he holds an M.S. in Metallurgical Engineering from Chongqing University, China and a Ph.D. in Computer Science from the Universität der Bundeswehr München, Germany .His research interest is working on production planning and scheduling, the main focus of his research is on modeling and simulation of complex systems and intelligent optimization algorithms. His email address is tao.zhang@unibw.de.

**OLIVER ROSE** holds the Chair for Modeling and Simulation at the Department of Computer Science of the Universität der Bundeswehr, Germany. He received an M.S. degree in applied mathematics and a Ph.D. degree in computer science from Würzburg University, Germany. His research focuses on the operational modeling, analysis and material flow control of complex manufacturing facilities semiconductor factories. He is a member of INFORMS Simulation Society, ASIM, and GI. His email address is oliver.rose@unibw.de.