A LEARNING AGENT FOR A MULTI-AGENT SYSTEM FOR PROJECT SCHEDULING IN CONSTRUCTION

Florian Wenzler
Willibald A. Günthner
Lehrstuhl für Fördertechnik Materialfluss Logistik
Technische Universität München
Boltzmannstraße 15, 85748 Garching, Germany
Email: wenzler@fml.mw.tum.de, guenthner@fml.mw.tum.de

KEYWORDS
MRCPSP; Project Scheduling, Multi-agent system, Discrete event simulation

ABSTRACT

The quality of the project plan created is essential for realizing a construction project. This is a big challenge for planners, because there are many constraints to be considered. The problem to be solved is known as the multi-mode resource-constrained project scheduling problem (MRCPSP). This paper presents a multi-agent approach in which resources and processes are represented as collaborative agents. Autonomous process and resource agents register themselves on a central blackboard where resource allocation to activities is negotiated. As expansion to prior works, a learning agent is integrated to improve the solutions created. A discrete-event simulation implements the model and it is evaluated with standardized project plans from the field of operations research.

INTRODUCTION

Insufficient construction-project planning often leads to overall progress delays and cost overruns. Most projects are nonetheless scheduled manually without using optimization tools. Hence, quality depends on the planner's experience and the available time. That's why project plans are often generated without much detail or consideration of constraints, which are primarily predecessor/successor dependencies, and limited resources and space.

The influence of unpredictable circumstances is also important for project management in construction. The former lead to delays and necessitate rescheduling, but the effect of delayed processes cannot be investigated in advance in detail without a computer-based tool.

Most project scheduling software uses methods such as Program Evaluation and Review Technique (PERT) or Critical Path Method (CPM) (Maroto and Tormos 1994). None of those methods considers resource constrains. A method for project scheduling in construction dealing with these topics therefore has to be developed, which

- considers all types of constraints in construction,
- is adaptable to specific situations, and
- enables easy rescheduling after unpredictable incidences.

PROBLEM STATEMENT

The general problem to be solved is known as the resource-constrained project scheduling problem (RCPSP) or the multi-mode resource-constrained project scheduling problem (MRCPSP), because every activity can be executed in different ways (modes). These modes' process time and required resources differ. The problem can be described generally as follows:

J: number of activities/jobs
j: activity ID with j = {0, …, J+1}
M: number of modes for each activity j
d_{jm}: duration of activity j executed in mode $m \in M$
S_j: successors of activity j
P_j: predecessors of activity j
R: number of different types of renewable resources
N: number of different types of nonrenewable resources
r_{jmk}: renewable resources of type $k \in R$ required by activity j in mode m
n_{jml}: nonrenewable resources of type $l \in M$ required by activity j in mode m

Jobs having IDs 0 and J+1 are dummy activities with neither a processing time nor resource requirements ($d_{0\ m|J+1\ m}=0$, $r_{0\ mk|J+1\ mk}=0$, and $n_{0\ ml|J+1\ ml}=0$). They serve as the project's start and end.

Minimizing a project's makespan while taking care of the given constraints is the goal. The following variances can be used to achieve this: first is the mode in which an activity executes; second is each process's starting time. That the starting time's predecessor/successor dependencies aren't violated has to be guaranteed.

The number of resources used in the project plan created is never allowed to exceed the given number of renewable and nonrenewable resources. The chosen solution is invalid if it does so.

Proceedings 30th European Conference on Modelling and Simulation ©ECMS Thorsten Claus, Frank Herrmann, Michael Manitz, Oliver Rose (Editors)
ISBN: 978-0-9932440-2-5 / ISBN: 978-0-9932440-3-2 (CD)

Some simplifications have to be made for upcoming parts of the paper:

- The execution of started activities cannot be interrupted.
- An activity's chosen mode cannot be subsequently changed.
- An activity's resources remain assigned until the job is finished.
- The number of available resources cannot be changed during the project time.

However for the intended use in construction, these restrictions have no big influence or can be considered by adjusting the input data (e.g., splitting an activity up into two or more parts with individual features).

Different schedules are needed during development and for tests. Kolisch and Sprecher created standardized examples for this purpose with their project generator, ProGen (Kolisch and Sprecher 1997). The plans fulfill all constrains mentioned and are built up systematically for selected parameters as Table 1 shows. A particular parameter is changed for every type of plan while the rest remain fixed. This allows selective investigation of each parameter's influence on the result.

Table 1: Structure of the Project Plans Used

Name	Parameter			
	J	M	R	N
j10	10	3	2	2
j16	16	3	2	2
j30	30	3	2	2
m1	16	1	2	2
m5	16	5	2	2
n0	10–20	3	2	0
n3	16	3	2	3
r1	16	3	1	2
r5	16	3	5	2

Knowledge of the minimal project duration is an essential advantage of the instances that Kolisch and Sprecher created. The quality of the method used to solve the MRCPSP can thus be compared and evaluated.

DIFFERENT APPROACHES TO SOLVING THE RCPSP AND MRCPSP

Blazewicz et al. proved that this problem is np-hard (Blazewicz et al. 1983). An optimal solution is hence nearly impossible to find within a reasonable amount of time. For smaller projects, approaches such as branch-and-bound (Johnson 1967) or lower bounds (Heilmann and Schwindt 1997) can be used to find the optimum. However, the solution space grows very fast with larger projects and these approaches become inefficient. That's why various heuristics and meta-heuristics were developed and adapted for the (M)RCPSP. Among these are simulated annealing (König and Beißert 2009, Józefowska et al. 2001), genetic algorithms (van Peteghem and Vanhoucke 2010, Senouci and Al-Derham 2008, Toklu 2002), ant colony algorithms (Li and Zhang 2013, Christodoulou 2005), or particle swarm optimization (Jarboui et al. 2008, Lu et al. 2008, Zhang et al. 2006). Despite their different basic ideas, they all create new combinations according to different rules, but also randomly, to try to find a better solution.

Since creating every possible solution within an acceptable time isn't possible, finding the optimal solution it is not guaranteed.

MULTI-AGENT SYSTEM FOR THE MRCPSP

Using a Multi-Agent System (MAS) is a different approach to solving this problem. The main benefit is being able to split the whole problem into smaller, easier parts. Furthermore, the latter are more robust and flexible than those in traditional methods (Davidsson et al. 1994). The agents themselves are also easy to understand and create due to the small number of capabilities.

A few MAS implementations exist for the resource-constrained scheduling problem. Horenburg presented a MAS for the RCPSP with agents for each activity as well as for each resource. Resource allocation to jobs is controlled by priority rules (Horenburg 2014). Knotts et al. introduced another agent-based framework for solving the RCPSP in minimal project duration (Knotts et al. 2000). Resources aren’t modeled as agents in this case.

Wauters et al. implemented a new aspect with the system's ability to learn (Wauters et al. 2011). Selecting the next activity is realized in two steps for solving the MRCPSP. The most important job is first identified, then one of its modes is chosen. This means consequently that not all modes of the activities have the same chance to get executed. If the mode of an activity with the highest priority cannot be executed, it is not possible to select a mode from an alternative activity, although it might be a better choice than the next mode from the previously chosen activity. With this issue deals the following presented MAS by including all possible modes in the process of resource allocation. Hence, only one step is needed for choosing the next mode of an activity and there might be potential for improvements of a Multi-Agent System.

Framework of the MAS

This section will present the structure of a multi-agent system for the MRCPSP. Figure 1 shows the different types of agents and communication. Different types of agents represent activities as well as renewable and nonrenewable resources. The central element is the blackboard. Resource allocation to current activities is negotiated there. This architecture simplifies communication (compared to the complexity required when all agents have to communicate with each other) and promotes efficient, transparent resource allocation.

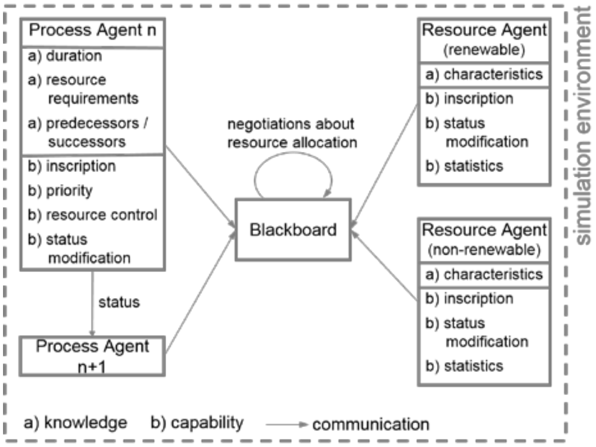

Figure 1: Multi-Agent System

All of the jobs' executable modes register themselves on the blackboard with resource requests as soon as their predecessors finish. All available resources are simultaneously inscribed there too. Every time a job finishes, this agent informs all the former's successors about the completion.

Information about duration, resource requirements, and previous/subsequent activities enables every process agent to act in the described way. Beside these characteristics, every agent can act on and communicate with its environment. As already mentioned, the most important capability is being able to register on the blackboard with the calculated priority value. Methods also exist for adjusting status and recording data for statistics.

Priority Rules for Resource Allocation

Insufficient resources typically exist for all of activities registered on the blackboard. To identify the most crucial current jobs, each process agent transmits a priority value. The activities' negotiation order is calculated based on this value. Whether enough resources are available at the moment is successively checked for each. If not, an activity is postponed until the next negotiation round. That the limits of simultaneously active renewable resources are never exceeded can be guaranteed this way.

The situation is different with nonrenewable resources. Subsequent activities cannot be started once the limit is reached, and the search for a solution stops prematurely. Due to the way the project plan is created, a valid combination of modes is not guaranteed. That an early negotiation can cause too many nonrenewable resources to be used is unavoidable with local decisions. The project is nevertheless planned completely for getting an (invalid) starting combination, which can be improved later.

Different priority rules for the MAS introduced to solve the MRCPSP were presented in a previous paper (Wenzler and Günthner 2015). They feature different activity attributes to compute the priority value such as duration, resource requirements, or number of successors.

The LPF_AVG (Longest Path Following) rule is chosen in the sequel. This was shown to provide - together with others - the best results and is defined as follows: Every activity determines the duration of its successor processes. The activity with the biggest value receives the highest priority. Since priority calculation occurs before or during project planning itself, the longest path has to be identified without resources. Appendix "_AVG" defines how to handle the different modes of every activity in the path. Every activity can be executed in only one mode, but which one will be chosen is unknown in advance. So the average of all modes is assumed for an activity's duration in this priority rule.

Introducing a Learning Agent

As mentioned, the first simulation run may be unable to find the optimal solution. That's why a new agent type, the learning agent (LA), was incorporated into the existing framework. Figure 2 shows its communication with other agents.

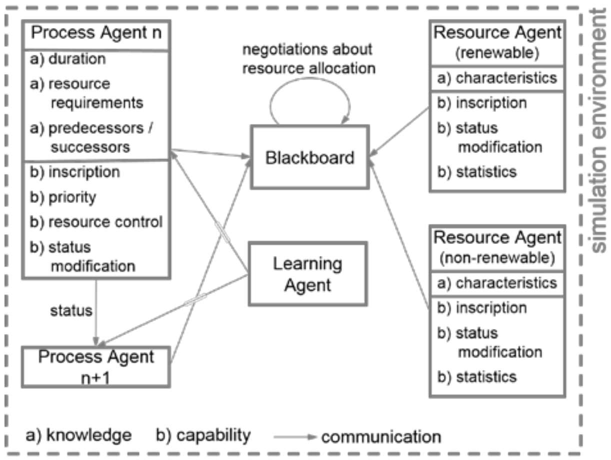

Figure 2: Multi-Agent System with Learning Agent

The learning agent can analyze the plan created so far and influence the process agents' mode choices. The LA subsequently restarts the planning procedure and compares the result with previous solutions.

THE LEARNING AGENT'S FUNCTIONALITY

The learning agent (LA) has two main tasks, which are executed in the order listed:

- Create a feasible solution that doesn't exceed resource limits.

- Improve a feasible solution as far as possible.

The LA is active until a stop criterion is satisfied. This can happen in several different ways:
- The optimum is found. Therefore, the best solution has to be known. Projects are usually so large that the optimum cannot be determined. For plans from the PSPLIB, which are used in this paper, the minimum makespan for each schedule is known and the LA can use this knowledge.
- One type of optimization rule is used successively more often than a defined limit.
- No improvements are made for too many consecutive times.

With the last two rules, the calculation will terminate whenever the LA cannot improve the solution with the defined settings.

Creating Feasible Solutions

A valid solution - even one with a longer makespan - is better than exceeding the constrains. Hence, the LA's first task is to generate a feasible plan.

The heuristic of the learning agent to get an acceptable solution operates as follows: An (invalid) solution is needed first. Then each activity's modes are cycled through to check for possible nonrenewable-resource improvements. The LA obeys the following rules to do this:
- The requirement for at least one type of nonrenewable resource is less stringent in the new mode than in the current one.
- The amounts of other types of resources used must not grow - unless enough reserve exists.
- The limits imposed on other types of resource may not be exceeded.

The result of this procedure depends on the initial solution. A new combination of modes will be chosen if a feasible solution was not generated. The mode with the highest savings is selected to avoid the previous bottleneck for the crucial type of nonrenewable resources.

As soon as a feasible solution is found, the LA transmits the defined modes to the process agents and the process of creating a schedule is started again.

Solution Improvement

Any feasible solution generated is unlikely to be the optimal solution. The LA is hence tasked with improving it via selective adjustments. Changing the mode, the earliest starting time, and the priority rule are possible adjustments.
One important solution analysis tool is identification of the critical path and the floats. If the value of an activity's float is greater than 0, this activity can be delayed up to this value without having any influence on the remaining schedule. Every activity with zero float is part of the critical path. With this information, searching the activities the adjustment of which probably effects the schedule most is possible. The following rules are implemented in the current state of research:
- Change some activities to a mode with shorter duration. The number of selected activities can vary. For every activity the ratio of duration to resource requirement is calculated and for those activities, whose value is above the average, a new mode is chosen.
- Shift some of the activities to a later start time in case they have enough float. The freed resources may allow other activities to start earlier.
- Execute some activities in a mode with less demanding resource requirements so other jobs can use more resources or start earlier. Chosen are those activities which save more resources than the average by changing the mode.

At this point, no rule uses random for changing the parameters. For that reason, every decision is understandable.

RESULTS

The MAS presented was implemented in a discrete-event simulation (DES). Monte-Carlo simulations were conducted to verify and validate the model as well as to provide reference values. Priority values are therefore generated randomly. Several different priority rules were evaluated in the next step (Wenzler and Günthner 2015). The "LPF_AVG" rule produced the best results so this rule was chosen in this paper (labeled "without LA" in tables or figures).

Comparison with the MAS without LA

This section will present the effect of activating the learning agent on the simulation results. The first goal for which the learning agent is implemented is to reduce the number of invalid schedules. Table 2 lists the number of projects for which no feasible solution was found.

Table 2: Number of Infeasible Projects

Type	Total number of projects	Infeasible solutions	
		Without LA	With LA
j10	536	315	0
j16	550	308	0
j30	552	300	0
m1	640	0	0
m5	558	309	4
n0	470	0	0
n3	600	372	15
r1	553	306	0
r5	546	286	0

Without the active LA for every type of plan, the MAS left a number of projects unsolved. The ratio is up to 62% except for m1 and n0, where all plans are solvable because of their structure.

Table 2 shows that a valid schedule was created using the LA for almost every project. The only exceptions are the most complex plans, m5 and n3, with 4 and 15 unsolved plans respectively. All possible combinations of modes for each activity were searched by enumeration to further investigation of why the MAS with LA still cannot solve some of the projects (Table 3). The "Feasible combinations" column represents the number of different combinations that can be created without exceeding nonrenewable-resources limits.

Table 3: Number of Feasible Combinations of the Unsolved Schedules

Type	Number	Feasible combinations	Possible combinations
n3	1_6	1	43 046 721
	3_3	189	43 046 721
	3_6	27	43 046 721
	6_3	4	43 046 721
	6_4	6	43 046 721
	6_6	1	43 046 721
	6_7	4	43 046 721
	6_8	4	43 046 721
	7_7	4	43 046 721
	8_2	16	43 046 721
	8_4	4	43 046 721
	36_7	18	43 046 721
	36_8	124	43 046 721
	36_9	1881	43 046 721
	36_10	2	43 046 721
m5	1_1	4	1 440 000 000
	1_2	2	35 156 250 000
	5_4	256	152 587 890 625
	36_9	1104	152 587 890 625

The results show that are only a few possible ways exist to get a feasible schedule. Sometimes the heuristic has to find the single way out of more than $43x10^6$ possibilities, as in case of the n3 plans, or one of two solutions from $35x10^9$ combinations theoretically possible for project-type m5.

The heuristics for solving these projects correctly have to be improved in future work. Integrating enumeration is not an option because of excessive computing time especially for large projects.

Table 4 shows the results of the LA's second task: solution improvement. The number of optimal solutions increased only slightly with the defined stop criteria for plan types j16, n3, and r1. However, a lot of the remaining projects finished within a shorter time.

Table 4: Project-Makespan Improvement

Type	Optimal solutions		Better solutions with LA
	without LA	with LA	
j10	112	112	351
j16	112	113	360
j30	116	116	366
m1	400	400	0
m5	96	96	305
n0	231	231	88
n3	107	110	393
r1	136	137	337
r5	136	136	353

The figures below show detailed results for some project types. The number of tested schedules having a certain deviation from the known optimum can be seen there. The bar with "0" deviation represents the optimal solutions, while the declared value of time units is also needed for completion of the other plans. The last bar shows the number of infeasible solutions if any exist.

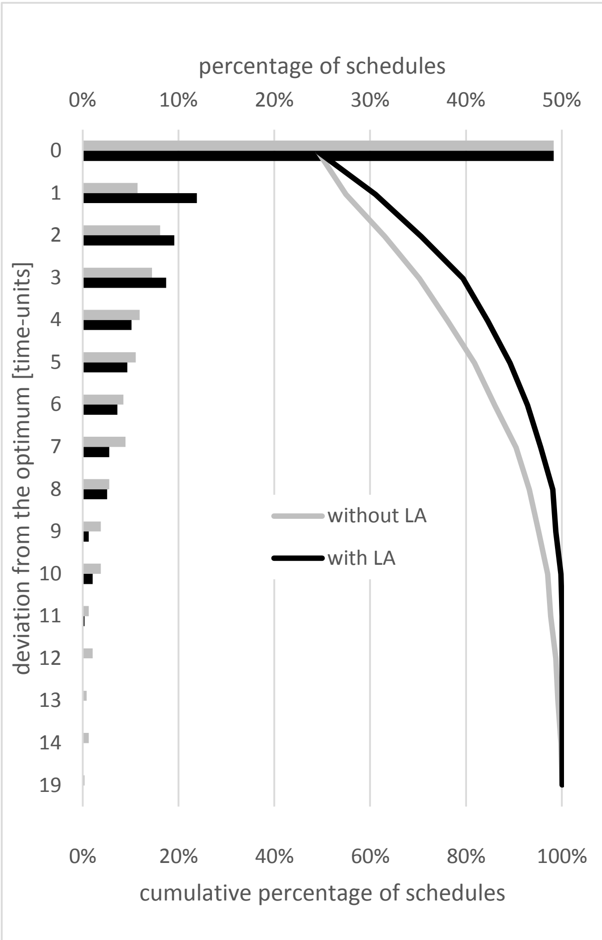

Figure 3: Detailed Results for Project Type n0

Nearly 50% of the plans specified n0 can be solved optimally with the MAS (Figure 3). This can be achieved even without the LA; however, the other results improve with the active LA. The largest deviation is reducible from 19 to 11 time units. The number of nearly perfect solutions with deviations of 1 to 3 also rose significantly.

The n3 projects' greater complexity is visible in Figure 4. More than 60% irregular schedules exist without the LA. In contrast, only 15 unsolved projects remain with the LA. The number of optimal solutions or of those with small deviations from the optimum also increased. The main point for further improvements is the large number of schedules for which the heuristics found a feasible but not optimal solution.

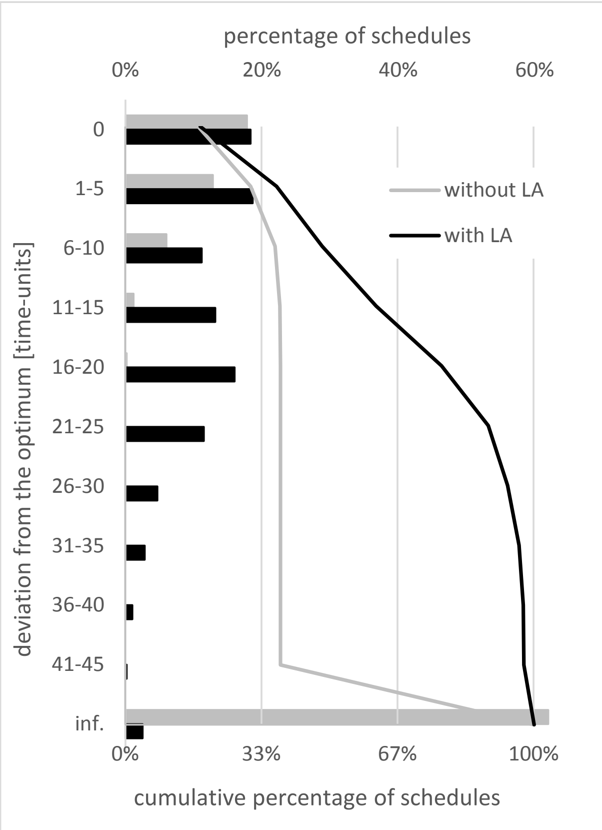

Figure 4: Detailed Results for Project Type n3

Comparison with other Approaches

In Table 5 the results of the MAS with the priority rule “LPF_AVG” for the datasets with 10 to 20 activities (j10-j20) are shown. For those, the comparison with other approaches is possible. The table shows the average deviation from the known optimal solution. The actual performance of the MAS is not as good as those of the alternative methods. This can be explained by the following issues. Firstly, the LA creates feasible solutions without giving the project duration top priority. These solutions have in general a large makespan (up to 200% of the optimal duration) and so even a few solutions with a long duration have a strong influence on the average deviation. To solve this problem, the LA has to improve the initial solution by changing some parameters. In the current state, only the mentioned rudimental rules are implemented and after about 10 iterations the solutions aren’t changing anymore. Therefore the heuristics have to be improved to create a larger solution space.

The positive aspect of the actual results is, that the average deviation is nearly constant, although the size of the datasets increases.

CONCLUSION

This paper presents a multi-agent approach to solving the MRCPSP. An individual collaborative agent, a new type of which (learning agent) was introduced, represents every activity and resource. It analyses a previously generated solution and influences the process agents' decisions concerning the chosen mode using the dependent resource requirements or the starting time.

The MAS is implemented in a discrete event simulation environment and tested with standardized projects from the field of operations research. Hence, these projects' optimal solution is known, and the quality of the project plans created could be evaluated.

With the learning agent (LA), the high quota of irregular project plans can be reduced significantly and the number of (nearly) optimal solutions increased compared to the MAS without learning agent.

The presented MAS is a preliminary result. The learning agent has to be improved further to create better solutions with as little rescheduling as possible by the end of the project.

Table 5: Comparison with other Approaches for the MRCPSP – Average Deviation from Optimum (%)

	j10	j12	j14	j16	j18	j20
LPF AVG	41.95	41.44	43.05	44.45	45.16	45.48
Li and Zhang (2013)	0.09	0.13	0.40	0.57	1.02	1.10
Wauters et al. (2011)	0.05	0.08	0.23	0.30	0.53	0.70
Van Peteghem and Vanhoucke (2010)	0.01	0.09	0.22	0.32	0.42	0.57
Jarboui et al. (2008)	0.03	0.09	0.36	0.44	0.89	1.10
Józefowska et al. (2001)	1.16	1.73	2.60	4.07	5.52	6.74

Some specific additions have to be made for use in construction. First, a new area agent allows the limited space on a construction to be taken into account site. That leads to new constraints, which have to be considered.

A type of resource agent for shared resources is missing. Several processes use some machines simultaneously (e.g., cranes) necessitating another agent. Include the emerging interactions between participating activities in the MAS is possible this way.

Finally, real project data will be used to demonstrate applicability.

ACKNOWLEDGMENTS

The research presented in this paper is part of the project “Entwicklung einer agentenbasierten Methodik zur Terminplanoptimierung im Bauwesen unter Berücksichtigung ressourcenabhängiger Prozesslängen” (GU 427/22-1), which is funded by the German Research Foundation (DFG).

REFERENCES

Blazewicz, J.; Lenstra, J. and Rinnooy Kann, A. H. G. 1983. "Scheduling subject to resource constraints: Classification and complexity." *Discrete Applied Mathematics* 5, No. 1 (Jan), 11–24.

Christodoulou, S. 2005. "Ant Colony Optimization on Construction Scheduling.“ In *Proceedings of the International Conference on Computing in civil engineering 2005* (Cancun, Mexico, July 12–15), American Society of Civil Engineering.

Davidsson, P.; Astor. E. and Ekdahl, B. 1994. "A framework for autonomous agents based on the concept of anticipatory systems." In *Proceedings of Cybernetics and Systems ’94*. World Scientific, Singapore, 1427–1434.

Heilmann, R. and Schwindt, C. 1997. "Lower Bounds for RCPSP/max." Report WIOR-511. Universität Karlsruhe.

Horenburg, T. 2014. *Simulationsgestützte Ablaufplanung unter Berücksichtigung aktueller Baufortschrittsinformationen*. Dissertation Technische Universität München, Lehrstuhl für Fördertechnik Materialfluss Logistik.

Jarboui, B.; Damak, N.; Siarry, P. and Rebai, A. 2008. "A combinatorial particle swarm optimization for solving multi-mode resource-constrained project scheduling problems." *Applied Mathematics and Computation* 195, No. 1 (Jan), 299–308.

Johnson, T.J.R. 1967. *An algorithm for the resource-constrained project scheduling problem*. Ph.D. Dissertation. MIT.

Józefowska, J.; Mika, M.; Rózycki, R.; Waligóra, G. and Weglarz, J. 2001. "Simulated annealing for the multi-mode resource-constrained project scheduling. " *Annals of Operations Research* 102, No. 1 (Feb), 137–155.

Knotts, G; Dror, M. and Hartmann, B. C. 2000. "Agent Based project scheduling." *IIE Transactions* 32, No. 5 (May), 387–401.

Kolisch, R. and A. Sprecher. 1997. "PSPLIB – A project scheduling problem library." *European Journal of Operational Research* 96, No. 1 (Jan), 205–216.

König, M. and Beißert, U. 2009. "Construction Scheduling Optimization by Simulated Annealing." In *Proceedings of the 26th Annual International Symposium on Automation and Robotics in Construction* (Austin, TX, June 24–27). International Association for Automation and Robotics in Construction, 183–190.

Li, H. and Zhang, H. 2013. "Ant colony optimization-based multi-mode scheduling under renewable and nonrenewable resource constraints." In *Automation in Construction* 35 (Aug), 431–438.

Lu, M.; Lam, H.-C. and Dai, F. 2008. "Resource-constrained critical path analysis based on discrete event simulation and particle swarm optimization." In *Automation in Construction* 17, No. 6 (Nov), 670–681.

Maroto, C. and Tormos, P. 1994. "Project Management: an Evaluation of Software Quality". In *International Transactions in Operational Research*, 1, No. 2 (Apr), 209–221.

Senouci, A. and Al-Derham, H. R. 2008. "Genetic algorithm-based multi-objective model for scheduling of linear construction projects." In *Advances in Engineering Software* 39, No. 12 (Dec), 1023–1028.

Toklu, Y.C. 2002. "Application of genetic algorithms to construction scheduling with or without resource constraints." In *Canadian Journal of Civil Engineering* 29, No. 3 (Jun), 421–429.

Van Peteghem, V. and Vanhoucke, M. 2010. "A genetic algorithm for the preemptive and non-preemptive multi-mode resource-constrained project scheduling problem." European Journal of Operational Research 201, No. 2 (Mar), 409–418.

Wauters, T.; Verbeeck, K.; Vanden Berghe, G. and De Causemaecker, P. 2011. "Learning agents for a multi-mode project scheduling problem." *Journal of the Operational Research Society* 62, 2 (Feb), 281–290.

Wenzler, F. and W. A. Günthner. 2015. "Ressourcen-beschränkte Terminplanung mit einem System kollaborativer Agenten." In *Simulation in Production and Logistics 2015*, M. Rabe and U. Clausen (Eds.). Fraunhofer, Stuttgart, 721–730.

Zhang, H.; Tam, C. M. and Li, H. 2006. "Multimode Project Scheduling Based on Particle Swarm Optimization." In: *Computer-Aided Civil and Infrastructure Engineering* 21, No. 2 (Feb), 93–103.

FLORIAN WENZLER studied Mechanical Engineering with concentration on automotive and power train technologies at the Technische Universität München (TUM). He obtained his degree in 2013 and has been working since then as a research assistant at the Institute for Materials Handling, Material Flow, Logistics (fml). His work focuses on operations research and the discrete event simulation of logistic systems. His email address is: wenzler@fml.mw.tum.de.

WILLIBALD A. GÜNTHNER is professor and head of the Institute for Materials Handling, Material flow, Logistics (fml) at the Technische Universität München (TUM) He is cofounder of the Wissenschaftliche Gesellschaft für Technische Logistik e. V. and advisory board member of several associations, federations, and companies.