Resource Allocation in Construction Scheduling based on Multi-Agent Negotiation

T. Horenburg, J. Wimmer & W. A. Günthner
Institute for Materials Handling, Material Flow, Logistics, Technische Universität München, Germany

Abstract

Computerized project management in construction is traditionally based on precedence diagrams such as PERT or CPM. Resources required for the execution of activities are usually scarce and therefore provide additional constraints to the predecessor-successor relationships. The objective of this NP-hard problem, which is also known as Resource Constrained Project Scheduling Problem (RCPSP), is to minimize the makespan considering all existing constraints.

Since larger project plans in construction contain thousands of individual activities, an exact solution of the latter is hardly possible within a reasonable amount of time. For this reason various approaches such as branch-and-bound methods or heuristics like simulated annealing, particle swarm optimization and genetic algorithms were applied. The procedures are based on the determination of processes, where resources are available and predecessors completed. Hence resources are allocated immediately after the completion of prior tasks and will not be reserved for more critical activities.

In contrast to prior works in Multi-Agent-Systems (MAS) the approach introduced in this paper uses agent-technology both for processes and resources. Autonomous process agents register their activities on a central blackboard, where all agents may negotiate on resource allocation depending on the available proposals offered by resource agents. The latter calculate their qualification for the corresponding activity using implemented utility functions. Hence resources are allocated to critical processes and identical resources are distributed to support ideal construction flow.

This concept for agent based project scheduling is implemented in Discrete-Event-Simulation (DES) and evaluated by multiple experiments. Therefore the quality of results were validated using standardized project plans from the field of Operations Research (OR). Additionally the results are compared with the results from Monte-Carlo-Simulations integrated in the same system, so that quality and computing time could be evaluated. The overall concept was proven as stable and reliable for the scheduling of construction projects.

Keywords: Project Scheduling, RCPSP, Multi-Agent-System, Discrete Event Simulation

1 Introduction and Related Work

The planning of construction projects is often of little detail resulting in high coordination effort, low productivity rates and delays in overall progress. Therefore precedence diagrams in its different specifications such as PERT or CPM provide a basis for computerized project management (Maroto and Tormos,1994). Resources required for the execution of activities are usually scarce and consequently provide additional constraints to the predecessor-successor relationships. In research this

NP-hard problem is known as Resource Constrained Project Scheduling Problem (RCPSP) (Brucker et al., 1999). Various approaches such as branch-and-bound or lower bounds (Johnson, 1967; Heilmann and Schwindt, 1997) methods were applied. Yet an exact solution is hardly possible within a reasonable amount of time. To increase the ratio of good results different heuristics and meta heuristics were developed and customized to meet the specific requirements of the RCPSP.

The procedures are usually based on the determination of executable processes at each time step depending on available resources and predecessors. Tasks executed are chosen randomly (constraint-based Monte-Carlo-Simulation) (König et al, 2007) or based on soft constraints such as execution strategies (Beißert et al. 2008). Further approaches for the allocation of resources consider particle swarm optimization (Lu et al., 2008; Zhang et al., 2006), simulated annealing (König und Beißert, 2009), genetic algorithms (Senouci and Al-Derham, 2008; Toklu, 2002) and ant colony optimization algorithms (Christodoulou, 2005). Knotts et al however introduce an agent-based system for project scheduling, where several priority rules determine the results in terms of minimum project duration (Knotts et al., 2000). Since resources are not modeled as agents, a differentiation of individual resource entities is hardly possible. Yet the environment of construction requires this handling as such. Especially the qualification of an individual agent for a specified task is vital for an efficient processing of construction works.

All approaches to determine an optimal or at least feasible execution sequence have in common that resources are allocated immediately after the completion of prior tasks. This implies that resources will not be reserved for potentially more relevant processes.

The general application of agent-based methods on problems in construction is quite common. These approaches do rarely discuss the minimization of project schedules and should therefore not be further evaluated within this publication.

The Resource Constrained Project Scheduling Problem (RCPSP) as such can be specified according to Brucker's et al notation (Brucker et al., 1999) and adapted to an agent-based framework for construction projects as follows:

- set of activities V Є {0,1…n+1}
- set of renewable resources R Є {1,2…m}
- set of non-renewable resources M Є {1,2…d}
- activity/job j Є [V] with duration $p_j>0$
- renewable resources k Є [R] with amount of available resources R_k
- non-renewable resources/material h Є [M] with amount M_h
- renewable resources r_{jk} required by activity j from resource k
- non-renewable resources h_{jk} required by activity j from resource h
- set of predecessors P_j of activity j
- set of successors S_j of activity j
- capital c_j of activity j

An additional dummy activity is implemented each for the start and the end of the project plan, requiring neither resources nor time ($p_{0\,/\,n+1} = 0$; $r_{0k\,/\,n+1k} = 0$). In operational research this problem is denoted as PS | prec | C_{max} (Brucker et al., 1999) or m,1 | cpm | C_{max} (Demeulemeester and Herroelen, 2002). Typically the objective is to minimize the project's makespan considering scarce resources as well as all predecessor and successor relations. Result is an optimized schedule containing start and completion times of every activity and respective resource allocations.

Agent-based systems usually are more robust and flexible than traditional approaches (Davidsson et al., 1994). The technology as such is an excellent solution for complex systems. Yet agents themselves are quite simple in nature and therefore easy to understand and compute (Jennings and Wooldridge, 1995). Thus a new concept for agent-based project scheduling is implemented in Discrete-Event-Simulation (DES).

2 Framework for Multi-Agent Negotations

As mentioned above the applied approach is based on prior works in Multi-Agent-Systems (MAS). The autonomous entities and its distributed intelligence split the complex problem into smaller tasks and enable a reliable, decentralized solving of the respective problem.

The introduced system however uses agent technology also for resources. Therefore individual resources can be allocated depending on specific criteria such as location, transport times or costs. A central blackboard facilitates all negotiations between processes and resources (see figure 1).

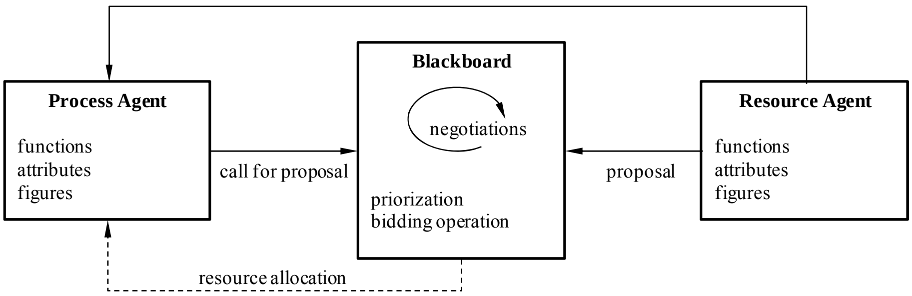


Figure 1. Multi-Agent system for project scheduling

Both processes and resources are implemented as autonomous agents. Hence resources are able to measure their qualification for the corresponding task by utility functions. Process agents request proposals on a central blackboard/marketplace, where all admitted agents may negotiate on resource allocation whilst considering existing constraints. Besides this further constraints appear in the construction environment, which can be implemented in the introduced MAS.

The objectives of each individual agent should lead to a general improvement of the system. Subject of optimization can both be minimal project duration as well as minimal resource deployment to complete all construction tasks in time. A schedule containing start and completion times of every activity and respective resource allocations is generated. Furthermore resource utilization is recorded.

2.1 Agents

Agents can be defined as self-contained program, which controls its own decisions, acts according to its perception of the environment and operates according to one or multiple objectives (Wooldridge and Jennings, 1995). Nwana specifies smart agents by the following three criteria, from which two must be satisfied.

- Autonomy – Agents operate without human guidance chasing specific objectives. An essential element is pro-activeness, i.e. the initial ability to operate itself rather than merely reacting.
- Cooperation – This characteristic is compulsory for MAS. Interaction between individual agents and possibly humans is necessary. For this purpose a communication language is set up.
- Learning – From reactions and/or interactions with their environment intelligent entities learn, so they can act ideally in similar situations (Nwana, 1996) (Ren and Anumba, 2004).

2.1.1 Process agent

Each activity is represented by a process agent holding all necessary scheduling information. Besides a clear identification key process agents have various attributes such as capital c_j, duration p_j and respective start and completion dates. Furthermore tables for the *set of predecessors* P_j and *successors* S_j as well as for *(non-)renewable resources* r_{jk} (h_{jk}) are provided. Control mechanisms for communication and interaction are combined in one central place, which forwards information to the respective targets. An agent processes these information, manipulates its state and operates accordingly. Table 1 shows the different states of an agent and its corresponding operations.

Table 1. Different states of process agents.

state	description	operations
blocked	preconditions not met	wait for predecessors
admitted	preconditions met for negotiations	registration at blackboard and call for proposals
accepted	resources allocated	wait for resources & predecessors' completion
active	resources allocated and working	finish activity in duration p_j, update successors
complete	activity finished	deallocation of resources

Since process agents interact amongst each other, these states can be varied according to the states of predecessors and the negotiation for resources. For the latter a new class of resource agents were modeled as follows.

2.1.2 Resource agent

As mentioned above in traditional approaches resources were not implemented as autonomous agents but rather as non intelligent entities, which were not part of the allocation process. This approach however considers resources as intelligent entities, which participate in the decision-making. Resource agents have specific functions and variables. Usually there are two types of resources:

- renewable – a set of resources available for every period of planning, such as machinery or workers
- non-renewable – a specific amount of units available for the entire planning horizon, e.g. material

Each entity is an instance of either renewable or non-renewable resource agents. Hence they differ in attributes and activities performable. Resource agents change between the three states *free*, *active* and *reserved*, which are self explaining.

Process agents place a call for proposals on the blackboard requesting *renewable resources* r_{jk} and *non-renewable resources* h_{jk} to complete their activities. Permitted resource agents on the contrary offer their labor depending on the specific qualification for the quoted activity. Thus the proposal can be dynamically generated from the current state of the specific resource, its characteristics and the influences of the environment. Agents consider their situation for negotiation and renew their proposals accordingly. Therefore utility functions were designed and implemented. Once allocated resources stay with the corresponding process agent until completion of the activity. Resource agents not allocated are placed in a central resource pool, from where new proposals may be placed on the blackboard. Overall workload for individual resources as well as for groups of identical characteristics can be determined and evaluated over time.

Both process and resource agents provide a central point of communication that routes all information between agents, blackboard and environment.

2.2 Communication protocol

Communication procedures are vital for the stability of the introduced MAS. Since communication effort in agent-systems is exceptionally high, all interactions require specific and robust protocols implemented in all agents as well as in the blackboard. A central point in every object merges all information and forwards them to the corresponding functions. The latter process these information and start respective actions, such as changing the state of the agent.

Process agents interact directly with their predecessors and successors. Further procedures for the communication to the blackboard are implemented, which place calls for proposals resulting in assignment of the requested resources. Resource agents also process all information in one particular protocol and act respectively. As soon as resources are deallocated (state *free*) by the process agent, they ask for further activities on the blackboard. New proposals are created accordingly.

2.3 Blackboard/marketplace

The blackboard or marketplace is a centralized trade forum for all agent negotiations. Admitted process agents call for proposals and resource agents offer their labor as introduced. To gain access to the blackboard agents have to meet specific requirements.

A particular challenge in traditional approaches is that resources are allocated immediately after the completion of prior activities. As shown in figure 2 potentially more critical processes have to wait for necessary resources. For this reason the possibility of reserving resources was designed.

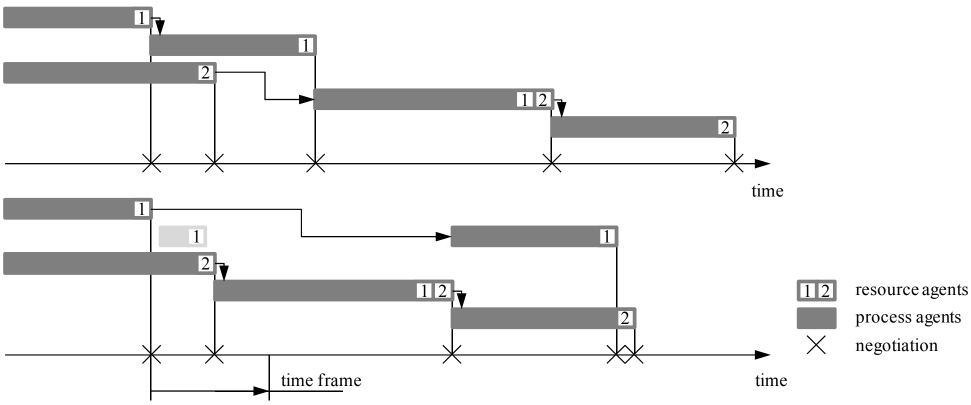


Figure 2. Schedule with (top) and without the possibility to reserve resources

To realize these reservations a time frame was designed that allows processes to enter negotiations with predecessing activities not completely finished. Processes within this time frame change their state to *admitted* as soon as all predecessors reach this time. Likewise the respective resource agents of predecessing activities are admitted to the negotiations. A timestamp confirms the future time of completion, where all succeeding processes are ready to begin and resources are deallocated.

The negotiations as such follow a defined pattern, which is based on the calls for proposals as well as the corresponding dynamic offers of resource agents. First step is a priorization of all existing calls. Agents with the largest *capital* c_j (see section 2.4) may review available offers at first. Provided that all required resources are available and predecessing activities are completed, the process can start its activities. If predecessing activities are not completed and/or resources are not deallocated yet – but will so within the time frame – resources will be reserved for the specific process. In case any of these requirements is not met, the process can not start yet. This operation is performed iteratively for all processes according to the sequence determined by the *capital* c_j.

Furthermore a second bidding procedure examines, whether resources allocated and reserved for a future process can be used for another process, which is completed before the appointed time of the former. Hence short-time activities with lower *capital* c_j can possibly be completed beforehand (see light grey process in figure 2).

2.4 Currency for negotiations

The *capital* c_j of individual process agents can be calculated based on various algorithms. Those have big influence on the quality of the computed results, since the capital decides on the priorization during bidding procedures. Due to space restrictions only one approach should be further evaluated.

The calculation of the capital is based on the longest path following, which stands for the accumulated maximum duration of all successors. For this purpose a recursive function was implemented, which determines the longest path through the network by self-referencing. Result is the capital for every process agent, *activity 0* having the largest capital. Consequently processes close to the critical path tend to have larger capital, so that this bottleneck is primarily served.

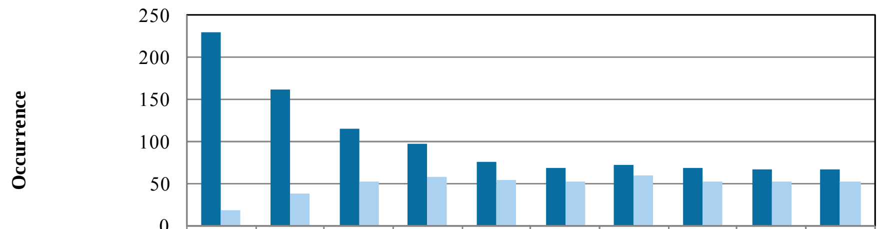
The resource agent calculates its proposals based on its parameters from utility functions. Since the individualization of agents does not influence the overall optimum, only workload and actual state are considered in this paper. Processes favor those resources with the highest qualification for the respective activity.

3 Results and Validation

The concept for agent based project scheduling is implemented in Discrete-Event-Simulation (DES). A central MAS-framework for the allocation of resources was designed, where all operations in terms of communication and interaction perform. Autonomous agents were implemented as object-oriented class; one each for processes and resources. From these any number of user-defined instances can be generated and specific parameters assigned. Furthermore a blackboard was created, which realizes the procedures introduced above. Additionally an observer was integrated that monitors resource agents and records their utilization over time.

For the evaluation of the developed MAS a specific interface was created and several examples from Operations Research were imported. 480 standardized instances were each tested for schedules containing n = 30, 60, 90 und 120 activities, that have been generated by the project generator ProGen (Kolisch and Sprecher, 1996). Thus the quality of the agent-based system is verified in relation to the optimal solution. For further validation the existing concept of constraint-based methods using Monte-Carlo-Simulation was implemented in the agent environment. Hence computing time and results of both concepts can be compared.

In figure 3 the occurrence of the best agent-based simulation results from the 480 examples with 120 activities is plotted over the applied time frame from zero to nine units. Different time frames often return the same (best) result. Yet most top results are created with a time frame of zero units, which is also used for the constraint-based approach. The occurrence of top results is decreasing with longer time frames. The optimal solutions however are less likely to be found with short time frames. Hence the concept of resource reservations with time frames is a crucial parameter to find good or optimal solutions for various scheduling problems.

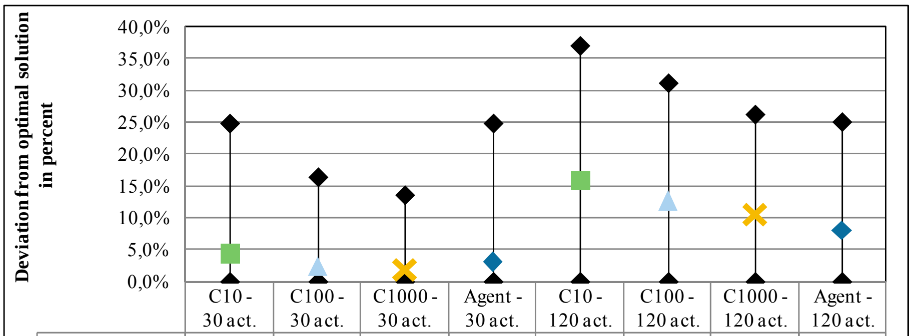


	0	1	2	3	4	5	6	7	8	9
best agent-based	229	161	115	98	75	69	72	69	66	67
optimal solution	19	38	53	58	54	53	59	53	52	53

time frame from zero to nine time units

Figure 3. Scheduling results of the agent-based simulation plotted over applied time frames

Figure 4 shows the results of 480 scheduling problems with 30 and 120 activities each in relation to the optimal solution for each use case. Constraint-based results improve with every additional exponentiation of simulation runs. For 1000 runs (C1000) with random resource allocation and 30 activities the best result is on average 1.7% higher than the optimal solution. The agent-based approach performs slightly worse for this use cases (3.0%). For both scheduling approaches the mean deviation increases with scenarios of 120 activities. Yet the agent-based simulation provides better solutions than the constraint-based approach.

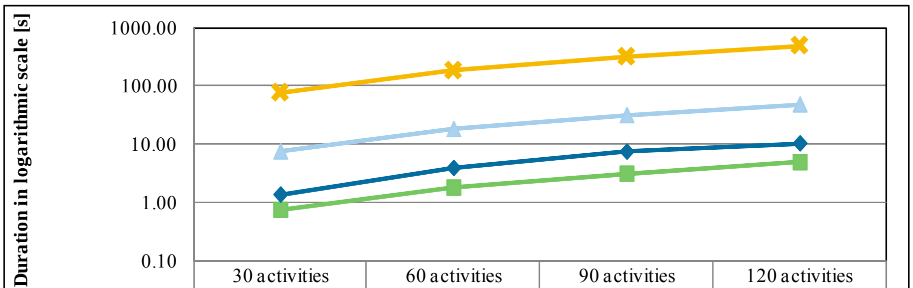


	C10 - 30 act.	C100 - 30 act.	C1000 - 30 act.	Agent - 30 act.	C10 - 120 act.	C100 - 120 act.	C1000 - 120 act.	Agent - 120 act.
Biggest deviation	25,0%	16,5%	13,6%	25,0%	37,0%	31,3%	26,3%	25,2%
Lowest deviation	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
Mean deviation	4,3%	2,4%	1,7%	3,0%	15,9%	12,6%	10,5%	8,0%

Figure 4. Scheduling quality of agent-based and constraint-based simulation

The quality of the constraint-based approach could be increased by further increasing the number of simulation runs. However common schedules in construction projects contain much more than 120 activities causing high computational effort for the constraint-based approach.

For reasons of comparison the agent-based as well as the constraint-based approach are implemented in the introduced simulation framework. Figure 5 shows the average time consumption for both concepts. The agent-based scheduling approach including ten experiments (time frame zero to nine) requires about double time compared to constraint-based scheduling. This effect can be explained with the more sophisticated resource allocation. The time for the constraint-based scheduling increases linearly with the number of runs. For one thousand runs the time required is 500 times higher than in agent-based scheduling – delivering worse results (figure 4). Additionally the duration of runs is increasing with the number of activities. Hence for big schedules the constraint-based approach requires long experimentation times in order to provide a moderate schedule quality.



	30 activities	60 activities	90 activities	120 activities
C10	0.75	1.80	3.11	4.90
C100	7.51	18.17	31.49	48.28
C1000	76.17	181.73	314.94	489.15
Agent-based	1.36	3.90	7.42	10.33

Figure 5. Mean computing time of different project sizes

4 Conclusions

The approach introduced in this paper improves the scheduling of construction projects by ensuring an intelligent allocation of resources based on Multi-Agent-Systems. Therefore both processes and resources were modeled as autonomous agents following individual objectives. Additionally a concept for reserving resources for more critical processes was realized by a time frame mentioned above. Subject of optimization can both be minimal project duration as well as minimal resource deployment to complete all construction tasks in time.

The validation proved the concept as very reliable creating high quality solutions for the RCPSP considering constraints specifically adjusted to construction projects. The computing time for the agent-based solution as well as the quality of results is remarkably well in relation to Monte-Carlo-Simulations. The number of runs can even be reduced by choosing less variations for the time frame. Further research addresses this challenge. Moreover different algorithms for the calculation of agent capital will be evaluated for construction scheduling and the utility functions of resource agents will be extended. A further step will be a case study using real project data.

Acknowledgements

The research presented in this paper was carried out within the project FAUST, which is funded by the Bavarian Research Foundation (Bayerische Forschungstiftung).

References

BEIßERT, U., KÖNIG, M. and BARGSTÄDT, H.-J., 2008. Simulation von Ausführungsstrategien mit Hilfe von Soft Constraints und lokaler Suche. *In: Rabe, M. (Ed): Advances in Simulation for production and logistics applications.* Stuttgart: Fraunhofer IRB Verlag, 141–150.

BRUCKER, P., DREXL, A., MÖHRING, R., NEUMANN, K. and PESCH, E., 1999. Resource-constrained project scheduling: Notation, classification, models, and methods. *European Journal of Operational Research*, 112, 3-41.

CHRISTODOULOU, S., 2005. Ant Colony Optimization in Construction Scheduling. *In: Soibelman, L.; Pena-Mora, F. (Ed): Computing in civil engineering.* Reston, Virginia.

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.

DEMEULEMEESTER, E.L. and HERROELEN, W.S., 2002. *Project scheduling: a research handbook.* International Series in Operations Research & Management Science, 49.

KOLISCH, R. and SPRECHER, A., 1996. PSPLIB – A project scheduling library. *In: European Journal of Operational Research*, 205--216.

JENNINGS, N.R. and WOOLDRIDGE, M., 1995. Applying agent technology. *In: Applied Artifcial Intelligence*, 9 ,357-369.

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

HEILMANN, R. and SCHWINDT, C., 1997. Lower bounds for RCPSP/max, *Report WIOR-511*, Universität Karlsruhe.

KNOTTS, G., DROR, M. and HARTMAN, B. C., 2000. Agent-based project scheduling. *In: IIE Transactions*, 32, 387–401.

KÖNIG, M., BEIßERT, U., STEINHAUER, D. and BARGSTÄDT, H.-J., 2007. Constraint-Based Simulation of Outfitting Processes in Shipbuilding and Civil Engineering. *In: 6th EUROSIM Congress on Modeling and Simulation.* Ljubljana.

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, Texas.*

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*, 17-6, 670–681.

MAROTO, C. and TORMOS, P., 1994. Project Management: an Evaluation of Software Quality. *In: International Transactions in Operational Research*, 209-221.

NWANA, H. S., 1996. Software agents: an overview. *In: The Knowledge Engineering Review*,11-3, 205–244.

REN, Z. and ANUMBA, C.J., 2004. Multi-agent systems in construction-state of the art and prospects. *In: Automation in Construction*, 13-3, 421–434.

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*, 39-12, 1023–1028.

TOKLU, Y.C., 2002. Application of genetic algorithms to construction scheduling with or without resource constraints. *In: Canadian Journal of Civil Engineering*. 29-3, 421–429.

WOOLDRIDGE, M. and JENNINGS, N.R., 1995. Intelligent agents: theory and practice. *In: The Knowledge Engineering Review*, 10-2, 115–152.

ZHANG, H., TAM, C.M. and LI, H., 2006. Multimode Project Scheduling Based on Particle Swarm Optimization. *In: Computer-Aided Civil and Infrastructure Engineering*, 21-2, 93-103.