18th International Conference on the Application of Computer
Science and Mathematics in Architecture and Civil Engineering
K. Gürlebeck and C. Könke (eds.)
Weimar, Germany, 07–09 July 2009

SIMULATION-BASED OPTIMIZATION OF CONSTRUCTION SCHEDULES BY USING PARETO SIMULATED ANNEALING

Matthias Hamm*, Ulrike Beißert, Markus König

**Bauhaus-University Weimar*
Marienstrasse 7, 99423 Weimar, Germany
E-mail: matthias.hamm@uni-weimar.de

Keywords: Construction Scheduling, Constraint Satisfaction, Multiobjective Optimization, Simulated Annealing, Pareto-Optimal.

Abstract. *Within the scheduling of construction projects, different, partly conflicting objectives have to be considered. The specification of an efficient construction schedule is a challenging task, which leads to a NP-hard multi-criteria optimization problem. In the past decades, so-called metaheuristics have been developed for scheduling problems to find near-optimal solutions in reasonable time. This paper presents a Simulated Annealing concept to determine near-optimal construction schedules. Simulated Annealing is a well-known metaheuristic optimization approach for solving complex combinatorial problems. To enable dealing with several optimization objectives the Pareto optimization concept is applied. Thus, the optimization result is a set of Pareto-optimal schedules, which can be analyzed for selecting exactly one practicable and reasonable schedule. A flexible constraint-based simulation approach is used to generate possible neighboring solutions very quickly during the optimization process. The essential aspects of the developed Pareto Simulated Annealing concept are presented in detail.*

1 INTRODUCTION

Construction projects consist of a multitude of construction tasks, which have to be scheduled efficiently in terms of different, partly conflicting project objectives such as time, cost, and quality. Therefore, several complex execution restrictions such as technological dependencies, safety aspects, and material availability have to be considered. This leads to a NP-hard multi-criteria optimization problem. Therefore, the analytical calculation of an optimal schedule with exact mathematical methods is computationally impractical.

In the past decades, so-called metaheuristics have been developed to find good and near-optimal solutions to scheduling problems in reasonable process time. Simulated Annealing is a well-known metaheuristic optimization approach for solving complex combinatorial problems such as scheduling construction projects. The concept of Simulated Annealing is inspired by the physical annealing process in metallurgy [1, 2]. It is a random search method that avoids getting trapped in local optima by accepting deteriorated neighboring solutions with a certain probability. Thereby, the probability of accepting a deteriorated solution decreases relative to the current temperature. In order to deal with several optimization objectives Czyzak and Jaszkiewicz [3] developed a modified Simulated Annealing algorithm based on the Pareto front. The Pareto Simulated Annealing (PSA) algorithm considers the complete Pareto front during the annealing process. Thus, the optimization result is a set of Pareto-optimal solutions based on the specified objectives. Afterwards, the Pareto-optimal solutions, in our case the Pareto-optimal schedules, can be analyzed to select exactly one practicable and reasonable schedule for the construction project. Of course, this selection depends on the individual preferences of the decision-makers with regard to the examined objectives.

Within this paper, the Pareto Simulated Annealing Metaheuristic is applied and adapted to determine near-optimal construction schedules. In the course of the optimization process a constraint-based simulation approach is used to generate possible solutions very quickly [4]. The simulation approach is based on the well-known constraint satisfaction concept, which is a powerful paradigm for modeling complex combinatorial problems [5]. The application of Pareto Simulated Annealing for construction scheduling requires different problem-related specifications. In this paper different Pareto Simulated Annealing aspects are presented in detail. Important aspects are the generation of neighboring schedules and the rules of acceptance of those schedules. Other aspects that will be presented in detail are temperature-based ones, the start temperature, decreasing rates, and determination criteria.

Primarily, the applied constraint-based simulation approach is presented to generate valid construction schedules regarding different requirements and restrictions. The specification of efficient schedules leads to a multi-objective optimization problem. Thus, in Section 3the evaluation of solutions with more than two objectives is discussed and the Pareto approach is highlighted. Nowadays, the metaheuristic Simulated Annealing can be applied to find near-optimal solutions for NP-hard scheduling problems. General aspects of the Simulated Annealing approach are described in Section 4. Construction scheduling is a NP-hard multi-objective scheduling problem. An adopted Pareto Simulated Annealing approach is presented in detail to solve construction scheduling problems using constraint-based simulation.

2 CONSTRAINT-BASED SIMULATION CONCEPT

Within the SIMoFIT (Simulation of Outfitting Processes in Shipbuilding and Civil Engineering) joint venture, a constraint-based simulation approach has been developed to improve execution planning [4]. Construction scheduling problems can be described by Constraint Satisfaction, which is a powerful paradigm for modeling complex combinatorial problems [5]. Classical constraint satisfaction problems are defined by sets of variables, domains, and constraints [6]. Accordingly, modeling the construction scheduling problems as constraint satisfaction problems, the construction tasks, materials, work force, equipment, and construction site layout are represented by variables. Different scheduling constraints can be specified between these variables. Typical stringent constraints of construction processes are technological dependencies between execution activities, certain equipment and manpower requirements, availability of materials, and safety aspects, such as specific working areas or maximum time allowances [4].

The solutions to constraint satisfaction problems are valid execution orders for the construction tasks, where all associated constraints are fulfilled. Usually, an analytical solution to complex constraint satisfaction problems is extremely time-consuming. However, simulation can be used to generate a possible solution, i.e., a valid execution order with fulfilled constraints, very quickly. Thus, the constraint satisfaction approach was integrated into a discrete event simulation application.

The simulation concept enables the generation of different events during the discrete simulation by the procedures *Starting Tasks* and *Stopping Tasks*. A new event occurs each time a working task is finished, as well as each time a new task is ready to be executed, i.e., each time the order to start a new task is delivered. However, a task can only be executed if all its associated constraints are fulfilled. In Figure 1 the procedure of starting tasks is depicted. If a new event occurs, all not-started tasks are checked on fulfillment of their associated stringent constraints. This leads to a set of next executable tasks. In the next step, one of these executable tasks is selected for starting. The currently started task is consecutively numbered and stored in an ordered task execution list (TEL). Consequently, a starting index is determined for each task. The tasks' presupposed objects like material, resources, or employees are locked during its execution and cannot be used by other tasks. This procedure is repeated until no more tasks can be started at that time. If the remaining time of a construction task has expired, the task is marked as finished. Its presupposed objects are unlocked and can be used by other construction tasks.

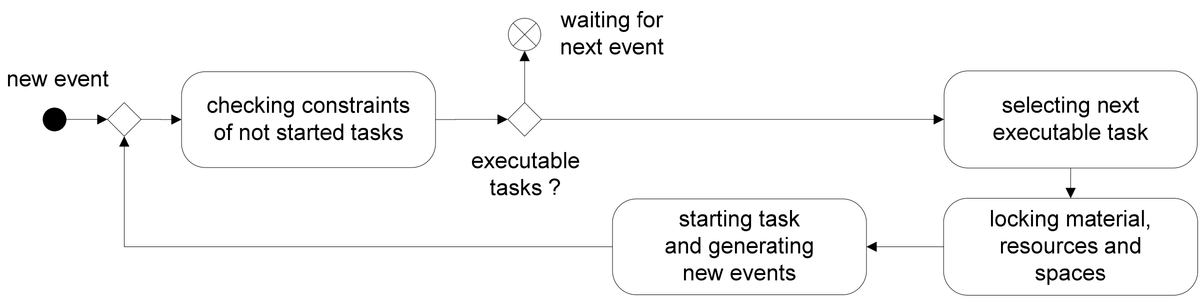

Figure 1: UML-diagram of starting tasks

The starting and stopping routines are carried out sequentially until all construction tasks are finished. All events, i.e., starting and finishing tasks and locking and unlocking resources, are recorded. Thus, one simulation run calculates one execution schedule with the respectively required material flow as well as the requirement for employees and equipment where all stringent construction constraints are fulfilled. The determined execution schedule is stored in the above-mentioned list, where the number of every task is stored in chronological order according to their starting time. The constraint-based approach guarantees a high flexibility of modeling construction processes. If additions or new prerequisites occur, the model can easily be adapted by adding or removing certain constraints.

3 MULTI-OBJECTIVE OPTIMIZATION

When modeling real-world problems, such as construction projects, it is often not possible to reduce the optimization problem to one single objective function. Often several, partly contrary objectives have to be considered. Hence, several objective functions have to be optimized simultaneously. This leads to so-called multi-objective optimization problems. Today, different methods exist to model multi-objectives for optimization. For example, it is possible to use a weighted aggregation function and treat the problem like a single-objective optimization problem [7]. Alternatively, the most important objective can be optimized while all other objectives only fulfill certain constraints [7]. Both of these methods, as well as other scalar methods like Goal Programming, have in common that they determine one optimal solution in an a-priori way, i.e., an optimal solution will be determined through the consideration of pre-defined input parameters. However, there are often situations where it is more favorable for the decision maker to choose a solution in hindsight, i.e., in an a posteriori way. First, a given number of good solutions are generated. Following this, one good solution is selected as the optimal solution based on subjective preferences. The a-posteriori Pareto approach is used in this paper [8].

During Pareto-Optimization good solutions are stored throughout the optimization process. To decide whether an identified solution will be saved or not, the concept of Pareto domination is applied. A solution is dominated by another solution if the solution does not outperform other solutions on all objectives and performs significantly worse for at least one objective. Thus, if the number of considered objective functions f is M, a solution A dominates a solution B, if

$$\begin{aligned} & f_i(A) \leq f_i(B) \forall i \in \{1,\dots,M\} \text{ and} \\ & \exists j \in \{1,\dots,M\} : f_j(A) < f_j(B). \end{aligned} \tag{1}$$

The domination relation is not a total order. Two solutions are mutually non-dominating if neither solution dominates the other. A solution that is non-dominated by any other solution is Pareto-optimal. All Pareto-optimal solutions that are detected during the optimization process specify a so-called Pareto-Front. Generally, all non-dominated solutions of the Pareto-Front are evaluated afterwards by decision makers.

In Figure 2 the Pareto-Front of a certain amount of solutions for two objective functions is depicted. Additionally, some dominated solutions are shown that are not included in the Pareto-Front. If a new solution is calculated and is not dominated by any other determined solution, the Pareto-Front must be updated. The new solution is added to the Pareto-Front and all prior non-dominated solutions have to be checked again. If a Pareto-Front solution is now dominated by the new solution, those solutions are removed from the Pareto-Front [8, 9].

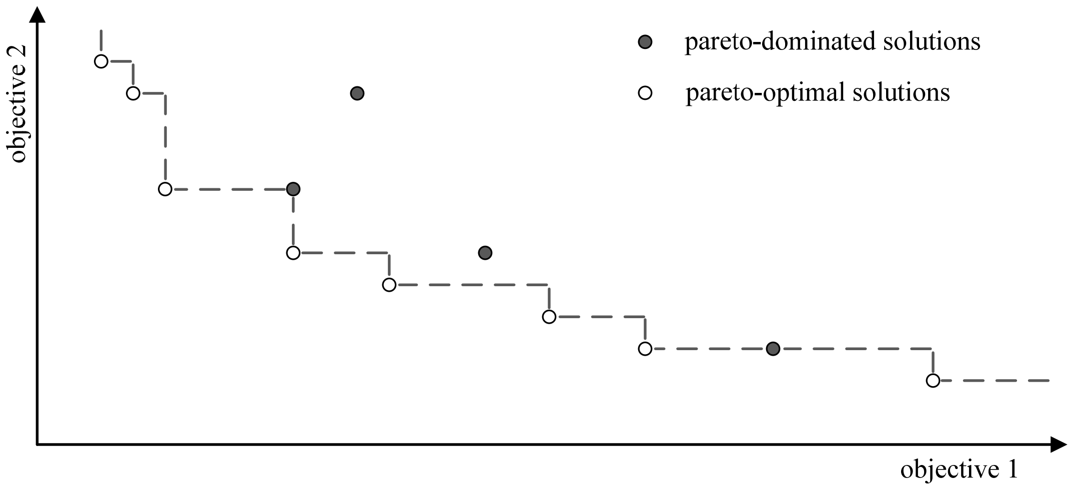

Figure 2: Pareto-Front of solutions based on two objective functions

4 SIMULATED ANNEALING

Simulated Annealing is a well-known local optimization approach for solving complex combinatorial problems. The general goal of local optimization methods is to find good solutions in an adequate amount of time. The concept of Simulated Annealing is inspired by the physical annealing process in metallurgy [1, 2]. In this context, annealing is known as the heating and controlled cooling of metal to bring the material structure from an arbitrary initial state to a state with the minimum possible energy. During heating, the metal atoms become unstuck from their current position and arrange themselves randomly. The slow cooling phase allows the atoms to find highly structured configurations with lower internal energy than in the initial configuration [10, 11, 12].

If this physical process is considered as an analogy for general optimization, the solutions of an optimization problem represent the possible configurations of the atoms. The objective value of a solution, the so-called cost factor, is equivalent to the internal energy state. Starting with a high temperature and a randomly selected initial solution, the Simulated Annealing heuristic calculates a new solution within a certain neighborhood of the current solution. If a new solution has a better cost factor than the current solution it will be always accepted as the new solution. If a new solution does not perform better than the current one, the acceptance of new solutions is based on a probability that depends on the difference between the corresponding costs and on the current temperature. Consequently, a high temperature allows the acceptance of a new solution, which causes higher costs. The probability of accepting higher costs decreases within the optimization process. Once accepted, the new solution is the new starting point for the next optimization step. In order to use the Simulated Annealing heuristic an appropriate neighborhood, a reasonable start-temperature, a good temperature-based probability, and an effective decreasing rate for the temperature have to be specified. The general Simulated Annealing procedure is depicted in Figure 3 accordingly to [10].

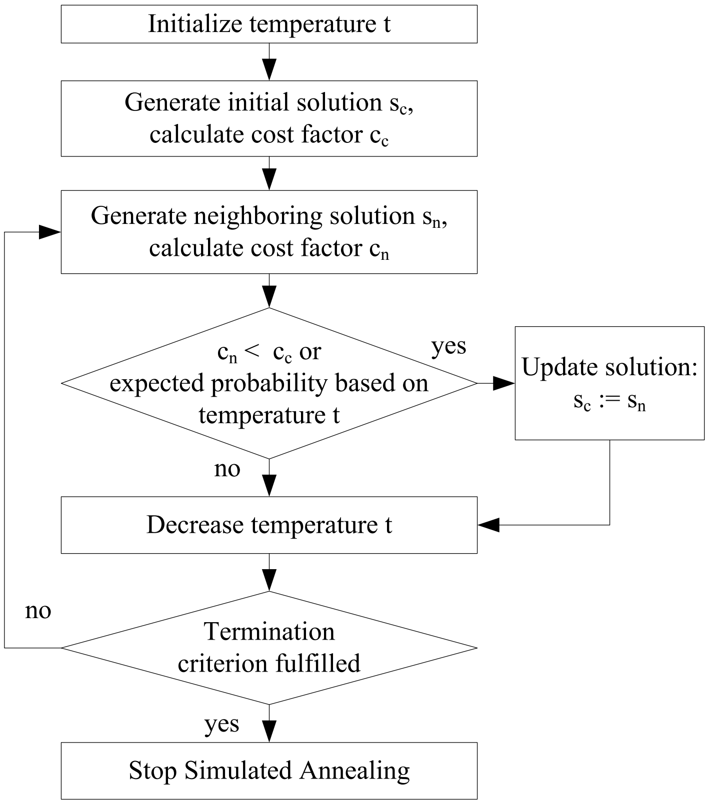

Figure 3: Simulated Annealing algorithm accordingly to [10]

5 CONSTRAINT-BASED PARETO SIMULATED ANNEALING

Specifying efficient construction schedules is a challenging task. Many different and complex objectives have to be taken into account. In this paper, Simulated Annealing in combination with Pareto-Optimization is used to generate efficient or near-optimal construction task sequences by applying the aforementioned constraint-based simulation concept. Thereby, the non-dominated solutions of the Pareto-Front, in this case the construction schedules, are stored in a so-called Pareto-Archive. In other words, the Pareto-Archive contains the "best" valid solutions at a particular point in time. Decision-makers can choose one of these solutions in an a-posteriori decision-making process based on their preferences. The Pareto-Archive is also used to compare a current solution with other solutions, which are stored in the archive during the Simulated Annealing optimization process. Related approaches have been presented by Smith et al. [9], Bandyopadhay et al. [13], and Suppapitnarm et al. [14]. Contrary to these concepts, an adapted Pareto Simulated Annealing procedure has been developed for use within the constraint-based simulation. In the following paragraphs, the adapted Pareto Simulated Annealing approach is presented in detail.

The generation of a neighboring solution is one of the main differences between this paper and the previously published and above-mentioned scholarly work on this topic. In contrast to other approaches, neighboring solutions are determined by using constraint-based simulation. This means that during the simulation several modification steps are performed reading the defined restrictions. The decision of whether a neighboring solution is chosen as the new current solution will be made in accordance with the standard rules of acceptance of the

Simulated Annealing algorithm. Thus, even deteriorated solutions may be accepted with a certain probability. This probability depends on the current temperature and the domination status of the current neighboring solution.

In the following paragraphs, a more precise description of different specifications that were implemented in the algorithm is presented. Starting with the generation of neighboring solutions, which is based on the implemented constraint-based simulation concept, the acceptation criteria are described in detail. The acceptance criteria are based on the so-called dominance measures or amount of domination. Moreover, specific requirements concerning the temperature stipulations of the Simulated Annealing Heuristic are presented.

5.1 Neighborhood

The definition of an appropriate neighboring solution is very important. A neighboring solution is determined by using the constraint-based simulation approach. The generation of the neighboring solution is based on the current solution, which means that the neighboring solution differs only in a few aspects from the current solution. Each time an event occurs, for example when a certain construction task is finished, the set of next executable tasks is determined (cf. Figure 1). If the set of next executable tasks contains more than one task, execution order is analyzed according to the current task execution list TEL (cf. Chapter 2). The current task order is modified by swapping two tasks randomly. All other tasks are selected for execution in the same order as they are stored in the current list TEL.

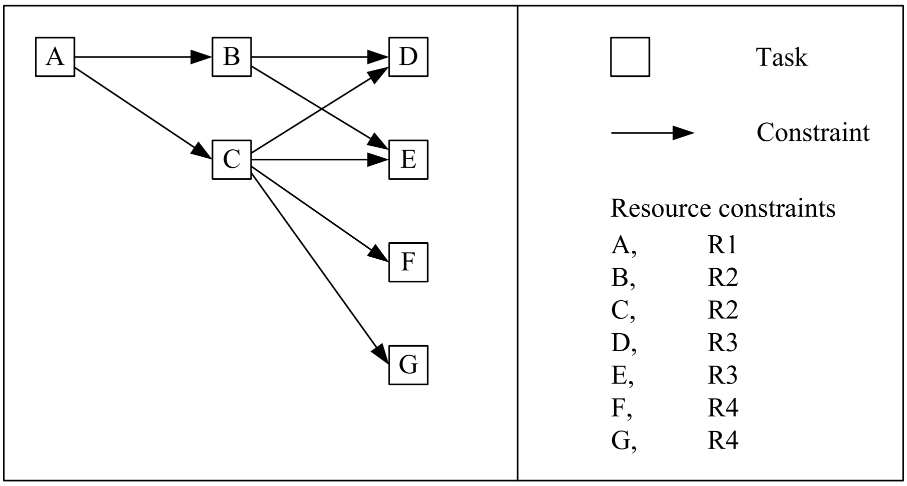

Figure 4: Topological ordering of construction tasks

Figure 4 shows the topological ordering of some construction tasks considering the depicted Hard Constraints for a simple scheduling problem. The problem consists of seven tasks {*A, B, C, D, E, F, G*} that have to be executed by four different resources {R1, R2, R3, R4}. Resource constraints are not considered when determining the topological ordering. Nonetheless, these constraints have a deep impact on the resulting schedules and therefore on the resulting objective function values. An initial solution can be generated based on the specified resource requirements (cf. Figure 5). In the presented case the initial task execution list is <*A, B, C, D, F, E, G*>.

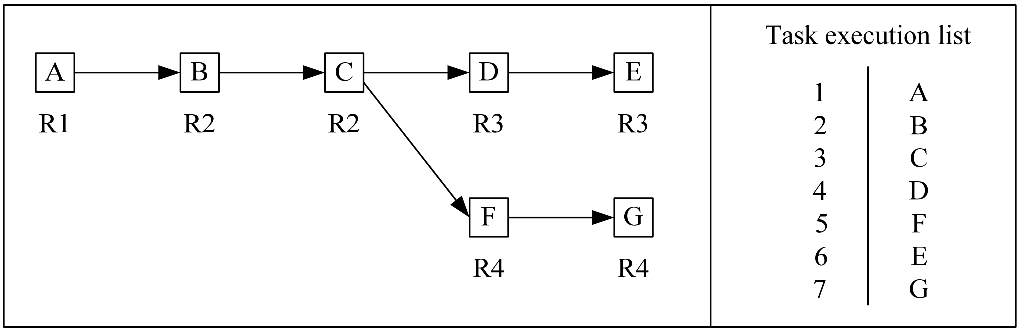

Figure 5: Initial execution sequence

Considering this current solution, a neighboring solution can be generated in the following manner. First, task A has to be executed. After task A is finished, an event occurs and all next-executable tasks are determined. In this case, these are tasks *B* and *C*. The positions of these tasks in the current execution list (2, 3) will be swapped. Thus, the partial order of these two tasks in the neighboring solution is $\{C, B\}$. All substitutions are stored in a Tabu list and cannot be used within further optimization steps. However, the size of the Tabu list is restricted. Registered task substitutions are removed following the FIFO method, i.e., if a certain number of task pairs is registered, the oldest entries are removed. Hence, these substitutions become possible again.

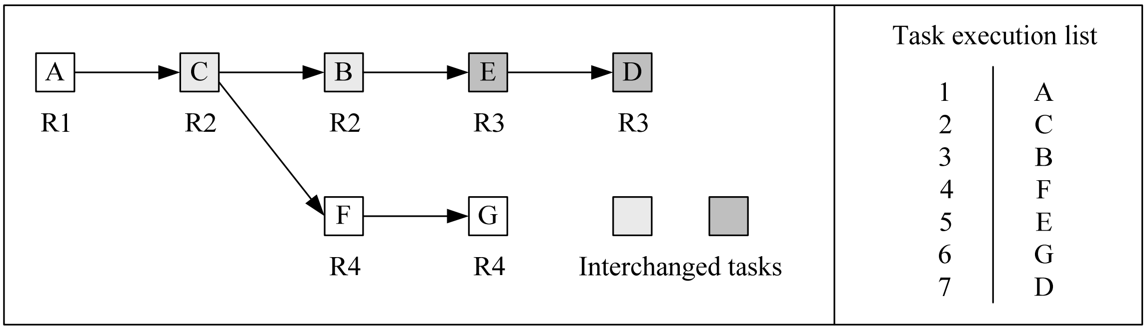

Figure 6: New neighboring execution sequence

Continuing in this way a potential neighboring solution is $\{A, C, B, F, E, G, D\}$. This new solution also fulfills the defined constraints and leads to another correct resource allocation (cf. Figure 6). The decision of whether the determined neighboring solution will be accepted as new current solution will be justified in the next paragraph.

5.2 Domination Status, Amount of Domination, and Acceptance Rules

As described above, the non-dominated solutions, which are determined during the simulation process, are stored in the Pareto-Archive. However, during the Simulated Annealing process a differentiation will be made between the non-dominated solutions and the current solution, which is a valid execution order for the examined construction tasks, where all associated constraints are fulfilled. The current solution is the respective origin for the generation of neighboring solutions. It may belong to the Pareto-Front, but doesn't necessarily have to. Based on this current solution, a neighboring solution is determined in the above-described manner. If a valid neighboring solution is defined, its acceptance within the archive is examined. A new solution is added to the archive if it is not dominated by any other solution in the archive. However, if the neighboring solution itself dominates solutions from the archive, the dominated solution will be removed while the new solution will be added to the archive.

The decision of whether the neighboring solution will be chosen as the new current solution is independent of its potential addition to the Pareto Archive. Three different cases are possible: the neighboring solution dominates the current solution, the neighboring and the current solution do not dominate each other, or the neighboring solution is dominated by the current one.

In the first case, when the neighboring solution dominates the current solution, the neighboring solution will be chosen as the new current solution. The Simulated Annealing iteration starts again based on the new current solution. In the second case, where the neighboring solution and the current solution do not dominate each other, the neighboring solution will also be accepted as the new current solution.

In the third case, when the neighboring solution is dominated by the current one, there is still a chance for the neighboring solution to be chosen as the new current solution. Based on the current temperature of the Simulated Annealing Algorithm, a probability is determined. If the probability is higher than a specified acceptance probability value the neighboring solution is chosen as new current solution. This probability also depends on the amount of domination between the neighboring solution and the current one. Given two solutions, the amount of domination is defined as

$$\mathrm{dom}_{A,N} = \prod_{i=1, f_i(A) \neq f_i(N)}^{M} \left(\left| \frac{f_j(A) - f_j(N)}{R_i} \right| \right) . \tag{2}$$

Where A is the current solution, M is the number of objectives and R_i is the range of the i-th objective. If the range is not known a-priori, R_i can be calculated by the difference of the greatest and the least observed values for the respective objective [13].

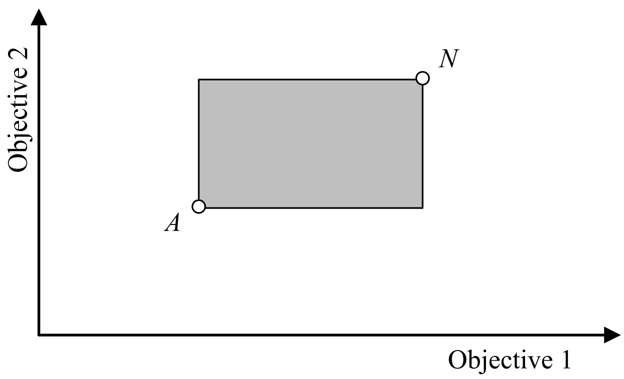

Figure 7: Amount of domination between the two solutions A and N

In figure 7, the amount of domination is illustrated for the case of two objectives. The probability of accepting the neighboring solution as new current solution is determined by

$$\text{prob} = 1 - \exp\left(-\text{dom}_{a,b} * \text{temp}\right). \tag{3}$$

The algorithm's positive probability of accepting a deteriorated solution avoids getting trapped in local optima. However, the temperature is decreasing during the simulation and thus the probability of accepting worse solutions decreases too. Obviously, the temperature specifications have a crucial influence on the performance and effectiveness of the algorithm. In the following paragraphs different temperature specification concepts are discussed in detail.

5.3 Temperature Specifications

Before starting a Simulated Annealing optimization the following temperature aspects have to be defined: the initial temperature, the decreasing rate, i.e., the rate at which the temperature decreases after a certain period, the number of iterations at one temperature level, and the termination criterion.

At the beginning, the Simulated Annealing algorithm accepts nearly every neighboring solution, i.e., the probability of accepting deteriorations should be near 1. Thus, a corresponding start temperature has to be defined. In this paper the start or initial temperature is specified according to Kirkpatrick [15]. Starting with a low temperature (i.e., 10) and a predefined acceptance probability (i.e., 0.8), a certain number of test runs are performed. If fewer than 90% of dominated solutions are accepted, the temperature is doubled. The test runs are repeated until the predefined rate is reached.

The number of iterations for the same temperature is increased over the course of the Simulated Annealing optimization. Starting with the iteration number $N_0 = 5$, the number is increased by a constant factor ρ each time the temperature is decreased:

$$N_{k+1} = \rho * N_k. \tag{4}$$

Within the constraint-based Pareto Simulated Annealing approach the increasing factor ρ is set to 1.15. After the described number of iterations at one temperature level, the current temperature will be decreased by

$$T_{k+1} = \alpha * T_k. \tag{5}$$

Where α is a constant cooling rate. The cooling rate is defined as less than and close to 1.0. Typically the cooling rate is specified as between 0.8 and 0.99 [16]. In the presented algorithm a cooling rate of α= 0.9 is used.

The termination criterion is defined in the following manner: the algorithm is performed until the temperature is less than 0.01. Furthermore, the Simulated Annealing optimization also stops if no neighboring solution is selected as the current solution during a certain amount of temperature reductions. The experience has shown that 10 temperature reductions without adaptation is a good termination criterion.

5.4 Constraint-based Pareto Simulated Annealing algorithm

In Figure 8, the complete constraint-based Pareto Simulated Annealing algorithm is depicted. It becomes obvious how the constraint-based simulation concept is implemented to generate neighboring solutions in the Pareto Simulated Annealing algorithm.

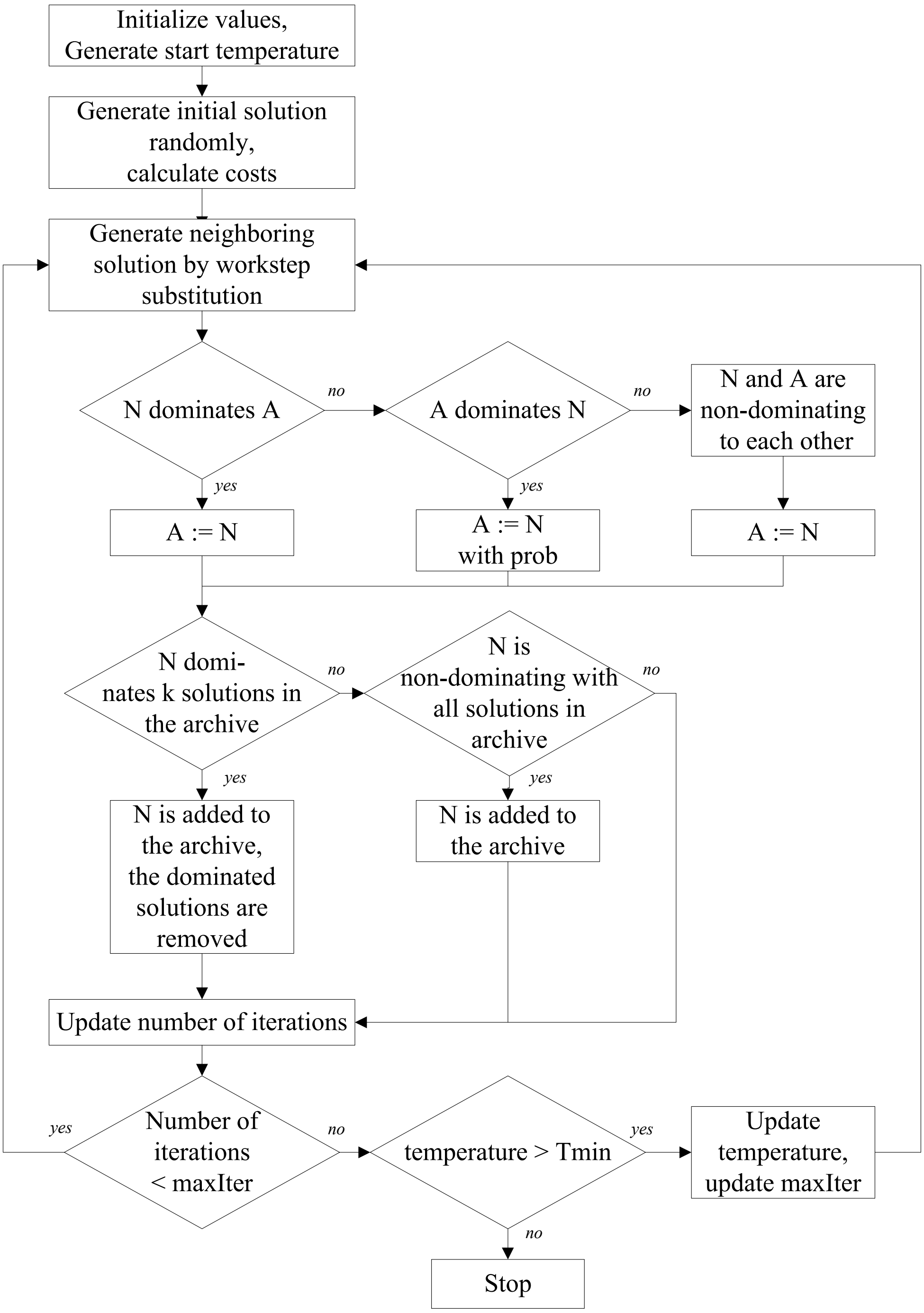

Figure 8: Constraint-based Pareto Simulated Annealing algorithm

The decision of whether a generated neighboring solution is accepted as the current solution, and thus as the basis for generating the next neighboring solution, is abstracted from the Simulated Annealing algorithm. Depending on the current temperature and the amount of domination, even worse solutions will be accepted with a certain probability. Within the Simulated Annealing algorithm, the concept of Pareto optimality is used to evaluate multi-objective solutions and store the not dominated ones in the Pareto Front. A certain number of iterations are run on each temperature level. Then the temperature is decreased and the number of iterations on the next temperature level is increased. The algorithm stops when a minimum temperature is reached or if there has been no improvement for a certain number of iterations.

6 CONCLUSION AND OUTLOOK

Within this paper a simulation-based optimization approach is presented to calculate near-optimal construction schedules regarding different, partly conflicting objectives. The approach is based on Simulated Annealing and Pareto Optimization in combination with constraint-based simulation. Thereby, constraint-based simulation is used to generate valid schedules and their neighboring solutions in reasonable time. Neighboring schedules are generated by swapping the execution orders of certain construction tasks. During the Simulated Annealing each determined neighboring solution is compared to the current solution using the presented Pareto domination concept. Based on the current temperature and the amount of domination a decision is made whether to accept a neighboring solution as the new current solution. The accepted solutions so-called Pareto-optimal solutions are stored in a Pareto archive. Consequently, the optimization result is a set of Pareto-optimal schedules, which can be analyzed to select exactly one practicable and reasonable schedule. Essential aspects for determining neighboring solutions, to calculate the domination status of solutions and the accepting rules of dominated solutions with a certain probability as well as temperature specifications are presented in detail.

The presented Pareto Simulated Annealing is still under development. Currently, generic software components for Pareto Simulated Annealing are implemented using a discrete event simulation framework [17]. First test cases in the area of construction finishing trades are set up to evaluate the presented optimization concept and its implementation. Furthermore, the integration of construction knowledge by using so-called soft constraints is projected [18].

REFERENCES

[1] S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi, Optimization by Simulated Annealing. *Science*, 220, 1983.

[2] V. Cerny, A thermo dynamical approach to the travelling salesman problem. *Journal of Optimization Theory and Applications*, Vol.15, 1985.

[3] P. Czyzak and A. Jaskiewicz, Pareto simulated annealing - a metaheuristic technique for multiple objective combinatorial optimization. *Journal of Multi-Criteria Decision Analysis*, Vol. 7, No. 1., pp. 34-47, 1998.

[4] U. Beißert, M. König and H.-J. Bargstädt, Constraint-based simulation of outfitting processes in building engineering. *24th W78 Conference*, 2007.

[5] J. Blazewicz, K. Ecker, E. Pesch, G. Schmidt and J. Weglarz, Handbook on scheduling: from theory to applications. *Springer*, 2007.

[6] F. Rossi, P. vanBeek and T. Walsh, Handbook of Constraint Programming. *Elsevier*, 2006.

[7] Y. Collette and P. Siarry, Multiobjective Optimization – Principles and Case Studies. *Springer*, Berlin, 2003.

[8] M. Ehrgott. Multicriteria optimization. *Springer*, Berlin, 2005.

[9] K. I. Smith, R. M. Everson and J. E. Fieldsend, Dominance Measures for Multi-Objective Simulated Annealing. *Proceedings of Congress on Evolutionary Computation*, CEC04, 23-30, 2004.

[10] E. Aarts, J. Korst and W. Michiels, Simulated Annealing in *Search Methodologies - Introductory Tutorials in Optimization and Decision Support Techniques* edited by E. K. Burk and G. Kandall, Springer, 2005.

[11] J. Dreo, P. Siarry, A. Petrowski and E. Taillard, Metaheuristics for Hard Optimization. *Springer*, Berlin, 2006.

[12] K. Y. Lee and M. A. El-Sharkawi, Modern Heuristic Optimization Techniques. *John Wiley & Sons, Inc.*, Hoboken, 2008.

[13] S. Bandyopadhyay, S. Saha, U. Maulik and K. Deb, A Simulated Annealing-Based Multiobjective Optimization Algorithm: AMOSA. *IEEE Transactions on Evolutionary Computation*, Vol.12, No.3, 269-283, 2008.

[14] A. Suppapitnarm, K.A. Seffen, G.T. Parks and P.J. Clarkson, A Simulated Annealing Algorithm for Multiobjective Optimization, *Eng. Opt. 33*, 59-85, 2005.

[15] S. Kirkpatrick, Optimization by simulated annealing: Quantitative studies. *Journal of Statistical Physics*, Vol. 34, Numbers 5-6, 975-986, 1984.

[16] E. Aarts, J. Korst and P. van Laarhoven, Simulated annealing, *in Local Search in Combinatorial Optimization*. E.H.L. Aarts and J.K. Lenstra (Eds.), John Wiley and Sons, 1997.

[17] M. König, U. Beißert, D. Steinhauer and H.-J. Bargstädt, Constraint-based simulation of outfitting processes in shipbuilding and civil engineering. *6thEurosim Congress in Modeling and Simulation*, 2007.

[18] U. Beißert, M. König and H.-J. Bargstädt, Execution Strategy Investigation Using Soft Constraint-Based Simulation; *IABSE ICT Conference 2008*, Helsinki, 2008.