Journal of Simulation (2010), 1–10 1747-7778/10

www.palgrave-journals.com/jos/

Soft Constraint-based simulation of execution strategies in building engineering

U Beißert[1], M König[2]* and H-J Bargstädt[1]

[1]Bauhaus-University Weimar, Weimar, Germany; and [2]Ruhr-University Bochum, Bochum, Germany

In building engineering different strategies can be used to schedule the execution processes of building projects. Currently, these strategies have not yet been sufficiently formalized, and considered for construction scheduling. This paper presents a concept for modelling and simulating execution strategies by using Soft Constraint representations. In particular, the GRASP heuristic is applied within a constraint-based simulation approach to improve construction schedules by observing well-established execution strategies. The benefits of this concept are shown for the strategies *Avoid Soiling* and *Human Strain Factor*.

Journal of Simulation advance online publication, 28 May 2010; doi:10.1057/jos.2010.8

Keywords: construction scheduling; constraint satisfaction; discrete-event simulation; knowledge representation; heuristic optimization

1. Introduction

During the scheduling of construction processes a multitude of different restrictions, such as technological dependencies between tasks and resource requirements, should be considered (Beißert *et al*, 2007), as should various execution strategies. In this context, strategies specify predefined successful global execution rules and process sequences. The application of such strategies guarantees an effective and robust process flow. However, their effectiveness strongly depends on local and project-specific requirements. Thus, the suitability of a particular strategy should be sufficiently investigated with respect to the aforementioned aspects before applying it in a scheduling process. However, in practice the specification of a schedule is mostly based on planners' subjective experiences, gained from formerly executed projects and transferred to upcoming projects. The strategies are neither sufficiently investigated nor statistically proven yet.

Within the cooperative agreement SIMoFIT (Simulation of Outfitting Processes in Shipbuilding and Civil Engineering) between Bauhaus-University Weimar, Ruhr-University Bochum and Flensburgers, a German shipyard, a constraint-based simulation concept was developed to improve the planning of construction processes, especially of finishing trades (Beißert *et al*, 2007; König *et al*, 2007). In today's construction practice execution strategies are manually applied in daily work planning in order to avoid additional work or disturbances. This paper highlights the formalization of the execution strategies *Avoid Soiling* and *Human Strain Factor* by using different kinds of Soft Constraint representations. Furthermore, the well-established Greedy Randomized Adaptive Search Procedure (GRASP) is integrated into the existing constraint-based simulation concept to find daily optimized work sequences with regard to a certain execution strategy. Finally, the presented concept is evaluated using a case study.

**Correspondence: M König, Gebaeude IA 6/153, Universitatsstrasse 150, Bochum 44780, Germany.*
E-mail: koenig@inf.bi.ruhr-uni-bochum.de

2. Related work

Commonly used scheduling instruments in the construction industry are mostly based on the Critical Path Method (CPM) (Sriprasert and Dawood, 2003b; Wang *et al*, 2004). This method provides a specification of execution times and sequencing of construction tasks, which can be defined manually as graph representations. Using these graphs execution dates, buffer times, and critical tasks can be calculated. The CPM is widely used in project management applications to manage employees and resources, to track delays, and to change orders (Sriprasert and Dawood, 2002). However, the inability to cope with non-precedence constraints, the difficulty of evaluating and communicating interdependencies, and the method's inadequacy for workface productions argue against the application of the method within complex planning tasks like construction scheduling (Pultar, 1990; McKinney and Fischer, 1998; Woodworth and Shanahan, 1998; Choo *et al*, 1999; Wang *et al*, 2004).

Other construction planning approaches include 4D visualization and animation concepts. Thereby 3D CAD components such as single building elements or whole element groups are linked with scheduled activities allowing the animation of the progression of construction in a 3D environment (Tulke and Hanff, 2007). Thus, 4D models enable the exploration and improvement of project execution strategies, facilitate improvements in constructability with corresponding gains in on-site productivity, and make possible the rapid identification as well as the resolution of time-space conflicts (Akinci *et al.*, 2002; Fischer and Kunz, 2004; Tulke and Hanff, 2007; Mallasi, 2004). However, 4D applications only allow time information from predefined schedules to be imported into static 3D models.

In manufacturing industries such as steel prefabrication and ship assembling, simulation models are used successfully to schedule and improve production processes (Steinhauer, 2006). The introduction of simulation models enables users to run experiments and what-if scenarios. In the construction industry, only a few research activities deal with construction-process simulation (Franz, 1989; Martinez, 1996; Halpin and Martinez, 1999; Akbas, 2004; AbouRizk *et al*, 2006; Wakefield and Sears, 1997). This research primarily focuses on fixed process sequences, but with varying resources and due dates. The construction processes or sub-processes are often modelled based on specific graphical simulation languages like Petri-Nets, CYCLONE, or STROBOSCOPE (Martinez, 1996; Halpin and Martinez, 1999; Wakefield and Sears, 1997). These approaches integrate common CAD tools to visualize and animate simulated execution solutions. Some of these simulation models are directly connected to CAD applications. In this manner dimensions and quantity of buildings can be used together (Akbas, 2004; Chahrour and Franz, 2004; AbouRizk *et al*, 2006).

Contemplated approaches are limited by either manually defined schedules or stringent process sequences that are animated or visualized. The modelling and analysis of various alternatives is very time consuming because a complete new schedule for the changed assumptions concerning construction methods has to be prepared. Thus, the variety of construction methods and processes is not considered adequately in the aforementioned approaches. Furthermore, Sriprasert highlights the importance of the consideration of precedence constraints like human knowledge. However, the introduced approaches lack this characteristic. Mikulakova *et al* (2008) employ case-based reasoning techniques to use scheduling experience from previous projects. Their execution sequence is presented to the project manager as a possible solution and can be adapted to new conditions. Sriprasert presents an evolutionary optimization system based on Microsoft Project to find an eligible solution for execution considering known constraints (Sriprasert and Dawood, 2003a).

Currently, the generation of consistent execution schedules through the use of constraint-based simulation models is not part of the mentioned research activities. Another advantageous planning aspect of using constraint-based planning approaches is the specification and consideration of existing know-how to define practicable and robust execution schedules. Defining these suitable schedules is currently an important topic within construction management.

3. Constraint-based simulation

In recent years simulation applications have been successively and intensively used in the manufacturing industry to optimize production processes. The modelling of construction sites is quite different from the modelling of production plants. The construction site layout, for instance, changes during the process, and thus transport paths and material flows have to be adapted. Owing to the fact that simulation applications in the manufacturing industry are limited to the support of static layouts, alternative simulation approaches have to be investigated to describe construction processes. Within the cooperation SIMoFIT a constraint-based simulation approach was developed to simulate construction execution processes. This approach guarantees a high degree of flexibility in modelling construction processes. Various requirements and restrictions can be described by different types of constraints. Furthermore, if additions or new prerequisites occur, the process can be easily adapted by defining or removing certain constraints.

3.1. Partial Constraint Satisfaction

The simulation approach introduced in this paper is based on the Partial Constraint Satisfaction principles (Freuder and Wallace, 1992). According to these principles, the simulation objects are specified by sets of variables and their associated domains. Relations between simulated objects are defined by a set of constraints. Contrary to classical Constraint Satisfaction approaches, the set of constraints is divided into Hard Constraints and Soft Constraints. Hard Constraints, on the one hand, specify stringent conditions of construction processes. Consequently, they have to be fulfilled completely before a construction task can be started. On the other hand, Soft Constraints specify functional conditions of the processes that can be violated within a limited range (Freuder and Wallace, 1992; Rossi *et al*, 2006). Solutions of Partial Constraint Satisfaction Problems are valid execution orders for all considered construction tasks, where all associated Hard Constraints are completely fulfilled, and the Soft Constraints are fulfilled to the greatest degree possible.

3.2. Construction task constraints

When a model for the simulation of construction execution processes is created, different objects and the relations between them have to be specified. Currently, construction tasks, resources like material and employees, and working areas are considered. The relations between these objects are described by the aforementioned Hard and Soft Constraints. In Table 1, an overview of appropriate and typical constraints for construction tasks is shown.

The consideration of defined strategies during scheduling enables the specification of certain preferences within the construction execution and therefore the evaluation of established process sequences or principles. To integrate strategies into the presented simulation concept in a standardized way, execution strategies are classified according to different execution aspects shown in Table 2. In this paper several well-known execution strategies are highlighted.

Simulation concept. The constraint-based approach is implemented within an event-discrete simulation concept to simulate single tasks. Therefore, each construction component is decomposed into its single process steps, which are referred to as *tasks*. Each task has a status of execution and is performed without interruption and changes to its associated employees, working space, and other resources. The simulation concept focuses on the fulfilment of constraints, which is checked whenever an event occurs. Typical events within an event-discrete simulation are, for example, a released resource or a completed material transport. During discrete event simulations different events are generated by the procedures *Starting Tasks* and *Stopping Tasks* (König *et al*, 2007). If a new event occurs, all not-yet-started tasks have to be checked in terms of the fulfilment of their Hard Constraints. Afterwards, all executable steps are checked and ordered by the degree to which they have fulfilled the Soft Constraints. The first of the listed steps will be started. Its required resources are locked during execution. All not-started tasks must be checked in the same way again on the fulfilment of their constraints, until no more tasks can be started at the current event (cf Figure 1).

Table 1 Typical construction task constraints

Hard constraints	*Soft constraints*
Technological dependencies *Constructional and formal aspects*	Strategies *Proven formal aspects*
Capacity *Resource boundaries*	
Availability *Material flow restriction*	
Safety *Protection criteria*	

The simulation time is continuously checked during the simulation run. Every started task exhibits a determined execution time. If the remaining time has expired, the task is marked as finished and its attached resources will be unlocked.

Both *Starting* and *Stopping Tasks* will be performed repeatedly until all tasks are finished. All events, such as the starting and finishing of tasks as well as locking and unlocking of resources, are recorded. Thus, one simulation run calculates one practical and valid execution schedule, with one material flow as well as the corresponding utilization of employees and equipment. Hence, this simulation run can be analysed with regard to material flow, utilization of employees, and total process time.

The constraint-based simulation concept is implemented using the simulation framework Plant Simulation from Siemens PLM Software II GmbH and the Simulation Toolkit for Shipbuilding (STS). The STS was developed by Flensburgers Shipbuilding, the SimCoMar cooperative agreement (Steinhauer, 2006), the Bauhaus-University Weimar, and the Ruhr-University Bochum. The components of the STS are presented in detail in König *et al* (2007).

Table 2 Developing execution strategies ordered by defensive strategic aspects (Beißert *et al*, 2008a)

Structural aspects	*Spatial aspects*	*Productive aspects*	*Qualitative aspects*
Avoid soiling	Closeness	Production flow	Quality control
Avoid damage	Distance	Human Strain Factor	
Avoid interference	Adjacency	Working spaces	
	Orientation		

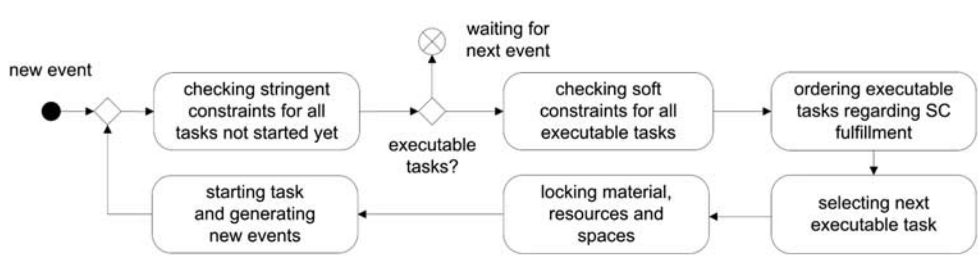

Figure 1 UML diagram of Starting Tasks.

4. Soft constraints for construction scheduling

Generally, in the context of Constraint Satisfaction a Soft Constraint specifies a practicable or advisable restriction. Therefore, Soft Constraints are particularly suitable for representing implicit knowledge such as execution strategies. Different concepts can be applied to model Soft Constraints, such as Weighted Constraints, k-Weighted Constraints, or Fuzzy Constraints (Rossi *et al*, 2006; Beißert *et al*, 2008b).

The varied types of Soft Constraint differ in their manner of representation, which affect the calculation of Soft Constraint fulfilment. In many cases, so-called Weighted Constraints are used to describe such advisable restrictions. Thereby, the Soft Constraints or involved objects, such as construction tasks, resources or working spaces, are weighted (Beldiceanu and Petit, 2004). If a Weighted Constraint cannot be completely fulfilled, the weights can be used to calculate a violation cost factor (Rossi *et al*, 2006). In contrast to Weighted Constraints, the k-Weighted Constraints have a threshold k that specifies an upper bound for the calculation of the cost factor (Beißert *et al*, 2008b). Consequently, different schedules can be evaluated regarding their strategy fulfilment by calculating the schedule's total cost factor due to constraint violations. In this paper Weighted and k-Weighted constraints are used to model the construction strategies *Avoid Soiling* and *Human Strain Factor* as introduced in the following subsections.

4.1. Avoid Soiling

The strategy *Avoid Soiling* specifies that *dirty* construction tasks should be executed before *clean* ones. This avoids extra work like cleaning processes during the working day. This unplanned extra work often causes disturbances within the execution progress and therefore adds time and costs. This strategy is applied in daily execution planning to generate schedules where succeeding processes preferably have the same degree of soiling (DoS) to avoid additional cleaning processes. A sequence of tasks with decreasing DoS is favourable, but a switch to higher levels is still possible, even though it contradicts the strategy.

The strategy *Avoid Soiling* is modelled by using Weighted Constraints. Therefore, a DoS is associated with each construction task. Each DoS, such as dirty, less dirty, clean, and very clean, is represented by a weight factor w. Table 3 shows practicable DoS and their associated weights.

To implement this strategy efficiently, the construction site is represented by a regular grid. Within the grid each cell status indicates its current DoS. Assuming that each task requires a certain working space in order to be executed, the required DoS status of cells can be determined. Consequently, the Soft Constraint fulfilment can be checked. Whenever a task is started, the DoS status of the associated cells is updated and changed to the DoS of the currently executed task.

Table 3 Degree of soiling (DoS) for construction tasks (cf Beißert *et al*, 2008a)

DoS	*Dirty*	*Less dirty*	*Clean*	*Very clean*
Weight	4	3	2	1

The strategy constraint *Avoid Soiling* is specified by $w_x \leqslant w_y$. This means that the degree of soiling of the predecessor task x should be lower than or equal to the degree of soiling of its successor task y. Soft Constraint fulfilment is determined by the calculation of an associated costs factor cf. Whenever the constraint is fulfilled, no additional costs arise. If on the other hand the Soft Constraint is not fulfilled, costs are compiled that are associated with the successor tasks y. The costs are calculated using the weight of the requested working space and the weight of succeeding task (cf Equation (1)). The strategy *Avoid Soiling* specifies that the more the strategy is violated the higher the costs are. In summary, the task order that causes minimal costs is the preferred execution alternative with regard to this strategy.

$$C_w(y, G_y) := y \in T_s \Rightarrow cf_y = \begin{cases} (\Delta s)^2 & \text{if } \Delta s \geqslant 0 \\ 0 & \text{else} \end{cases} \quad \text{with } \Delta s = s_{avg}(G_y) - s_y \tag{1}$$

where y is next executable task; G_y is working space of task y; T_s is set of next executable tasks; s_y is degree of soiling of task y; s_{avg} is average degree of soiling; cf_y is constraint violation costs of task y.

4.2. Human Strain Factor

The construction strategy *Human Strain Factor* considers the daily strain of participating workers. The strain affecting workers is caused by the execution of construction tasks. Each execution task demands physical as well as mental work by the executors (cf Figure 2). It has been shown that high worker stress leads to lower productivity, which entails additional costs and execution time. Furthermore, high worker stress results in quality defects in the final product. Observing the strategy *Human Strain Factor* helps to avoid placing excessive demands on workers and helps to ensure uniform utilization of the workers.

The *Human Strain Factor* strategy considers the workers' daily strain caused by the assigned construction tasks. The *Human Strain Factor* strategy can be modelled by k-Weighted Constraints. As already mentioned, each Soft Constraint violation causes costs. The threshold k is used to specify an acceptable limit for accruing costs. In this case, the threshold k_d specifies the tolerable daily strain of workers.

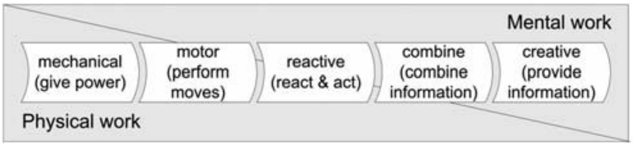

Figure 2 Classification of Human Strain according to Luczak (1998).

```
procedure grasp()
    while GRASP stopping criterion not satisfied do
        solution ← ConstructSolution();
        solution ← LocalSearch(solution)
        bestSolution ← UpdateSolution(solution)
    end;
    return(bestSolution)
end grasp;
```

Figure 3 Pseudo-code of the GRASP metaheuristic (cf Feo and Resende, 1995).

For each construction task an associated strain weight w_h is defined. Currently, only strain caused by physical work is considered. The daily workers' strain w_d is determined by the sum of the strain weights of all assigned tasks that are executed by a worker on that day. The task's strain weights w_h, daily strain assignments of workers w_d, and the daily strain threshold k_d are specified in kJ/min. Several investigations (Spitzer *et al*, 1982) provide basic statistics to describe task strain as well as thresholds for worker demand. If the daily strain w_d of a worker exceeds the specified threshold k_d, the *Human Strain Factor* strategy is violated and total strategy costs arise as shown in Equation (2).

$$C_{kw}(x, R) := x \in T_s \Rightarrow cf_x = \begin{cases} w_h & \text{if } \Delta w \leqslant 0 \\ w_h - \Delta w & \text{if } \Delta w - w_h < 0 \quad \text{with } \Delta w = k_d(R) - w_d(R) \\ 0 & \text{else} \end{cases} \tag{2}$$

where x is next executable task; T_s is set of next executable tasks; R is set of associated workers; cf_x is constraint violation costs of task x; $w_d(R)$ is daily strain of workers; $k_d(R)$ is maximum daily strain.

5. Optimization of execution strategies using GRASP

Valid and efficient execution schedules can be generated by different optimization concepts. In this paper the GRASP is applied and integrated into the constraint-based simulation concept. GRASP is a well-established metaheuristic for solving hard combinatorial optimization problems (Feo and Resende, 1995). The GRASP heuristic is an iterative heuristic consisting of two consecutive phases: solution construction and solution improvement (cf Figure 3).

The solution construction phase generates a feasible start solution. The best current elements are added successively to the partial solution. In the next step (solution improvement) a local search algorithm is used to improve the start solution by investigating its current neighbourhood. Whenever an improved solution is found, this solution is updated as the best current solution. Both phases are executed iteratively until a certain GRASP stopping criterion is met (Resende and Ribeiro, 2003). In this case, the GRASP heuristic stops when a given number of iterations are reached without any improvement.

Considering the presented execution strategies *Avoid Soiling* and *Human Strain Factor,* the GRASP heuristic is used to support the selection of the next executable tasks within the *Starting Tasks* routine (see Figure 4). On the basis of the *Avoid Soiling* strategy the GRASP heuristic can be used to determine a daily-optimized task combination, which causes minor deviation from the predominating degree of soiling. Similarly, the GRASP heuristic can be used to generate daily-optimized schedules concerning the strain utilization of the workers. Currently, only one strategy can be selected at a time for daily optimization. Consequently, this leads to a single objective function.

Whenever a new working day is started, the next executable tasks and their successors are determined. Afterwards an optimized execution order with associated worker assignment for the impending tasks is identified using the GRASP heuristic. For that purpose, the Soft Constraint violation costs are calculated. This means that the task combination with minimal costs is determined for each day. In the next section the implementation of the two phases for the optimization of daily-improved construction schedules is presented.

5.1. Construction phase

The GRASP construction phase generates an initial schedule for a working day. For each working day the cell status and daily strain of workers are reset to their default value. Consequently, all next-executable tasks are calculated by checking their Hard Constraint fulfilment. All executable tasks are stored in a restricted *candidate list* (*RCL*) and their processing status is updated. All listed candidates are temporarily marked as finished (cf Figure 5a). Afterwards, all not-started tasks are checked again for Hard Constraint fulfilment. This brings up some new *RCL* candidates. Subsequently, the list is updated and all additionally included tasks are also marked as finished (cf Figure 5b). Simultaneously, a topologically ordered graph G_{dt} (cf Pahl and Damrath, 2001) is created based on the candidates of *RCL* and the status-update steps. The procedure of checking Hard Constraints as well as adapting the candidate list *RCL* and the daily task graph G_{dt} is repeated until the minimal path length of the graph G_{dt} exceeds the specified working time of the current analysed day (cf Figure 5c).

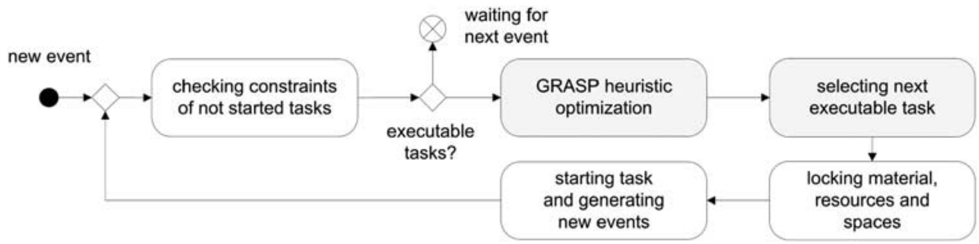

Figure 4 GRASP heuristic within constraint-based simulation.

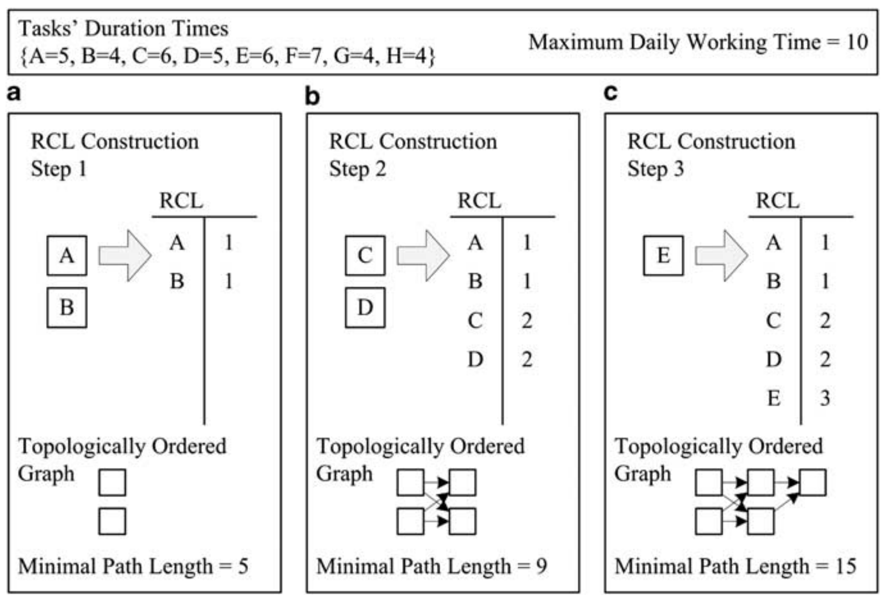

Figure 5 Generation of a daily restricted *candidate list* (RCL).

The initial solution for the current working day is determined based on the candidate list *RCL*. The resources are successively assigned to the tasks according to the topological order of the graph G_{dt}. To give an example, the first candidate of rank 1 is randomly selected and added to the current candidate combination J^*, and its required resources as well as work spaces are locked. For the selected strategy the total constraint violation costs are calculated and updated. Consequently, regarding the *Avoid Soiling* strategy the soiling of the requested working space is analysed. For the *Human Strain Factor* strategy, on the other hand, the daily strain of the assigned workers has to be taken into account. Furthermore, the *RCL* has to be refreshed by deleting all candidates that do not fulfil the Hard Constraints any more. Candidates are added successively to the current combination based on their ranks as well as the release of assigned workers. Finally, the GRASP construction phase stops if no more tasks can be scheduled or if the candidate list *RCL* is empty.

5.2. Local search

The GRASP local search phase is based on a standard Tabu search algorithm (cf de Werra and Hertz, 1989; Grabowski and Wodecki, 2005). Thereby, the current candidate combination J^* is used as the initial combination. A neighbouring solution is generated by randomly exchanging one candidate of the combination J^* with another candidate from the *RCL* (cf Figure 6). A task substitution can only be applied if all Hard Constraints are still satisfied. All substitutions are stored in a Tabu list and cannot be used within further local search steps. This results in the total constraint violation costs being updated. Once an improved combination J is found, the current solution is updated ($J^* = J$) and the search process is continued. The Tabu search stops when a given iteration number is reached without any improvement.

The calculated current solution then represents the best combination of succeeding tasks with regard to the *Avoid Soiling* strategy or the best assignment of workers to tasks for that day considering the *Human Strain Factor* strategy.

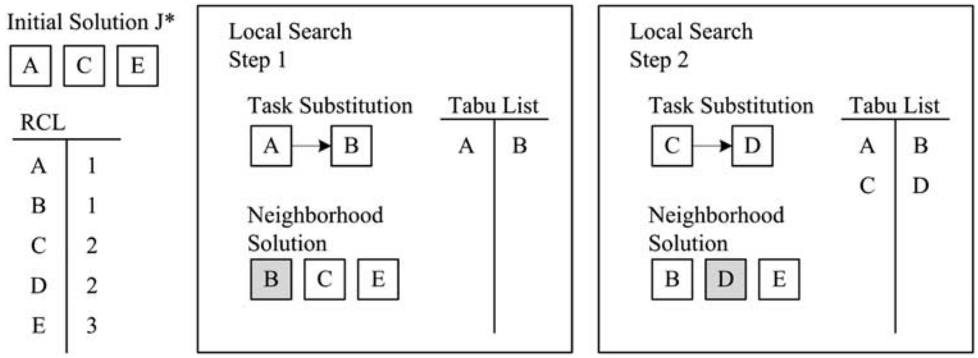

Figure 6 Generation of a neighbouring solution.

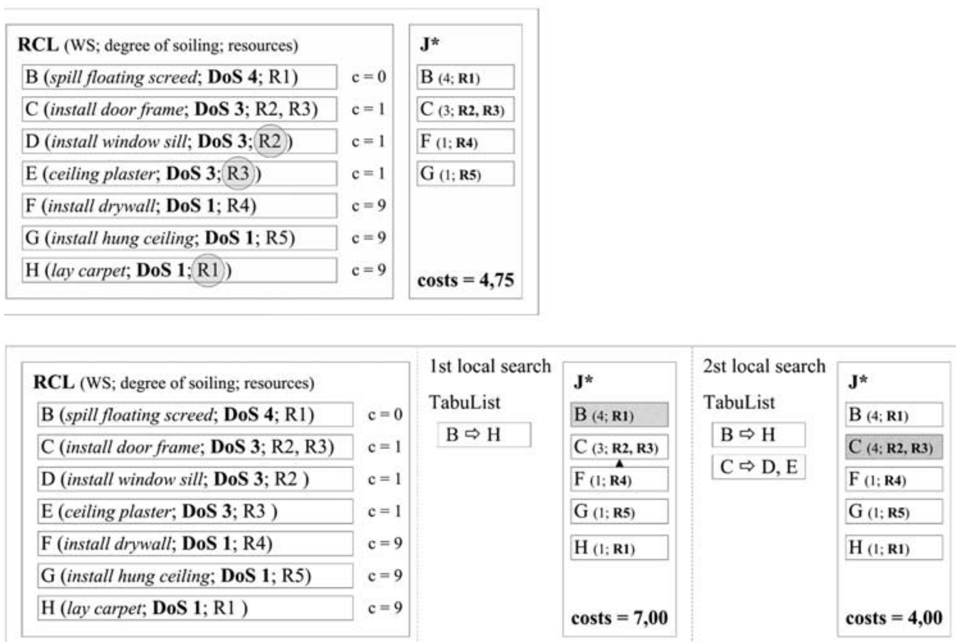

Figure 7 Example of the two GRASP phases relating to execution strategy *Avoid Soiling*.

In the following paragraphs the GRASP procedure is presented in more detail, taking the strategy *Avoid Soiling* as an example (cf Figure 7). Different construction tasks A, B, C, D, E, F, and G such as spilling floating screed, installing window sills, and laying carpet, which are all being executed in almost the same working area, have to be scheduled. The Hard Constraints of all construction tasks are fulfilled and the tasks are stored in the candidate list $RCL = \{B, C, D, E, F, G, H\}$ (cf Figure 7). For each construction task, the DoS is defined; for example, dirty processes get a weight of 4 and clean processes are weighted with 1. Furthermore, the current DoS of the working area has to be calculated. In this example the working space is specified as *dirty* and weighted with 4. As a consequence, all tasks could be executed simultaneously assuming that sufficient resources such as workers and material are available. However, a set of restricted resources $R = \{R1, R2, R3, R4, R5\}$ is given for this example, and consequently not all tasks can be executed simultaneously.

Within the construction phase the Soft Constraint costs of these candidates are calculated and listed in ascending order. First, the best candidate $\{B\}$ is added to the candidate combination J^*, and all the resources it requires are locked. In the next step all candidates with Hard Constraints that are not any longer fulfilled, for example caused by temporarily unavailable resources, are deleted from the candidate list (such as task H). Thus, a valid candidate combination $J^* = \{B, C, F, G\}$ is established causing an average cost factor of 4.75.

In the following local search phase an improved candidate combination based on the current solution J^* is generated by substitution of single candidates. In the first local search step the candidate B is exchanged with candidate H. This substitution is stored in the Tabu list. This new candidate combination $\{C, F, G, H\}$ leads to an average cost factor of 7. The new combination fulfills the strategy *Avoid Soiling* to a lesser degree than the solution computed before, so that the search process is continued based on the current solution.

In the second search step candidate B cannot be exchanged with other candidates. The next candidate C is investigated and replaced with the candidates D and E. The new candidate combination $\{B, D, E, F, G\}$ causes average costs of 4. The current solution is updated because the new costs are lower than the current costs. This local search procedure is repeated until all combinations are investigated. Finally, a new initial combination is randomly generated by the GRASP construction phase. Successively, both phases are executed until the GRASP stopping criterion is reached.

Comparing the *Human Strain Factor* strategy contrary to the *Avoid Soiling* strategy, the accruing costs of a certain resource assignment to a task is decisive for the next task combination. During the first construction phase low-duty resources are preferably assigned to tasks. The average of the workers' assignment costs of a generated solution is used to evaluate the current solution. Whenever the GRASP stopping criterion is met, the best current solution is going to be started. Successively, all combined tasks from this solution are executed, occupying their assigned resources.

6. Case study

To illustrate the performance and the practicability of the presented concept, the formalized execution strategies *Avoid Soiling* and *Human Strain Factor* are evaluated. Taking into account the GRASP heuristic concept for optimization and its implementation the scheduling of three different finishing trades of a storey of an office building is exemplarily calculated. The storey consists of 12 rooms built with 13 drywalls (cf Figure 8).

The finishing trades are floor covering, drywall construction, paperhanging, and painting. For each trade different construction task types are specified. The finishing trades, construction task types, and required resources to execute the task types, as well as their associated soiling degrees and strain weights, are shown in Table 4.

Figure 8 Finishing elements of building storey.

Other simulation input data like Hard Constraints, material elements, the construction site, and resources, for example employees and equipment, can in general be defined either manually or automatically (Beißert *et al*, 2007). The modelling and generation of these data is highlighted in Beißert *et al* (2007) and König *et al* (2007). On the basis of the given number of rooms and drywalls as well as the specified finishing trades, 365 single tasks and 1923 associated Hard Constraints were considered. For the case study six floorers, four drywallers, and four painters were specified. The daily working time of each worker was set to 8 h. In the following paragraphs the results of two simulation experiments are presented and evaluated.

Two different simulation studies were performed. In the first study valid schedules were determined by constraint-based simulation considering the strategy *Avoid Soiling*. The second study considered the strategy *Human Strain Factor*. In both studies the result of the GRASP optimization experiment was compared with results of the Monte Carlo simulation experiments. The Monte Carlo simulation is based on the random selection of next executable tasks during the *Starting Tasks* procedure (cf Figure 1). An adequate amount of Monte Carlo experiments has to be executed to provide a significant set of solutions. In this case study, 2000 Monte Carlo experiments were performed and evaluated statistically. The total execution time, the so-called makespan without breaks or weekends, and the total strategy violation costs were recorded.

The GRASP heuristic calculates an optimized sequence of the next executable tasks during the adapted procedure *Starting Tasks* (cf Figure 4). Consequently, the GRASP experiment is executed once for each strategy. The stopping criterion of the GRASP procedure and the stopping criterion of the Tabu search are implicitly defined in relation to the number of construction tasks of the current candidate list *RCL*.

The results of the first simulation study, considering the *Avoid Soiling* strategy, are shown in Table 5. The standard deviation of the Soft Constraint violation costs of the Monte Carlo experiments is significant. The minimum makespan of the Monte Carlo experiments leads to *Avoid Soiling* costs of 741. An average schedule of the Monte Carlo Simulation experiments has a makespan of 527 h and constraint violation costs of 823. The makespan of the GRASP optimization experiment is almost equal to the average makespan of the Monte Carlo simulations. The constraint violation costs of the GRASP experiments are significantly lower than the average costs of the Monte Carlo runs and even a little lower than the minimal Monte Carlo violation costs. By comparing the results, it can be noted that an efficient schedule can be generated by applying the GRASP heuristic considering the *Avoid Soiling* strategy.

For the second simulation study, resulting from physical work the maximum daily strain k_d for each worker is set to 8000 kJ/day (Spitzer *et al*, 1982). If the daily strain of a

Table 4 Finishing trades and Soft Constraint weights

Finishing trades	*Task types*	*Resource requirement*	*Soiling weight*	*Strain weight kJ/min*
Floor covering	Levelling	Floorer	4	11.9
	Priming	Floorer	3	18.9
	Laying	Floorer	1	12.2
Drywall construction	Calibrating	Drywaller	1	06.5
	Assembling	Drywaller	2	21.0
	Plastering	Drywaller	3	08.5
Paperhanging and painting	Plastering	Painter	3	12.5
	Paperhanging	Painter	2	14.4
	Painting	painter	3	13.0

Table 5 Evaluation of simulation study 1

First simulation study		*Monte Carlo*	*GRASP*
Makespan [h]	Minimum	506	523
	Average	527	—
	Standard deviation	25	—
Avoid Soiling violation costs [–]	Minimum	522	511
	Average	823	—
	Standard deviation	278	—
	Minimum makespan	741	—

Table 6 Evaluation of simulation study 2

Second simulation study		*Monte Carlo*	*GRASP*
Makespan [h]	Minimum	506	524
	Average	527	—
	Standard deviation	25	—
Human Strain violation costs [–]	Minimum	2486	2107
	Average	2872	—
	Standard deviation	114	—
	Minimum makespan	2581	—

worker exceeds this maximum daily strain, constraint violation costs are generated. The results based on the *Human Strain Factor* strategy are shown in Table 6. The average Soft Constraint violation cost of the Monte Carlo experiments amounts to 2872 with a standard deviation of 114. The constraint violation costs of the Monte Carlo result with minimal makespan are 2581. Consequently, in this case study an average schedule results in very high *Human Strain Factor* violation costs. The GRASP optimization experiment leads to a construction schedule with an average makespan of 524 h. However, the constraint violation costs are about 15% lower than the minimal violation costs of the Monte Carlo experiment. It can be noted that for this building storey example, as well as for the considered tasks and workers, an efficient schedule has been generated with regard to the makespan and the *Human Strain Factor* strategy.

7. Conclusion and outlook

In this paper a new approach to the simulation of construction schedules and their execution strategies is presented. The *Avoid Soiling* and *Human Strain Factor* strategies are formalized in detail by using so-called Weighted and k-Weighted Constraints. Constraint-based simulation is used to schedule construction tasks based on a selected execution strategy. The GRASP heuristic approach is integrated to calculate efficient daily construction schedules with regard to the specified strategy. A computational comparison with a Monte Carlo simulation has been performed in order to study the applicability of the presented approach. The results show that the presented strategy-modelling concept, which also implements the GRASP heuristic, allows for an efficient and purposeful generation of knowledge-based construction schedules.

Further research work is planned to integrate other human strain aspects such as mental or environmental strains into the *Human Strain Factor* strategy approach. Furthermore, this research will model relaxation times of workers as well as the definition of individual strain attributes for the workers in order to support more precise execution planning. Currently, only one strategy can be considered at a time within one simulation run. The possibility of combining different execution strategies should be explored in future work.

Acknowledgements—We gratefully acknowledge the financial support of the German Research Foundation (Deutsche Forschungsgemeinschaft—DFG) for this project.

References

AbouRizk S, Halpin DW and Mohamed Y (2006). Modeling construction operations using CYCLONE based systems. In: Wenzel S (ed). *Simulation in Produktion und Logistik, Proceedings of the 12th ASIM symposium 2006*. Society for

Modeling and Simulation International and SCS Publishing House e.V.: San Diego/Erlangen, pp 15–31.

Akbas R (2004). *Geometry-based Modeling and Simulation of Construction Processes*. Technical Report 151. Center for Integrated Facility Engineering (CIFE), Stanford.

Akinci B, Fischer M, Levitt R and Carlson R (2002). Formalization and automation of time-space conflicts analysis. *J Comput Civ Eng* **16**(2): 124–134.

Beißert U, König M and Bargstädt H-J (2007). Constraint-based simulation of outfitting processes in building engineering. In: Rebolj D (ed.). *Proceedings of the CIB 24th W78 Conference*, Faculty of Civil Engineering: Maribor, Slovenia, pp 491–497.

Beißert U, König M and Bargstädt H-J (2008a). Execution strategy investigation using soft constraint-based simulation. In: IABSE – AIPC IVBH (ed.). *IABSE Conference, Information and Communication Technology (ICT) for Bridges, Buildings and Construction Practice*, ETHZ Hönggerberg: Zürich, Switzerland.

Beißert U, König M and Bargstädt H-J (2008b). Generation and local improvement of outfitting schedules using constraint-based simulation. *Proceedings of the 12th International Conference on Computing in Civil and Building Engineering (ICCCBE-XII)*, Tsinghua University: Beijing, China.

Beldiceanu N and Petit T (2004). Cost evaluation of soft global constraints. In: Régin J-C (ed). *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*. Springer Verlag: Berlin.

Chahrour R and Franz V (2004). Computersimulation im Baubetrieb—Forschungsstand, innovative Einsatzmöglichkeiten. In: Mertins K (ed). *Experiences from the Future: New Methods and Applications in Simulation for Production and Logistics*, Proceedings of the 11th ASIM Symposium 2004, Fraunhofer IRB-Verlag: Stuttgart, Germany.

Choo HJ, Tommelein ID, Ballard G and Zabelle TR (1999). WorkPlan: Constraint-based database for work package scheduling. *J Constr Eng Mngt* **125**(3): 151–160.

Feo TA and Resende MGC (1995). Greedy randomized adaptive search procedures. *J Global Optim* **6**: 109–133.

Fischer M and Kunz J (2004). *The Scope and Role of Information Technology in Construction*. Technical Report 156. Center for Integrated Facility Engineering (CIFE), Stanford.

Franz V (1989). *Planung und Steuerung komplexer Bauprozesse durch Simulation mit modifizierten höheren Petri-Netzen*. PhD Thesis, University of Kassel.

Freuder EC and Wallace R (1992). Partial constraint satisfaction. *Artif Intell* **58**: 21–70.

Grabowski J and Wodecki MA (2005). Very fast Tabu search algorithm for job shop scheduling. In: Rego C and Alidaee B (eds). *Metaheuristic Optimization Via Memory and Evolution—Tabu Search and Scatter Search*. Kluwer Academic Publishers: Boston, pp 117–144.

Halpin DW and Martinez LH (1999). Real world applications of construction process simulation. *Proceedings of the Winter Simulation Conference*, Society for Computer Simulation International: San Diego, pp 956–962.

König M, Beißert U, Steinhauer D and Bargstädt H-J (2007). Constraint-based simulation of outfitting processes in shipbuilding and civil engineering. *6th EUROSIM Congress on Modeling and Simulation*, CD-ROM Publication.

Luczak H (1998). *Arbeitswissenschaft* 2nd edn. Springer Verlag: Berlin.

Mallasi Z (2004). Identification and visualization of construction activities' workspace conflicts utilizing 4D CAD/VR tools. *Proceedings of the 1st ASCAAD International Conference e-Design in Architecture*, KFUPM: Dhahran, Saudi Arabia. pp 235–253.

Martinez JC (1996). *STROBOSCOPE–State and Resource Based Simulation of Construction Processes*. PhD Thesis, University of Michigan.

McKinney K and Fischer M (1998). Generating, evaluating and visualizing construction schedule with CAD tools. *Automat Constr* **7**(6): 433–447.

Mikulakova E, König M, Tauscher E and Beucke K (2008). Case-based reasoning for construction tasks. *Proceedings of the 12th International Conference on Computing in Civil and Building Engineering (ICCCBE-XII)*, Tsinghua University: Beijing, China.

Pahl PJ and Damrath R (2001). *Mathematical Foundations of Computational Engineering: A Handbook*. Springer Verlag: Berlin.

Pultar M (1990). Progress-based construction scheduling. *J Constr Eng Mngt* **116**(4): 670–688.

Resende MGC and Ribeiro CC (2003). Greedy randomized adaptive search procedures. In: Glover F and Kochenberger GA (eds). *Handbook of Metaheuristics*. Kluwer Academic Publishers: Boston, pp 219–250.

Rossi F, van Beek P and Walsh T (2006). *Handbook of Constraint Programming*, 1st edn. Elsevier: Amsterdam.

Spitzer H, Hettinger T and Kaminsky G (1982). *Tafeln für den Energieumsatz bei körperlicher Arbeit*, 6th edn. Beuth: Berlin.

Sriprasert E and Dawood N (2002). Requirements identification for 4D constraint-based construction planning and control systems. *International Council for Research and Innovation in Building and Construction—CIB w78 Conference 2002*, http://itc.scix.net (accessed 30 October 2007).

Sriprasert E and Dawood N (2003a). Genetic algorithms for multi-constraint scheduling: An application for the construction industry. *International Council for Research and Innovation in Building and Construction—CIB w78 Conference 2003*, http://itc.scix.net (accessed 30 October 2007).

Sriprasert E and Dawood N (2003b). Multi-constraint information management and visualization for collaborative planning and control in construction. *ITcon* **8**: 341–366.

Steinhauer D (2006). Simulation in shipbuilding—Supporting shipyard planning, production planning and product development. In: Wenzel S (ed). *Simulation in Produktion und Logistik, Proceedings of the 12th ASIM Symposium 2006*. SCS Publishing House: San Diego/Erlangen, pp 1–14.

Tulke J and Hanff J (2007). 4D construction sequence planning—New process and data model. In: Rebolj D (ed.). Proceedings of the CIB 24th W78 Conference Faculty of Civil Engineering: Maribor, Slovenia, pp 79–84.

Wakefield RR and Sears GA (1997). Petri nets for simulation and modeling of construction systems. *J Constr Eng Mngt* **123**(2): 105–112.

Wang HJ, Zhang JP, Chau KW and Anson M (2004). 4D dynamic management for construction planning and resource utilization. *Automat Constr* **13**: 575–589.

de Werra D and Hertz A (1989). Tabu search techniques. *OR Spektrum* **11**: 131–141.

Woodworth BM and Shanahan S (1998). Identifying the critical sequence in a resource-constrained project. *Int J Proj Mngt* **6**(2): 89–96.

Received 4 February 2010;
accepted 8 March 2010