Proceedings of the 2013 Winter Simulation Conference
R. Pasupathy, S.-H. Kim, A. Tolk, R. Hill, and M. E. Kuhl, eds

INTEGRATION OF SIMULATION AND PARETO-BASED OPTIMIZATION FOR SPACE PLANNING IN FINISHING PHASE

Trang Dang
Hans-Joachim Bargstaedt

Bauhaus Universitaedt Weimar
Marienstr. 7a, 99423 Weimar, Germany

ABSTRACT

In order to improve the flexibility and adaptability of an automation model to various different projects and under different circumstances, various solutions should be generated as its result, instead of only one solution as in recent researches. The paper therefore presents a method for generating a series of reasonable detailed schedules for mapping workplaces of activities over time. This model is an integration of Pareto-based optimization and simulation. The optimization engine takes a role for generating and choosing good schedules. The simulators which simulate the way managers dealing with problems, in the other side, are responsible to manipulate activities to resolve spatial conflicts, deal with limits of crews and investigate efficiency of a solution. A prototype implementation is then developed and implemented in software based on Building Information Modeling (BIM), which enables the model to automatically retrieve geometry data. The output solutions are finally analyzed through an example application to prove their feasibility and adaptability to various potential situations on site.

1 INTRODUCTION

Construction projects involve various subcontractors. Communicating the schedule and strategy among them is a problem on a construction site. This problem is especially serious in finishing stage of execution, when lots of stakeholders, such as construction, electric, pipe, facility and other subcontractors work at the same time in a limited space. Besides, tasks in this period have a lot of time slack, so there are various alternatives for subcontractors arranging the schedule to adapt their own conditions. If a detailed schedule with a map of subcontractors' work places over time is not created beforehand, the space disputes between them will occur. Through empirical studies of Mallasi et al., the lack of detailed planning has been indicated as a reason causing 30% non-productive time (Mallasi and Dawood, 2001). It has been also stated as one of main reasons of stacking of trades (Hana, Russell and Emerson, 2008). Generating a sufficiently detailed short-term schedule, therefore, is necessary for finishing period, when several trades work together at the same time in a limited space.

A challenge for establishing an efficient scheduling method is its adaptability. Every construction project has its own different characteristics such as its contractor efficiency and capability to mobilize finance. In some cases, for example, in order to resolve a problem, compressed schedules can be considered as a solution if project managers are able to mobilize extra crews. In other cases, however, it is easier to deal with the problem by letting some activities suspended to give their work places to others first, if extra costs are acceptable, etc. Therefore, it is difficult to find a solution which can fit every different context. This paper proposes a method which is able to generate various alternatives for arranging activities to deal with spatial conflicts. From that, construction managers can choose the best adaptive one which is suitable to their situation.

The model presented is an integration of simulation and heuristic optimization. Firstly, two simulators, which simulate the way managers dealing with conflicts on site, and an optimization engine are developed. They play a role as the control center of the model. The simulators respond to analyze the behavior of schedules considering workspace and crew requirements, investigate how much spatial congestion occurs and how these congestions may impact on the schedule in return. In this process, tasks are considered able to be interrupted (if possible). The optimization engine will base on the evaluation results of the simulation process to generate a set of schedules those are "near" Pareto front. Finally, feasibility and flexibility of solutions given out are analyzed via an example application.

2 RELATED RESEARCH AND MOTIVATION

Many researchers have been interested in space planning and advocated significant works in this area. Bargstädt, Elmahdi et al. presented a model based on discrete event simulation to allocate workspaces and resolve time-space conflicts (Bargstädt and Elmahdi, 2010, Bargstädt and Elmahdi, 2010, Elmahdi, Wu and Bargstädt, 2011). Riley et al. suggested a method for space planning (Riley and Sanvido, 1997, Riley and Sanvido, 1995). Akinci et al. built a 4D-CAD based model to generate workspace requirements, detect and prioritize potential time-space conflicts (Akinci, Fischen, Levitt and Carlson, 2002, Akinci, Fischer and Kunz, 2002, Akinci, Fischer, Kunz and Levitt, 2002). Dawood and Mallasi developed a model to investigate and measure the severity of spatial congestions and then searched for a strategy to minimize this problem (Mallasi, 2006, Dawood and Mallasi, 2006). Winch et al. proposed the concept of critical space analysis (CSA) (Winch and North, 2006), Zhang et al. suggested a cell-based model in a discrete-event simulation (Zhang, Hammad, Zayed, Wainer and Pang, 2007) and Bansal applied GIS and Topology to identify and overcome this issue (Bansal, 2011).

All of these researches confirm the necessity of workspace management during planning and scheduling as well as the importance of visualization in detecting workspace conflicts. This allows users to aware of situations which are occurring on site in advance and then make decision based on their experience and intuition. Beside researches those just focus on generating workspace requirements and detecting workspace clashes (Akinci, Fischen, Levitt and Carlson, 2002, Akinci, Fischer and Kunz, 2002, Akinci, Fischer, Kunz and Levitt, 2002), the others also consider about searching for strategies to overcome this problem. Currently, there are two directions in dealing with spatial conflicts. One is to create a schedule which eliminate all potential spatial congestions (Bargstädt and Elmahdi, 2010, Bargstädt and Elmahdi, 2010, Elmahdi, Wu and Bargstädt, 2011, Zhang, Hammad, Zayed, Wainer and Pang, 2007, Jongeling, 2006); and the other is to adjust a schedule to keep the congestion minimum (Mallasi, 2006, Dawood and Mallasi, 2006, Winch and North, 2006, Bansal, 2011).

Application of the conflict-free method for look-ahead planning may be not feasible. In order to keep the flexibility of planning, detailed schedules for three weeks look-ahead are normally developed as an expansion of a part of macro-schedules in execution phase. At this time, the time frame of a project is already determined. Manipulating activities to eliminate all workspace conflicts may cause over-controlled delay of the project. This makes the solution of the method insufficient. The research in this paper follows the later method, keeping congestions minimal within an acceptable time frame of the project.

In order to search for schedules with minimal congestion, researchers have different algorithms. Dawood and Mallasi integrated a simple genetic algorithm with different work rates and execution patterns to find out the best solution. However, they considered start dates of activities fixed; such the assumption makes the approach not able to receive benefits from time slacks of activities. Winch and North used the “brute force” algorithm; that means they investigated almost all of possibilities of adjusting schedule for choosing the best one. Bansal suggested a model, in that user can manually adjust a schedule, spatial requirements or split activities to find a suitable solution. These methods are time-consuming and not compatible with the large-scale searching like this problem.

Moreover, all of these approaches, except the manual model of Bansal (Bansal, 2011), consider tasks not able to be interrupted. It means that a task once starts; it will occupy resources, e.g. workspace, until it finishes. This reduces number of potential schedules to search for and also not able to reflect the fact of the priority of critical tasks. In reality, some tasks must be interrupted to give resources to those belonging to the critical path when they need.

Another disadvantage of the researches mentioned above is its limitation of a number of results given out. All of them provide only one solution as a result. This limits the adaptability of researches in varied construction area, which are much different from this project to other project, even in the same project but with different stages, situations are different as well.

3 INPUT DATA SYSTEM

In order to enable the model automatically to invoke input data, a 4D model has been developed by connecting a schedule (2D) and a product model (Figure 1:). This issue has been already considered thoroughly in various previous researches (Tulke, 2010, Tauscher, 2011). So in this research, it is not considered much anymore. We just use a simple XML file as a data template about information interacting between a schedule and a product model. Then an activity can be assigned automatically with objects, which have right categories defined in the template and lay on the right floor as required. After that, based on kind of its trade and positions of its products, workspaces can be automatically generated as well (Dang, Elmahdi and Bargstädt, 2012). All of this kind of works has been programmed in Visual C# with the support of the namespaces Revit API and the control ActiveGantt.

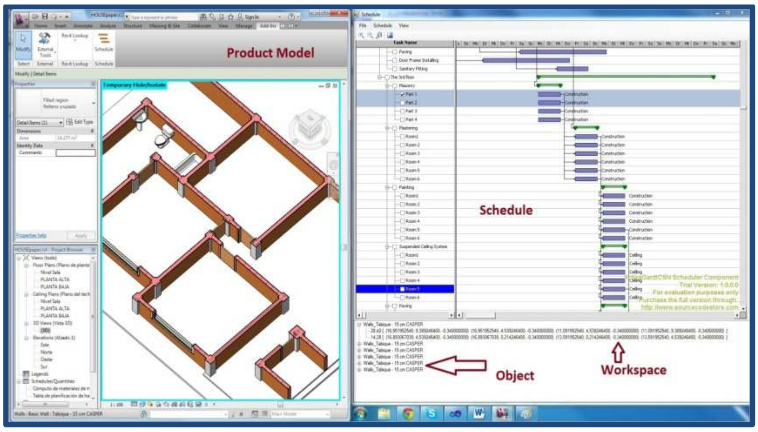

Figure 1: *4D Model developed in Revit Architecture environment*

In this model, data of an activity includes three types: 1) information of a schedule such as earliest start/end date, free slack, total slack, constraints with other activities, kind and number of crews required; 2) information of geometry data such as location of its products and location of the products' workspace correspondingly; and 3) task property which defines whether or not this task can be interrupted.

4 TIME-BASED SIMULATION MODEL

In order to investigate time-space conflicts and allocate workspaces to activities, a time-based simulation model has been developed.

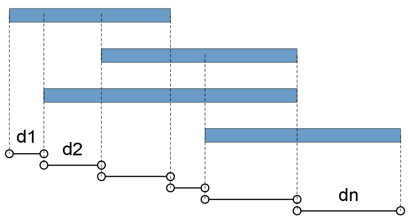

Figure 2: *The time interval of the time-based simulator*

In this model, the time interval is not constant during the simulation process. It is defined as the duration between the current simulated point and the nearest start/end date of activities investigated (Figure 2:).

During the simulation process, workspace and crew are considered as "near" hard constraints. It means a specific workspace and a crew are just served to only one activity at a point of time. If two or more tasks, according to the schedule, require the same space at the same time, only one task occupies this place and the others must be moved to later time; unless these tasks do not have enough time to be moved, due to the overrun of their total slacks and the accepted delay which is identified before the process, then they must stay there and the workspace conflicts will be counted in the procedure of the schedule evaluation. Similarly, if at a point of time, a crew requirement is over its capability, some tasks must be moved in order to deal with this inadequacy; unless they have no time buffers.

However, such the process causes a problem that lots of congestions sometimes seem have a tendency to meet together at a point, at which the tasks have no more slacks to be manipulated; therefore it is difficult to make a solution on site based on this kind of schedule. In order to diversify the results and distribute the conflicts suitably, the approach proposes two simulators: 1) the tense conflict simulation (TCS); and 2) the distributed conflict simulation (DCS).

The input data for these simulation processes includes the acceptable delay of the project as well, beside the information mentioned in the part of input data system. In both of TCS and DCS, tasks those have start-dates earlier will have a priority of receiving workspaces if disputes occur, unless tasks which have start-dates later have no time-buffer.

4.1 The tense conflict simulation (TCS)

The TCS considers that a task can be interrupted (if its property is interruptible) and conflicts just occur if the relative tasks have no longer time-buffers to be moved. The simulation process can be presented in detail as follows.

```
currentDuration is assigned as d1 (figure 2);
do
{
    1. List all tasks investigated taking place in the currentDuration
    2. If any couple of tasks in the list requires the same workspace && none of them has not been moved to later
    && at least one of their (total slack + acceptable delay) greater than zero, then
        a. choose the task to be allowed to occupy the workspace based on the following priority
            i. for the case: at least one of them is interruptible one
                1. its property is not interruptible && its start-date is earlier than currentDuration
                2. Its (total slack+accepted delay) is zero
                3. Its start-date earlier
                4. Its (total slack + accepted delay) smaller
                5. If they have the same (3) and (4) values, then choose one of them randomly
            ii.for the case: both of them are not interruptible && at least one of their start-dates is within
            currentDuration
                1. its start-date earlier
                2. Its (total slack + acceptable delay) smaller
                3. If they have the same (1) and (2) values, then choose one of them randomly
        b. move the not-completed part of the not-chosen task at the step (a) to the later time, so that after this
        step its (total slack + acceptable delay) is not smaller than zero and its new start-date is not later one
        day than the end-date of the chosen one.
        c. update the whole schedule based on its constraints after moving the not-chosen task.
    3. List all tasks investigated taking place in the currentDuration (some of tasks has been moved in the step 2).
    4. If any crew group is inadequate, list all the tasks being conducted in this period requiring this resource, then
        a. take a task out of the list if
            i. either its property is not interruptible && its start-date is earlier than currentDuration
            ii. or its (total slack + accepted delay) is zero
        b. for the tasks still existing in the list, choose some tasks to be moved so that the crew requirement is
        not greater than its capability and the number of tasks moved is minimum. In the case that the
        requirement is still greater than its capability after moving all tasks in the list, the crew overruns will be
        generated.
    5. currentDuration is assigned as the nextDuration (figure 2)
} while (currentPoint is not the lastPoint yet)
```

Figure 3: *The simulation process of TCS*

4.2 The distributed conflict simulation (DCS)

Like the TCS, the DCS also allows to interrupt a task (if possible) in the simulation process. Different from the TCS, however, a spatial congestion and a crew overrun in the DCS can occur even if the relative tasks still have time-buffers to be moved. In order to produce this kind of result, a change from TCS has been made. In the step 2, before choosing a task being allowed to occupy the disputed place, a random number will be created with 20% probability that they will together stay at that time; hence, the congestion will be generated. Of course, the number of the probability can be adjusted to be less or greater than 20% but for the experiments so far, 20% has been the reasonable for this problem.

5 PARETO-BASED OPTIMIZATION MODEL

The optimization model tries to adjust the schedule in order to find out the solution which has values of objectives, such as project lead time, conflict duration, and number of workers inadequate, etc., and acceptable. In order to deal with multiple objectives, there are three principle methods (Mumford-Valenzuela, 2005).

(1) Combine all the objectives into a single scalar value by using weighted factors corresponding to objectives, and optimize for the scalar value.

(2) Arrange the objectives in a priority order, optimize for the first objective, then if there is more than one solution, optimize these solutions for the second objective, and repeat for the third, etc. if appropriate.

(3) Consider all objectives equivalent; find a set of non-dominated solutions, in which when attempting to improve an objective further, the other objectives suffer as a result. This is called Pareto optimal and the set of non-dominated solutions is called Pareto front (Figure 4:).

The methods (1) and (2) give the only one solution. This may ignore good solutions which do not have the best value for a particular objective, but it makes good scene if all objectives are considered together. With the idea to provide best information to managers, the approach chooses the method (3). Such a method involves no judgments and produces a set of viable alternatives, from which a decision maker can reach an informed selection at a later stage.

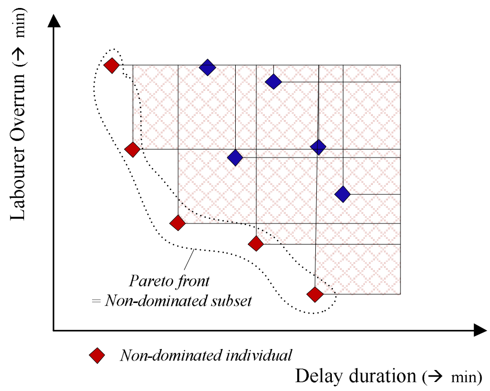

Figure 4: *Pareto optimality*

5.1 Objective Definition

Five parameters are taken into account in the approach. They can be categorized in two groups. The first group called project properties contains: project lead time, conflict duration and crew overrun. The second group, which is named feasibility properties, contains: split number and conflict number. The optimization process aims to keep the values of these five parameters minimum.

$$Objectives = \ to\ minimise \begin{cases} -\ project\ properties & \begin{cases} project\ lead\ time\ (1) \\ conflict\ duration\ (2) \\ crew\ overrun\ (3) \end{cases} \\ -\ feasibility\ properties & \begin{cases} split\ number\ (4) \\ conflict\ number\ (5) \end{cases} \end{cases}$$

5.1.1 Project Lead Time

Project duration is always one of the most important factors regarded in construction management. In this approach, this is also not an exception. The project duration, more precisely **the project lead time** in this case, is considered an important item to evaluate the goodness of a schedule. This value of the lead time always lies within the acceptable duration which is given as an input data before an investigating process. If the longer an acceptable duration is given, then the larger a searching domain is investigated. This issue is discussed more clearly in the section named application of SEAMO.

$$\textbf{\textit{Project lead time}} = \textit{actual end date of project} - \textit{as planed end date of project} \quad (1)$$

5.1.2 Conflict Duration

Although manipulating of an activity in simulation process is conducted to resolve workspace conflicts, spatial congestions still exist in a schedule depending on the acceptable delay duration. In this research **conflict duration** is regarded as the second objective the optimization process must take into account. This value is counted as the number of days in which the schedule contains workspace conflicts. The better solution is associated with the shorter conflict duration.

$$\textbf{\textit{conflict duration}} = \sum_{k=1}^{n} d_k \quad (2)$$

There, n is number of durations in which workspace conflicts occur; d_k is number of days of the k^{th} duration.

5.1.3 Crew Overrun

In the evaluation process, laborer overruns are also one of important things to evaluate a schedule. If this information is not taken into account, a good theoretical solution can be achieved by letting all activities, which are the same kind of trades and have difference workspace requirements, take place at the same time. Such this schedule, however, is not practical, since the number of crews is not infinitive and dealing with limits of workers is also a difficult problem in construction management on site. Therefore, in this approach, crew overrun is considered as the factor to evaluate a solution. This item is defined as a total number of laborers exceeding the capability.

$$\textbf{\textit{crew overrun}}\ (\%) = \max\left\{\frac{CR_i^k - CC_i^k}{CC_i^k} \times 100\right\}; i = 1 \div n; k = 1 \div m \quad (3)$$

There, n is the number of durations in which the laborer requirements are over the capability.
m is the number of crew groups which has the capability smaller than requirements
CR_i^k is number of Crew Requirements of the k^{th} crew group at the i^{th} duration
CC_i^k is number of Crew Capability of the k^{th} crew group at the i^{th} duration

5.1.4 Split Number

Beside of the objectives which present clearly negative consequences of a solution, like the three parameters referred above, some others do not. But they are very important to identify the feasibility of a schedule. One of them is relevant to the splitting number of a task. As mentioned in the simulation, an activity is once split means that it has to be suspended in order to give its work place to another task with higher priority. This may lead a necessary of reallocation of equipment and materials in consequence. Therefore, if a task has too much interruptions, the schedule corresponding is not feasible to be chosen as a solution. In order to take into account this problem, the parameter named split number will be considered.

$$\textbf{\textit{split number}} = \max\{\textit{the interruption number of the task } i\}, i = 1 \div n \quad (4)$$

There, n is the number of tasks investigated.

5.1.5 Conflict Number

Beside the split number, the number of tasks conflicting at one point of time should be kept less to make a solution feasible. The more number of tasks dispute each other for a same work place, the more difficult managers must deal with them. This parameter is called **conflict number**.

$$\boxed{\boldsymbol{conflict\ number} = max\{(number\ of\ tasks\ having\ same\ workspace\ at\ duration\ i) - 1\}\ (i = 1 \div n)} \quad (5)$$

There, n is the number of durations, which is referred in Figure 2:.

5.2 Chromosome Definition

A chromosome represents for a schedule (Figure 5:). The number of ADNs in a chromosome is one unit greater than the number of tasks which must be investigated considering workspace and crew requirements. Except the last ADN, which contains information about the acceptable lead time of the project for the simulation process, the others provide the information about the duration which is counted from the early start date to the actual start date of tasks. In this approach, an individual is also considered as a chromosome. So in this paper, the terms "chromosome" and "individual" can be used interchangeably.

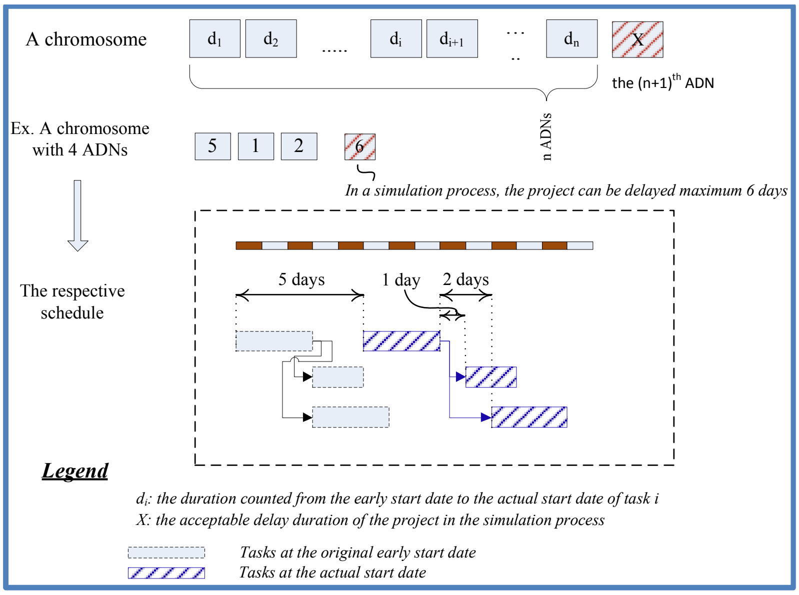

Figure 5: *Chromosome*

5.3 Original Generation

An individual in the original generation is created by take (n+1) random numbers. There, n is the number of tasks which is needed investigating. For the first n ADNs, di is defined by a random number which lies within zero and total slack of the task i. For the last ADN, X is identified by a not-negative random number which is not greater than the acceptable lead time of the project which is identified as an input data of the model.

According to what is presented in the part chromosome definition. Such the way to generate an individual may cause the project delay duration greater than the X value, after updating di to obtain actual dates of tasks. If this case occurs, X will be assigned again with the value of the project delay duration if it is not greater than the acceptable lead time of the project. Otherwise, this individual will be taken out of the population.

By using the X variable for each individual, it makes assure the diversity of population when the acceptable lead time is great. It should be noticed that, however, the greater the acceptable lead time is, the larger scale the population should be required.

5.4 Crossover Operator

A simple crossover operator is applied to generate an offspring. From two parent schedules, a crossover position is randomly chosen, the first parent provides the first part of schedule for the offspring, and the second provides the rest. The offspring also takes the last ADN from either of its parents. In the approach, all of individuals in the population join the crossover process. After this process, the offspring has the probability of 10% being followed by a mutation process before being analyzed and evaluated.

5.5 Mutation Operator

In the mutation operation, a position (x), number of tasks (n) and the day to be changed (t) will be randomly created first; notice that t may be either negative or positive. The mutation operator will be then applied for n continuous activities from the position x by moving them t days uniformly.

5.6 Selection Process

In order to generate next generations which are moved ever closer to the Pareto front, the simple evolutionary algorithm for multi-objective optimization (SEAMO) (Mumford, 2010) (Figure 6:), which was developed by Mumford, is adopted. This algorithm uses a replacement strategy to move the solutions during the searching process ever closer to the Pareto front, and to widen the spread of the solution set. Like the name it is called, this algorithm is kind of simple and such this property suits the approach. Honestly, the simulation process, which is used to evaluate a chromosome (or rather a schedule in this approach), takes time. Therefore, a strategy, which is simple but able to generate diversified solutions in the end, is priority to be chosen for the approach.

It should be mentioned that, this algorithm however has been modified a little bit in the approach. Instead of checking all individuals of a population, in order to know whether or not an individual exists which is dominated by the offspring, the approach just takes maximum 20% of a population randomly. Such the modification gives individuals an equal probability of being replaced regardless their positions in a population.

5.7 Working Mechanism of Optimization Engine

Optimization engine is described in Figure 6:. The process commences with generating an original generation. For each individual in this generation, the simulation process is then applied to evaluate its value. The record of the global best-so-far for objectives are created by taking the minimum number of its values from the individuals evaluated.

After that, all individuals take part in the crossover process and 10% of them continue with mutation to generate offspring. Once an offspring is born, it will be evaluated by a simulator and pass through the selection process to know whether or not it can exist by replacing an existed individual in the population. The optimization ends whether the possible number of evaluated generations is exceeded; or no new individual comes in the population.

It should be reminded again that two simulators are applied in the approach (TCS and DCS). If both of them are used for one population, by choosing a simulator to analyze a schedule randomly for example, the population required must be large and it takes long time to obtain the converged solutions. Therefore, two independent populations have been used. One uses TCS and the other uses DCS in the analyzing and evaluating process of a schedule. After these two optimization processes, the results will be combined together and a filtering process will be applied in order to take dominated chromosomes out of the population. As a result, a set of schedules which is considered “near” Pareto front is presented.

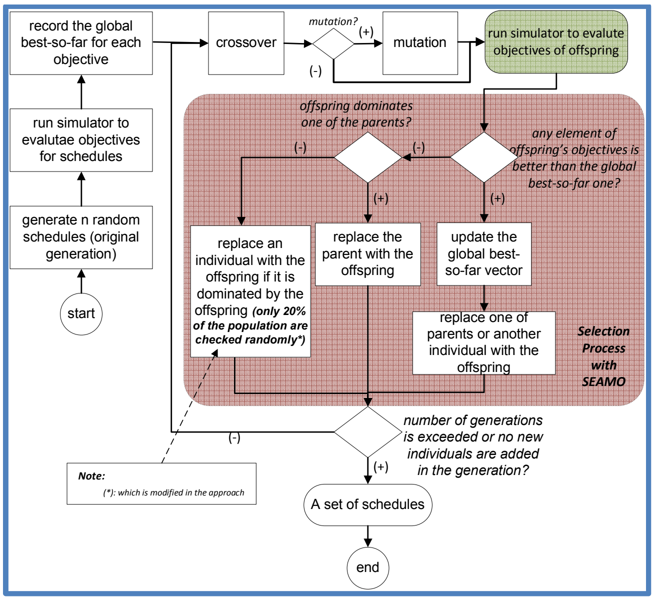

Figure 6: *Optimization Engine*

6 EXAMPLE APPLICATION

The model is experimented in the finishing period of a building floor. The trades involved in this experiment include masonry, plastering, painting, installing suspended ceiling system, installing windows and doors, paving and installing sanitary facilities.

The masonry trade contains 4 activities which respond to 4 regions of the walls; the sanitary fitting is completed with only one activity; and the others are divided into 6 activities corresponding to 6 different rooms. In summary, 35 activities are investigated.

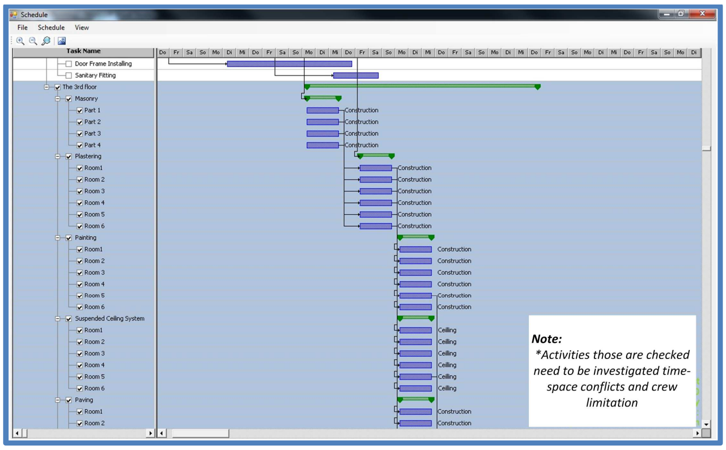

Figure 7: *The original schedule*

The optimization will be conducted with the acceptable lead time is 4 days, a population includes 50 individuals and the number of generations is 5.

As a result, a set of schedules is generated. Each value vector of objectives may contain several schedules.

It is also noticed that an objective vector contains 5 parameters and the graph is only able to show maximum three values at a point of time. Therefore, it is necessary to use the filter and the axis data setting to enable investigating all sides of solutions.

According to the objective vectors in the graph, the "near" Pareto-front solutions have the minimum delay duration of zero and the maximum delay of two days. With two days delay, the schedule has no time-space conflicts and crew inadequacies. Following are some schedules resulted are considered in order to analyze the efficiency of solutions under a construction manager's perspective.

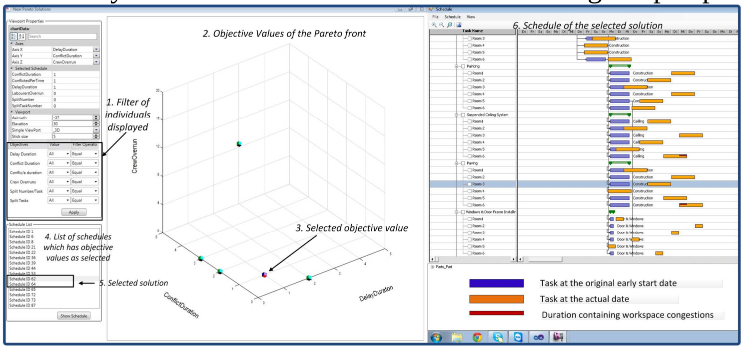

Figure 8: *Screen shot of a result*

Case 1: the delay duration is zero

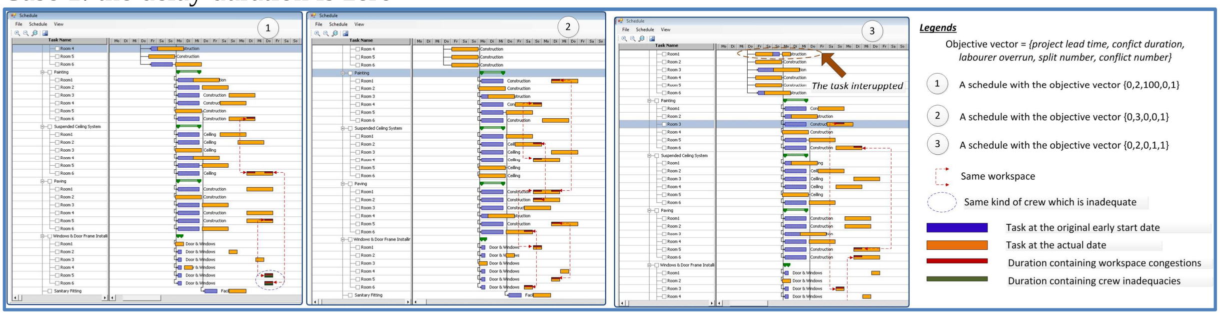

Figure 9: *Solutions without delay*

For the schedule number 1, the problems which managers must face include both of workspace congestions and the inadequacy of the crew for installing windows and doors with an amount of 100% (in this case it means two laborers). However, if the trade plastering at the room 1 can be interrupted like what the schedule number 3 has shown, the crew overrun would not occur. Therefore, only one problem to be deal is the workspace dispute.

For the schedule number 2, the crew requirements are always within their capability, no interruption of tasks are required. However, this schedule brings many spatial disputes. Especially, the conflict between installing suspended ceiling system and paving; this conflict is very serious since both of them normally require a whole room for their works. So this solution is not feasible.

Case 2: the delay duration is one day.

For the schedule number 4 (Figure 10:), a congestion occurs only in one day between the trades installing suspended ceiling system and installing windows and doors. If a one-day delay can be accepted, this solution is considerable since this kind of congestion can be solved on site.

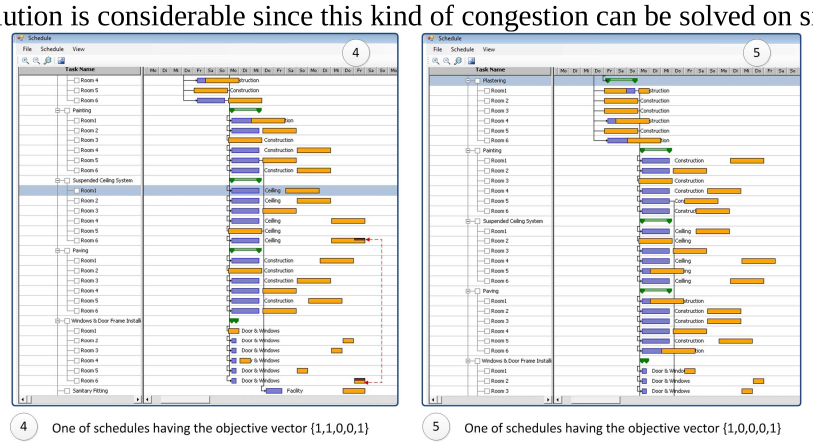

Figure 10: *Solution with one day delayed (4) and two days delayed (5)*

Case 3: the delay duration is two days

If it is the case, that two days of delay for the project is acceptable, this schedule (Figure 10:) is also feasible. It has no inadequacies of crews, no tasks must be interrupted and no time-space conflicts occur during the construction process.

7 DISCUSSION

The case study mentioned above is experimented with just 5 generations for the evolutionary process. The result would be really better if this case would be carried out with 10 generations. Although the delay duration of the project is still 2 days in order to deal with all of the problems. But when the delay duration is zero, the solution is more feasible. When the delay duration is one day, just with an interruption of the trade plastering at room 1, no longer workspace congestions or crew inadequacies occur (Figure 11:).

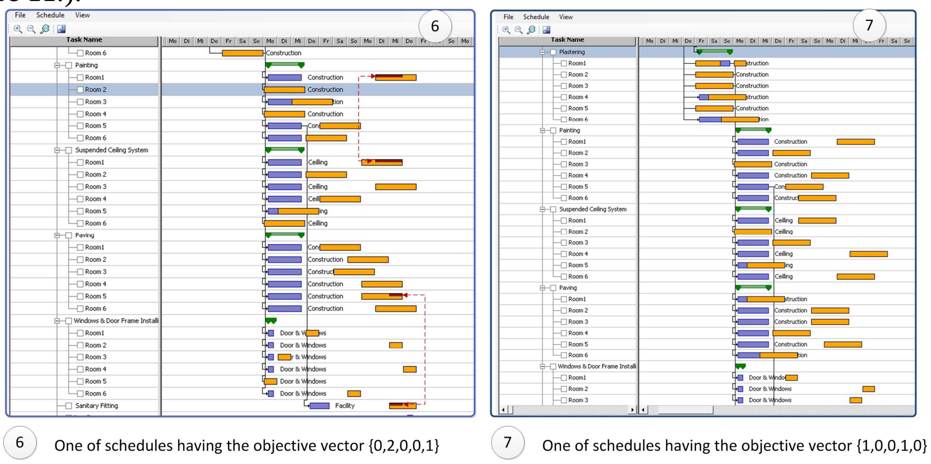

Figure 11: *Solutions of an optimization with 10 generations*

One of the matters, which also should be regarded here, is how better the SEAMO has been worked in the approach compared to a random searching algorithm. An experiment with a random searching is implemented. For the SEAMO, the experiment has been conducted with a population containing 50 individuals, 10 generations. For the random searching, the experiment has been conducted with a population with 12250 individuals; this number is equal to the maximum number of individuals which has been checked in the experiment with SEAMO.

The results have confirmed the efficiency of using SEAMO compared to a random searching (Figure 12:). With the SEAMO, the results have converged and the maximum delay duration is just two days in order to get other parameters' values is zero; otherwise, this number when using the random searching is three days. In addition, when the delay duration and labourer overrun are zero, schedules resulted from SEAMO have two days containing spatial congestions; however with a delay duration of zero, the minimum conflict duration in the random searching is five days. With just two objective vectors extracted from the results, it is enough to recognise how much more efficiently the optimization using SEAMO works compared to a random searching.

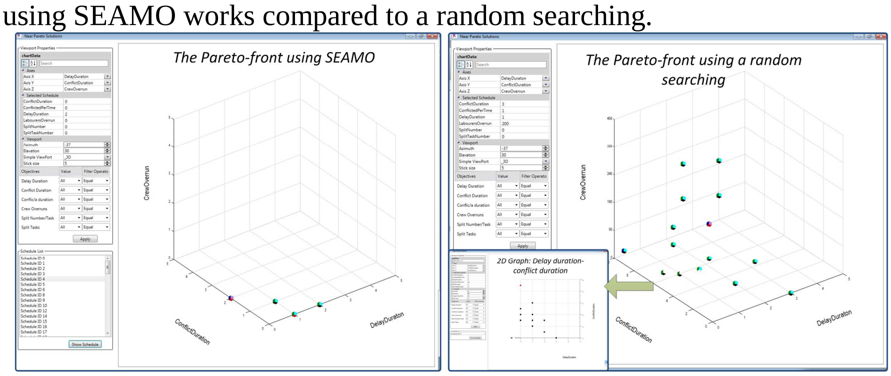

Figure 12: *Pareto-fronts with different searching methods*

8 CONCLUSION

The goal of the approach is to find a set of feasible strategies which resolve time-space conflicts and limits of crews. The integration of simulation with evolutionary algorithm has been successfully achieved. Like other results from random searching techniques, different schedules given through the proposed method in this research are really “diversified” and the searching process also converges quickly. Therefore, decision makers can choose the suitable solutions depending on their individual conditions such as crew size and material quality and quantity, etc.

Besides, project managers are able to evaluate solutions efficiently and make their decision based on the proposed methodology either for the whole schedule or for a selected part of the schedule (short term activities). However, the authors recommend the use of this methodology for the short term activities. Investigating whole a project is time consuming and still requires much detailed information. Moreover, a detailed schedule for whole a project loses the flexibility of planning and then not feasible in practice.

REFERENCES

[1] Mallasi, Z., and Dawood, N. (2001). "Assessing space criticality in sequencing and identifying execution patterns for construction activities using VR visualisations." *ARCOM doctoral research workshop: Simulation and modelling in construction*Edinburgh University, UK, 22-27.

[2] Hana, A. S., Russell, J. S., and Emerson, E. O. (2008). "Stacking of Trades." *Construction Productivity: A Practical Guide for Building and Electrical Contractors*, E. M. Rojas, ed., J Ross Publishing, 75-110.

[3] Bargstädt, H.-J., and Elmahdi, A. (2010). "Simulation von Bauprozessen - ein Qualitätssprung in der Arbeitsvorbereitung." *8. Grazer Baubetriebs- und Bauwirtschaftssymposium* Graz Tecnhnische Universität Graz, Germany, 131-146.

[4] Bargstädt, H.-J., and Elmahdi, A. (2010). "Automatic Generation of workspace Requirements Using Qualitative and Quantitative Description." *10th International Conference on Construction Applications of Virtual Reality*, Sendai CONVR2010 Organizing Committee, Japan, Japan, 131-137.

[5] Elmahdi, A., Wu, I.-C., and Bargstädt, H.-J. (2011). "4D Grid-based simulation framework for facilitating workspace management." *11th International Conference on Construction Applications of Virtual Reality*, Bauhaus Universität Weimar, Weimar Germany, 403-412.

[6] Riley, D. R., and Sanvido, V. E. (1997). "Space Planning Method for Multistory Building Construction." *Journal of Construction Engineering and Management*, 123(2), 171-180.

[7] Riley, D. R., and Sanvido, V. E. (1995). "Patterns of Construction-Space Use in Multistory Buildings." *Journal of Construction Engineering and Management*, 121(4), 464-473.

[8] Akinci, B., Fischen, M., Levitt, R., and Carlson, R. (2002). "Formalization and Automation of Time-Space Conflict Analysis." *Journal of Computing in Civil Engineering*, 16(2), 124-134.

[9] Akinci, B., Fischer, M., and Kunz, J. (2002). "Automated Generation of Work Spaces Required by Construction Activities." *Journal of Construction Engineering and Management*, 128(4), 306-315.

[10] Akinci, B., Fischer, M., Kunz, J., and Levitt, R. (2002). "Representing Work Spaces Generically in Construction Method Models." *Journal of Construction Engineering and Management*, 128(4), 296-305.

[11] Mallasi, Z. (2006). "Dynamic Quantification and Analysis of the Construction Workspace Congestion Utilising 4D Visualisation." *Automation in Construction*, 15(5), 640-655.

[12] Dawood, N., and Mallasi, Z. (2006). "Construction workspace planning : Assignment and analysis utilizing 4D visualization technologies." *Anglais*, 21(7), 498-513.

[13] Winch, G. M., and North, S. (2006). "Critical Space Analysis." *Journal of Construction Engineering and Management*, 132(5), 473-481.

[14] Zhang, C., Hammad, A., Zayed, T. M., Wainer, G., and Pang, H. (2007). "Cell-based representation and analysis of spatial resources in construction simulation." *Automation in Construction*, 16(4), 436-448.

[15] Bansal, V. K. (2011). "Use of GIS and Topology in the Identification and Resolution of Space Conflicts." *Journal of Computing in Civil Engineering*, 25(2), 159-171.

[16] Jongeling, R. (2006). "A Process Model for Work-flow management in construction: Combined Use of Location-based scheduling and 4D CAD."Doctoral Dissertation, Luleå University of Technology, Luleå University of Technology.

[17] Tulke, J. (2010). "Kollaborative Terminplanung auf Basis von Bauwerksinformationsmodellen."Doctoral Dissertation, Bauhaus Universität Weimar.

[18] Tauscher, E. (2011). "Vom Bauwerksinformationsmodell zur Terminplanung: Ein Modell zur Generierung von Bauablaufplänen."Dotoral Dissertation, Bauhaus Universität Weimar.

[19] Dang, T., Elmahdi, A., and Bargstädt, H. J. "Generating Workspace Requirements in a finishing execution phase." *Proc., 12th International Conference on Construction Application of Virtual Reality*, 521-531.

[20] Mumford-Valenzuela, C. L. (2005). "A simple approach to evolutionary multiobjective optimization." *Evolutionary Multiobjective Optimization - Theoretical Advances and Applications*, L. J. Ajith Abraham, Robert Goldberg, ed., Springer, United States of America, 55-104.

[21] Mumford, C. (2010). "A multiobjective framework for heavily constrained examination timetabling problems." *Annals of Operations Research*, 180(1), 3-31.