CIFE CENTER FOR INTEGRATED FACILITY ENGINEERING

Decision Making for Schedule Optimization

By

Fabian Märki, Martin Fischer, John Kunz & John Haymaker

CIFE Technical Report #169
MARCH 2007

STANFORD UNIVERSITY

If you would like to contact the authors, please write to:

c/o CIFE, Civil and Environmental Engineering Dept.,
Stanford University
Terman Engineering Center
Mail Code: 4020
Stanford, CA 94305-4020

Decision Making

for

Schedule Optimization

Fabian Märki[1], Martin Fischer[1], John Kunz[1], John Haymaker[1]

[1] Department of Civil and Environmental Engineering, Center for Integrated Facility Engineering, Stanford University, Stanford CA 94305-4020, USA
markif@stanford.edu
http://www.i4ds.ch
fischer@stanford.edu
kunz@stanford.edu
haymaker@stanford.edu
http://www.stanford.edu/group/CIFE/

Abstract

This paper presents a novel formulation of scheduling and decision information which allows a concurrent optimization of both, the decisions leading to a specific schedule and the schedule itself. Our methodology allows a dynamic adaptation of the optimization criteria according to the quality measurement criteria of the involved decision making stakeholders. Major types of possible quality measurement criteria are project duration considerations, cost considerations, resource levelling considerations, safety considerations and some miscellaneous considerations like distances resources have to cover from one assignment to the next or time space conflicts of resources.

Decisions and their alternatives are represented in a Decision Breakdown Structure (DBS) (Kam, 2006). The DBS defines the search space for the optimization algorithm which is based on a Genetic Algorithm (GA) approach. The optimization algorithm uses the novel formulation of scheduling and decision information to find a Pareto optimal decision alternative combination which leads to a Pareto optimal schedule.

First tests of the decision and schedule optimization algorithm show that optimizations can be performed within one minute. This short latency suggests that the proposed concepts about decision optimization could, for instance, be utilized in meetings or in an Integrated Concurrent Engineering (ICE) environment where short latency is extremely important (Chachere, 2004) because stakeholders need to get a quick idea about good decisions and their predicted outcome.

Keywords: Automated Project Planning, Automated Decision Making, Integrated Concurrent Engineering, Resource Modelling, Optimization, Genetic Algorithm

Introduction

Have you ever asked yourself how different combinations of decision alternatives like decisions about the methodology to build a specific component or specific resource assignments could affect the outcome of a project and which of these combinations meet the interests of the involved stakeholders in the best possible way? How long are you usually waiting for an answer to these questions?

An in depth sensitivity analysis can provide an answer to the above questions. Unfortunately, such an analysis is usually time consuming and if decision alternatives or quality measurement criteria change the analysis has to be redone.

This work addresses the above questions and problems and describes a concept that enables decision makers to improve decision alternative combinations with very short latency while preserving the ability to dynamically change the underlying decision quality measurement criteria.

Calvin Kam's dissertation (Kam 2005) provides the point of departure for the modeling of decision alternatives. Kam develops the concept of a Decision Breakdown Structure (DBS) which describes all decisions and the alternatives decision makers consider for a specific project. The DBS also ensures that each involved stakeholder is aware of interrelationships between decisions, so that important interrelated considerations will not be forgotten. Because the DBS centralizes decisions in one model, stakeholders can focus on this model which drastically simplifies the human interrelationship network (Figure 1).

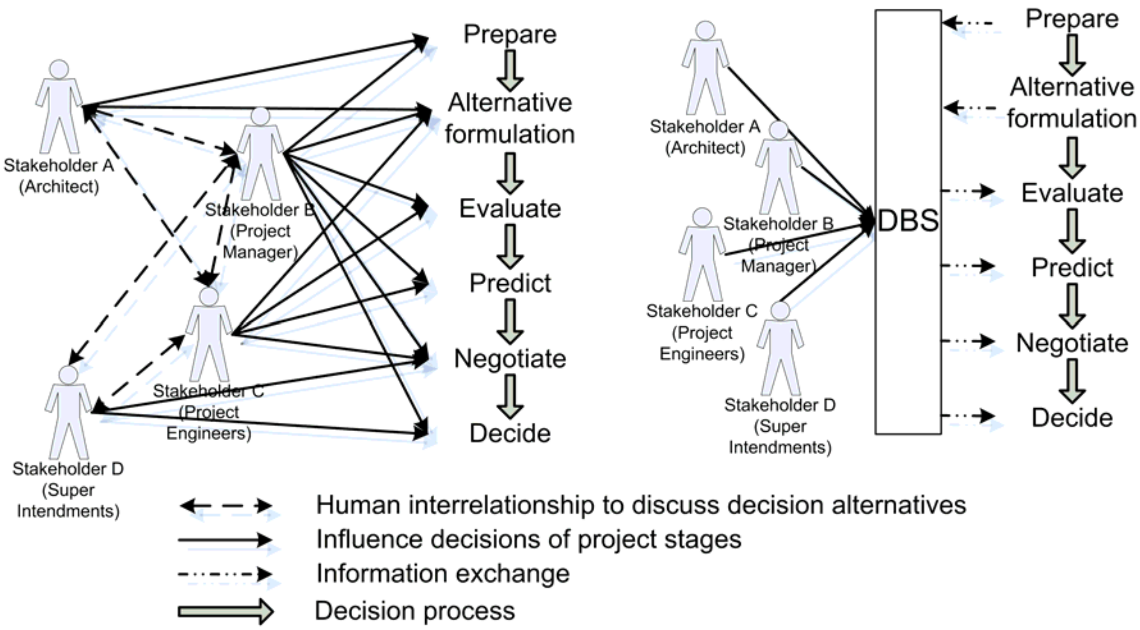

Figure 1: Decision making interrelationship with and without DBS. The standard procedure consists of a complex network which makes it difficult to keep track of decisions and their alternatives. In contrast, the DBS centralizes decision alternatives and makes them available for every involved stakeholder. Based on this collection of decision alternatives, stakeholders can make sound decisions which consider the interests of all involved stakeholders.

The evaluation of a combination of decision alternatives is based on a Genetic Algorithm (GA) approach. GAs are a class of heuristic search methods based on the Darwinian principle of evolution[1]. It mimics and exploits the genetic dynamics underlying natural evolution to search for Pareto[2] optimal solutions of general combinatorial optimization problems (Coley 1999). The needs and goals of such an evaluation are illustrated in Figure 2.

[1] Wikipedia: http://en.wikipedia.org/wiki/Genetic_algorithm (Last accessed: February 07th, 2007)

[2] Given a set of alternative allocations and a set of individuals, a movement from one allocation to another that can make at least one individual better off, without making any other individual worse off, is called a Pareto improvement or Pareto optimization. An allocation of resources is Pareto efficient or Pareto optimal when no further Pareto improvements can be made. Wikipedia: http://en.wikipedia.org/wiki/Pareto_optimal (Last accessed: February 07th, 2007)

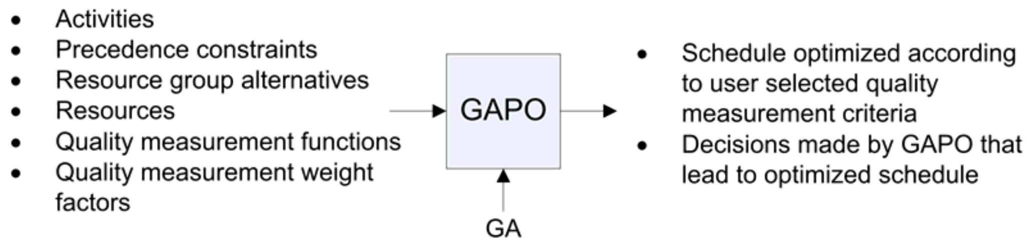

Figure 2: GAPO's input is a list of activities, a definition of precedence constraints between these activities, alternatives of resource groups that can be assigned to the activities, the resources used by the resource groups, functions which measure the quality of GAPO's decisions based on the quality measurement criteria of the involved stakeholders and weight factors which reflect how the stakeholders prioritize the quality measurement criteria (see chapter on Decision and Schedule Quality for further explanations). Based on all the resulting decision alternative combinations, GAPO searches for the decision alternative combination which leads to a schedule that meets the quality measurement criteria of the involved stakeholders in a Pareto optimal way.

Prior research in schedule optimization done at i4Ds[3] (Märki and Suter, 2003) and more recent research done at CIFE[4] (Märki et al. 2006) provide the point of departure for the decision optimization aspect of this research. The Genetic Algorithm Process Optimization (GAPO) framework has been extended to accommodate an analysis of all the different combinations of decision alternatives concurrently. Consequently, this leads to a Pareto optimal decision about which combination of alternatives conforms best to the defined quality measurement criteria of the involved stakeholders (Figure 3).

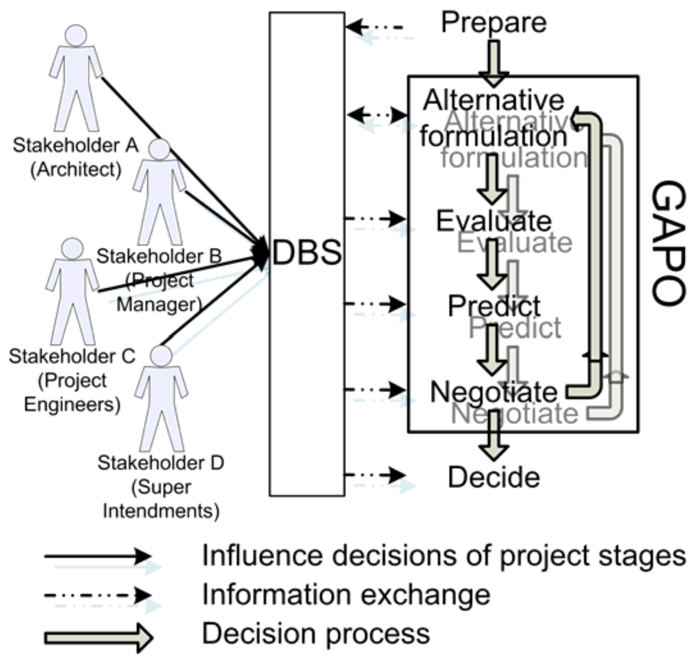

Figure 3: GAPO further simplifies the decision making process (compare with Figure 1). GAPO uses the DBS as the search space and concurrently performs an alternative formulation, evaluation, prediction, and negotiation of the different combinations of all decision alternatives of the DBS and finally decides upon a Pareto optimal decision alternative combination based on decision quality measurement criteria which were predefined by the involved stakeholders.

This research focuses on decision alternatives about resource quantities and resource group assignments to activities:

[3] Institute 4D–Technologies & DataSpaces, University of Applied Sciences Northwestern Switzerland
[4] Center for Integrated Facility Engineering, Stanford University

- A change in the quantities of the resources used within a resource group leads to a change in the duration of an activity.
- A change in the resource group assigned to an activity can affect the process method used to perform an activity. An example of such a change is the usage of a resource group consisting of builders to erect a wall using bricks. The same wall could also be built with alternative materials which would allow a prefabrication of the wall. In order to erect the same but prefabricated wall, a resource group consisting of a crane operator and a crane to lift it into place would be appropriate.

GAPO's particular focus is to find the decision alternative combination for the resource quantity decision and the resource group assignment decision which will, combined with the concurrently optimized schedule, lead to a Pareto optimal project schedule. Thereby, users can adapt the decision quality measurement criteria dynamically according to their needs.

Figure 4: Changes in used resource quantities and resource group are the decisions which are considered in this research.

Consider a simple example that illustrates the possibilities GAPO provides. A project manager is compiling a resource leveled project schedule. The decisions he has to make are how he wants to schedule the activities, which resource group he should assign to the activities and how much of each resource of the resource groups he should use. The decision alternative combinations resulting from such a problem can be enormous (Table 1). Therefore, the project manager does not have the time to search for the decision alternative combination which leads to a good resource leveled schedule. Consequently, the project will most likely be preformed with a suboptimal schedule.

The above problem is a typical case where the project manger could utilize GAPO. The input data he would have to provide are illustrated in Figure 2. In our example, the project manager would have to provide a quality measurement function which favors schedules with good resource leveling. He could also define other quality measurement functions (for instance a function that measures project duration) but he would have to prioritize the resource leveling function most since it is his major concern.

When we run GAPO with a sample project (see chapter on Motivating case example), GAPO needs about 10 seconds to improve the resource leveling of this project from its initial schedule to its final Pareto optimal schedule as it is illustrated in Figure 5.

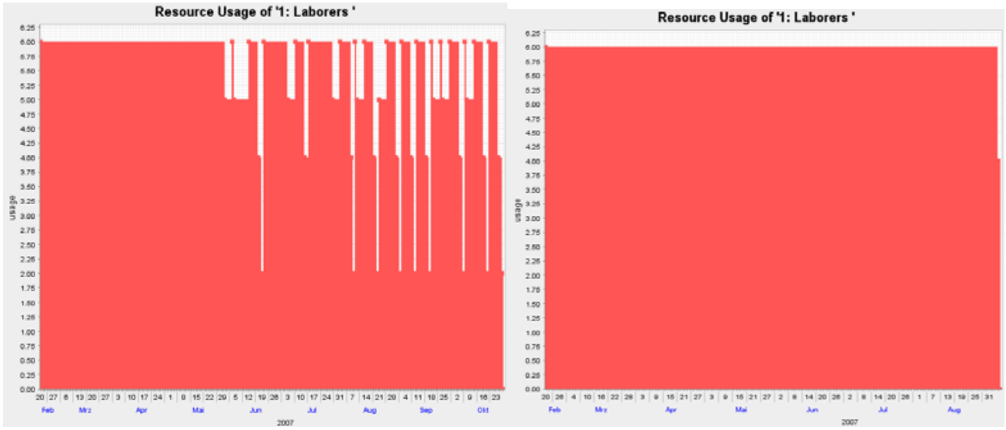

Figure 5: GAPO improves the resource leveling of the resource *Laborers* within 10 seconds from the initial schedule (as shown on the left graph) to its final Pareto optimal schedule (as shown on the right graph). This suggests that the algorithm has a short latency to find the Pareto optimal decision alternative combinations.

Motivating case example

The case example is based on a drywall installation case of an office building project (Staub-French et al. 2002)[5].

The drywall (Table 1) can be installed by using a resource group consisting of *laborers* and *rolling scaffolding* or by a resource group consisting of *laborers* and a *scissor lift*. The estimator has a preference to use the same type of equipment for all the drywall activities. We made the following further assumptions:

- There are a total of 10 rooms
- There are three rolling scaffoldings available and the usage of rolling scaffolding is cheaper than the usage of a scissor lift.
- There are three scissor lifts available and the usage of a scissor lift is more expansive than the usage of rolling scaffolding. However, through the usage of scissor lifts, the activities can be performed faster.

Figure 6 illustrates the Organization Breakdown Structure (OBS) and the Work Breakdown Structure for this scope of work. Table 1 summarizes all possible decision alternatives. GAPO's task is to find the best combination of these alternatives based on different decision quality measurement criteria defined by the involved stakeholders.

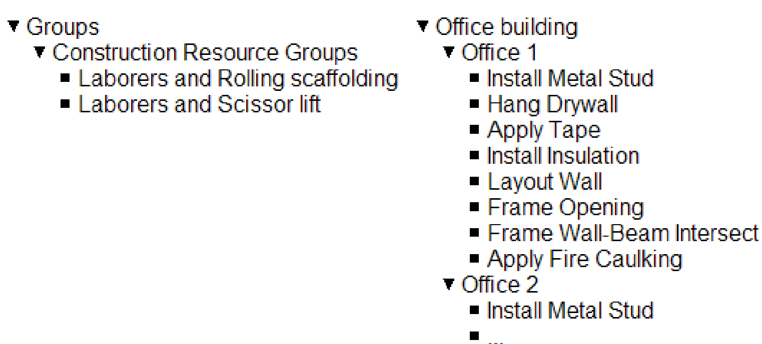

Figure 6: Organization Breakdown Structure (OBS) and Work Breakdown Structure (WBS) for the scope of the motivating case example. Each activity in the WBS can be performed by *laborers and rolling scaffolding* or *laborers and scissor lift*. The estimator prefers to use the same type of equipment for all activities.

[5] Further details about this project can be obtained from http://cife.stanford.edu/online.publications/WP071.pdf

The calculation of the duration of the activities is based on the quantity of materials to install and on the resource group and resource quantities assigned to the activity (see chapter on Considered Decision for further explanations).

Table 1 illustrates that this case example has approximately 10^{62} possible decision alternative combinations. This is a enormous search space and a human project manager will never have the time to explore the whole search space to find an optimal schedule. More likely, he will search for the first schedule he believes to be feasible and the project will be performed according to this schedule.

Activities	Assigned ResourceGroup	Possible Durations	# Decision Alternatives
Office 1			
Install Metal Stud	Laborers and rolling scaffolding	12,14,18,24,35	10
	Laborers and scissor lift	10,12,14,19,28	
Hang Drywall	Laborers and rolling scaffolding	4,5,7,10	9
	Laborers and scissor lift	3,4,5,6,8	
Apply Tape	Laborers and rolling scaffolding	5,6,7,10,14	10
	Laborers and scissor lift	4,5,6,8,11	
Install Insulation	Laborers and rolling scaffolding	1,2,3	6
	Laborers and scissor lift	1,2,3	
Layout Wall	Laborers and rolling scaffolding	1,2	4
	Laborers and scissor lift	1,2	
Frame Opening	Laborers and rolling scaffolding	1	2
	Laborers and scissor lift	1	
Frame Wall-Beam Intersect	Laborers and rolling scaffolding	1	2
	Laborers and scissor lift	1	
Apply Fire Caulking	Laborers and rolling scaffolding	1	2
	Laborers and scissor lift	1	
Office 2			
Install Metal Stud	Laborers and rolling scaffolding	12,14,18,24,35	10
	Laborers and scissor lift	10,12,14,19,28	
…	…	…	…
Total decision alternative combinations			**approx. 10^62**

Table 1: From the Decision Breakdown Structure (DBS – see Figure 10), we can derive that each activity can be performed by *laborers* using *rolling scaffolding* or a *scissor lift*. The duration of an activity changes depending on the assigned resource group and the amount of assigned resources. This results in a specific number of possible decision alternatives for each activity. The total number of possible decision alternative combinations can be derived from the formula $\prod_{i=1}^{nActivities} DA_i$ (the product of the number of decision alternatives of each activity over all activities). Consequently, the case example has approximately 10^{62} possible decision alternative combinations.

Optimization of schedules

Compiling schedules is a time consuming and complex task. Due to time constraints and the NP-hardness of the scheduling optimization problem[6], one does usually not implement a sensitivity analysis and projects are performed with the first found feasible schedule. With Genetic Algorithm Process Optimization (GAPO[7]), we want to address this problem and show

[6] Wikipedia: http://en.wikipedia.org/wiki/Scheduling (Last accessed: 19th of February 2006)

[7] GAPO was initially developed at i4Ds (Märki and Suter 2003). It is programmed in Java™ and consists of about 270 classes.

the potential of a software application that supports the compilation of a sound Pareto optimal schedule.

Genetic Algorithm Process Optimization (GAPO)

GAPO is based on a Genetic Algorithm (GA) approach to optimize schedules in terms of time, cost and resource management (Märki and Suter 2003). GAs are a class of heuristic search methods based on the Darwinian principle of evolution[8]. It mimics and exploits the genetic dynamics underlying natural evolution to search for Pareto optimal solutions of general combinatorial optimization problems (Coley 1999).

GAs start from a pool of individuals which are scored by fitness functions[9] measuring their quality as a candidate solution of a given problem. By some probabilistic mechanism, these solutions are exposed to an artificial evolution consisting of selection, recombination and mutation yielding a new generation of candidate solutions which are expected to have a higher quality of fitness. GAs have been successfully applied to a wide variety of practical problems in diverse fields like chemistry, biology, operations research, and many engineering disciplines (Koza et al. 2003, Coley 1999, KHosraviani et al. 2004).

GAPO Evolution Model

The evolution model starts with an initial population of randomly generated schedule individuals. A subsequent population will then be assembled using five strategies which can be weighed by the user.

A fraction *q* of the best individuals will be directly passed to the next population. This guarantees that the quality of the most suited candidates will monotonically increase from generation to generation. A second fraction *r* of individuals will be passed to the next population after a mutation. On one side, this process opposes early convergence in a local optimum and thereby helps to open new search regions. On the other side, it also allows a fine tuning of suitable solutions by applying small changes on them. A third fraction *s* of the new population is created by recombining individuals from the old generation. This process forces convergence into an optimum. A fourth fraction *t* is also created by recombining individuals but instead of passing them directly into the new population the new individual is mutated beforehand. Last, a fraction *u* of the new population is created randomly. This process also helps to open new search regions and prevents early convergence in a local optimum (Gonçalves 2002).

[8] Wikipedia: http://en.wikipedia.org/wiki/Genetic_algorithm (Last accessed: February 07th, 2007)
[9] A fitness function is a particular type of objective function that quantifies the optimality of a candidate solution so that that particular solution can be ranked against all the other solutions. Wikipedia: http://en.wikipedia.org/wiki/Fitness_function (Last accessed: February 11th, 2007)

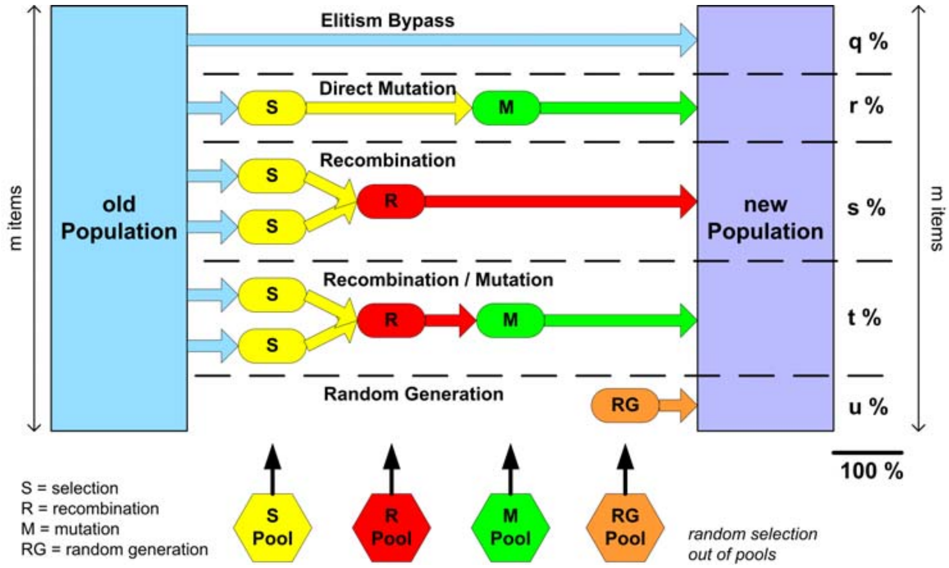

Figure 7: GAPO Evolution Model (Märki et al. 2006). It describes how a new population is formed based on an old population. This process uses five different strategies which are Elitism Bypass, Direct Mutation, Recombination, Recombination combined with a Mutation and Random Generation. Prior to each run of a strategy, it randomly chooses the appropriate operators (selectors, recombinator, mutator and random generator) from their pools. The number of new individuals generated by each strategy can be weighed.

GAPO Data Structure

The success of GAs crucially depends on an appropriate encoding of all the different parameters. Additionally, schedule optimization has to consider activity sequencing constraints which have to be satisfied. Based on prior experience we decided to use a genotype data structure that is kept as simple as possible. By genotype we refer to the genetic encoding of a schedule as it is represented in the computer's memory. The genotype representation allows the application of operators like mutators and recombinators. The counterpart of the genotype is the phenotype. In our optimization problem the phenotype is the actual schedule represented in CPM format.

The genotype representation consists of four arrays whose length is equal to the number of activities contained in the schedule. The first two arrays are used for the schedule encoding and the second two arrays are used to represent chosen decision alternatives (see Figure 8 and the chapter on Project Planning Decision Making for more details about the genetic encoding of decisions).

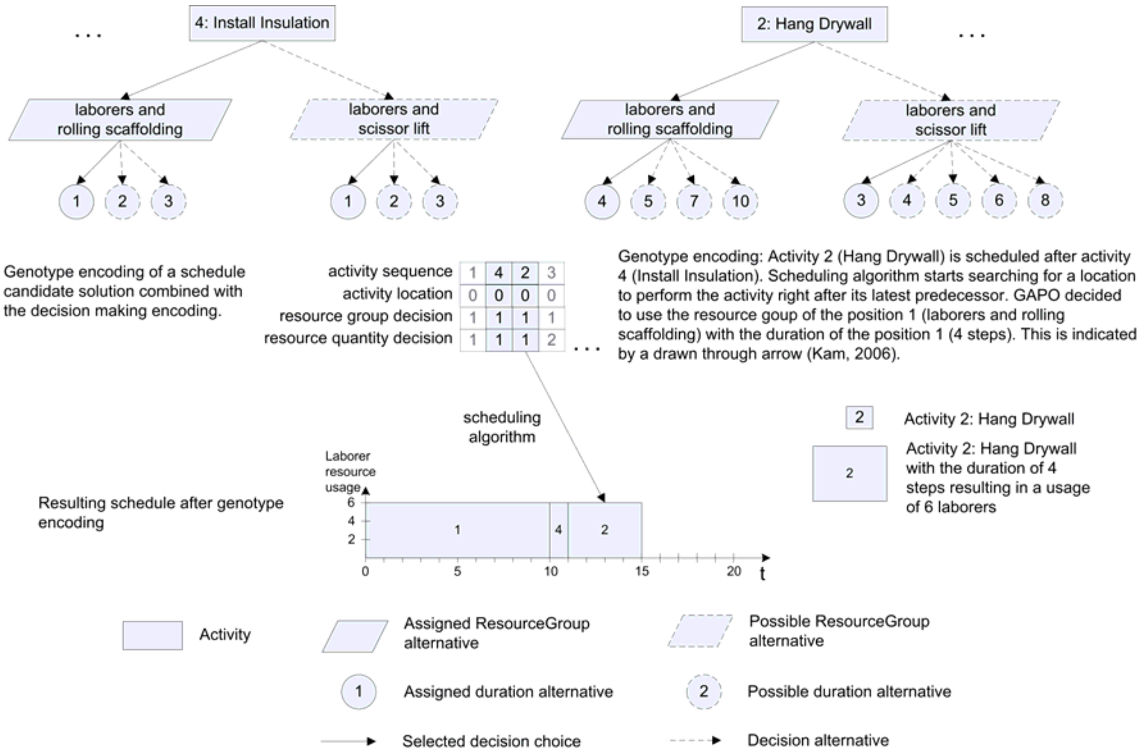

Figure 8: Probabilistic mechanisms create candidate solutions in the form of a genotype as shown in this figure. The artificial evolution process applies mutation and recombination operations on the genotype encoding which is then used to form the actual schedule. This figure illustrates how the schedule optimization genotype (first two arrays) is combined with the decision genotype encoding for the decision about the resource group assignment (third array) and the decision about the activity duration (fourth array). Through this combination, GAPO is able to concurrently expose the schedule and the decisions to an optimization process.

The first array of the genotype encoding represents the sequence of how the scheduling algorithm should schedule the activities. The second array defines the position where the scheduling algorithm starts to search for a location where the activity can be performed without violating a resource constraint. By definition, this position is between the end of the latest predecessor of the handled activity and the end of the latest scheduled activity. If it is not possible to schedule the activity within this range, it will be added to the end of the schedule as the last activity.

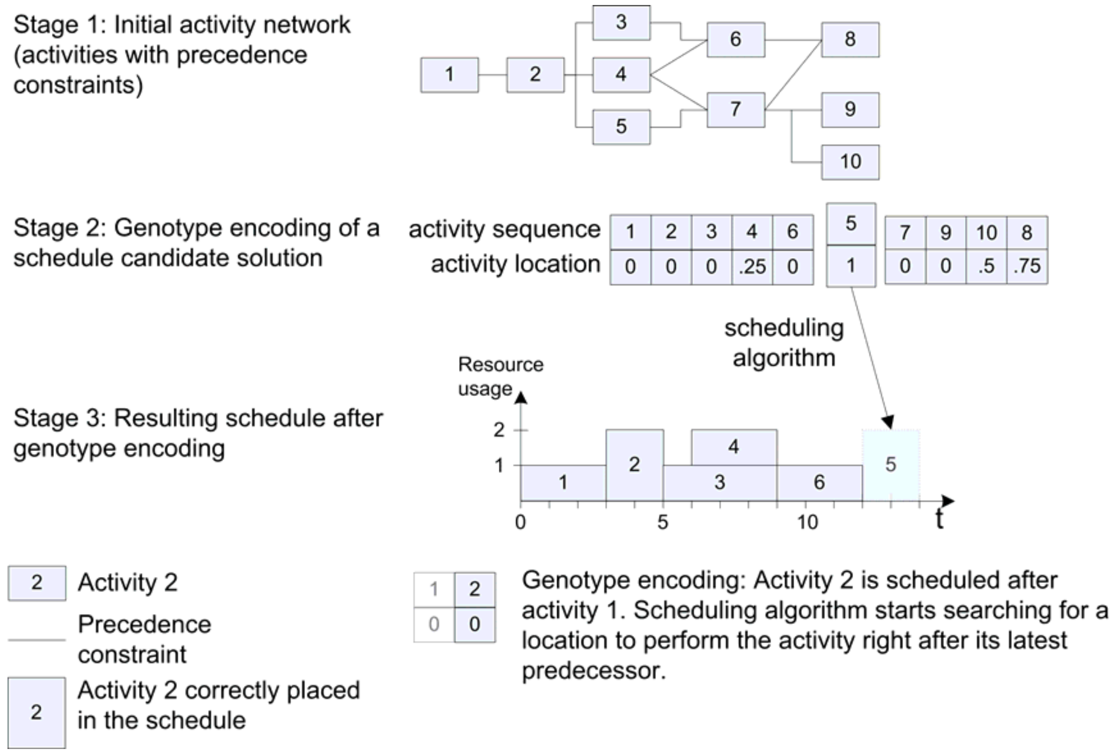

Figure 9: This figure focuses on the first two arrays of the genotype encoding. These two arrays encode how the activities are scheduled. The basis for this encoding builds an activity network (activities with precedence constraints). The first array encodes the sequence of how the scheduling algorithm should schedule the activities. The second array defines the position where the scheduling algorithm starts to search for a location where the activity can be performed without violating a resource constraint.

Project Planning Decision Making

This chapter describes how we combine schedule optimization with project planning decision making.

Kam's dissertation (Kam, 2005) shows how decision alternatives can be represented and related to each other. Kam develops a dynamic Decision Breakdown Structure (DBS) which describes decisions project managers consider when they decide about how to perform a specific project (Figure 10). At the same time, the DBS also incorporates alternatives for each decision. Kam emphasizes that there exist interrelationships between pairs of alternatives of different decisions. He refers to these interrelationships as *impact relationship* and *requirement relationship*. If one wants to come up with a sound decision, he has to consider these interrelationships because they might exclude alternatives of other decisions.

This explicit description of decisions and their alternatives gives project managers a sound information basis to make the actual decision. Because crucial information is at hand, it becomes unlikely that decisions are invalid because important aspects were forgotten. However, the decision must still be made by human beings and due to the tremendous number of decision alternative combinations and the complexity of the interrelationships it is very unlikely that decisions will be made which lead to a Pareto optimal project schedule.

It is possible to come up with the decision alternative combinations which result in a Pareto optimal project schedule by concurrently exposing scheduling and decision making to an optimization process. The quality of the decision alternatives is measured by schedule quality measurement criteria. This gives an insight about the quality of a decision alternative combination in terms of project planning criteria.

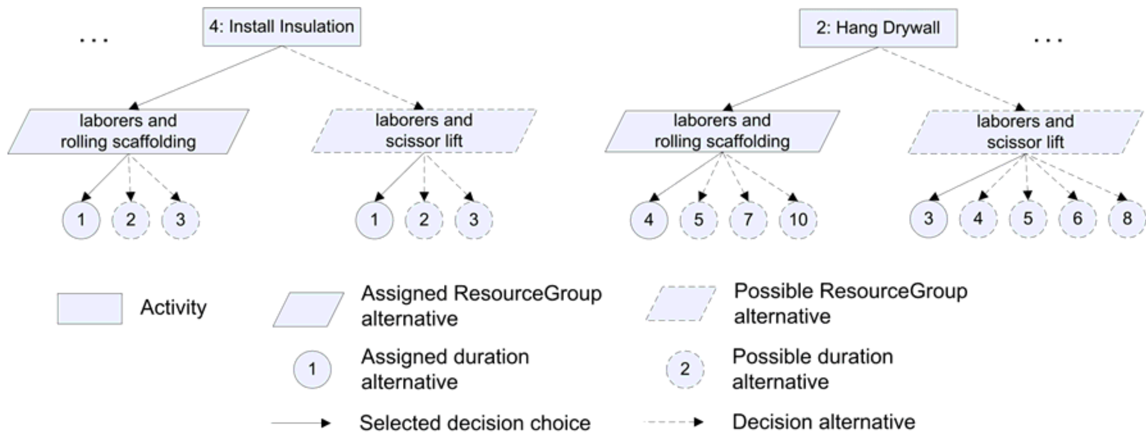

Figure 10: A DBS for all the possible decision alternatives of the activities *Install Insulation* and *Hang Drywall*. *Hang Drywall* can be performed by using the resource group *laborers and rolling scaffolding* or *laborers and scissor lift*. In this example, the decision was made to use *laborers and rolling scaffolding*. Within the resource group *laborers and rolling scaffolding* it is possible to assign resource quantities which will result in activity durations of 4, 5, 7, or 10 days. In this example, the decision was made to assign resource quantities which result in an activity duration of 4 days as shown by the decision choice arrow link (Kam, 2006).

Incorporation of DBS into GAPO

The information contained in the DBS describes the search space for all possible decision alternative combinations. Thereby, the interrelationships between decision alternatives are of importance because they make it impossible to combine some alternatives with some other alternatives. A concrete decision alternative combination must not violate any of these interrelationships because otherwise it will not be sound.

Decision alternatives and their interrelationships can be transformed into a *Constraint Satisfaction Problem* (CSP). This has the advantage that standard CSP solving algorithms (Prosser, 1993; Bacchus and Run, 1995) can be used to generate concrete and sound decision alternative combinations.

Genetic Encoding of Decisions

A decision is encoded as an array of integer values (Figure 11). A position of this array represents a specific decision and the integer value assigned to this position defines which alternative of this decision is used.

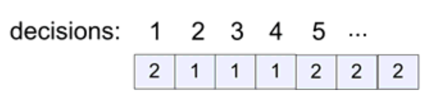

Figure 11: Decision encoding into an integer array. The array position determines the specific decision and the integer value of the position specifies the used alternative.

After a decision encoding undergoes a mutation or recombination operation, the decision alternatives might violate interdependency constraints and consequently the decision might not be sound anymore (Figure 12). Therefore, a post-processing step is necessary which checks decision arrays for violations and, if necessary, corrects them. These corrections must be as minimal as possible because otherwise a recombination operation will be blurred into a mutation operation.

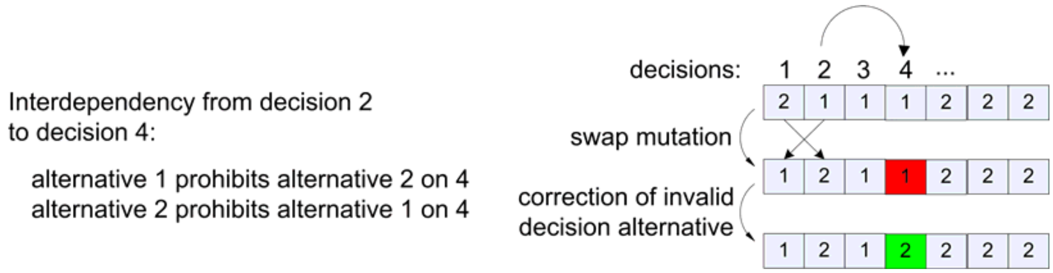

Figure 12: Correction of interdependency violation between decision 2 and 4 provoked by a swap mutation operation.

Accordingly, we adapted the standard CSP solving algorithm to provide an initialization with any valid or invalid solution. Thereafter, the adapted CSP solver starts a local search for a new valid solution.

Considered Decisions

This research focuses on decision alternatives about resource quantities and resource group assignments to activities and the combinations of these decision alternatives that lead to a Pareto optimal project schedule (see Figure 10 for a DBS example for these decisions).

Resource Group

An activity might need more than one resource in order to be performed. The resources necessary to perform an activity are combined in a resource group. A change in the resource group assigned to an activity can affect the process method used to perform an activity. An example of such a change is the usage of a resource group consisting of builders to erect a wall using bricks. The same wall could also be built with alternative materials which would allow a prefabrication of the wall. In order to erect the same but prefabricated wall, a resource group consisting of a crane operator and a crane to lift it into place would be appropriate.

In the case example it is possible to perform the activity *Hang Drywall* by using *laborers* with *rolling scaffolding* or *laborers* with a *scissor lift*. Accordingly, the activity *Hang Drywall* has two resource group alternatives and the construction planner will have to decide which one to use (Figure 13).

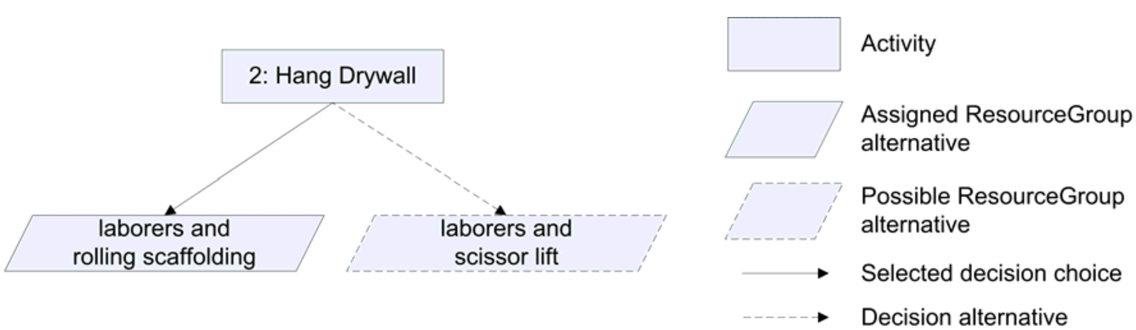

Figure 13: The decision alternatives between the resource groups *laborers and rolling scaffolding* and *laborers and scissor lift* which can be used to perform the activity *Hang Drywall*.

The cost estimator of the case example stated a preference to use the same type of equipment for all the activities. This is a typical interrelationship between decision alternatives. It has the implication that if one decides to use the resource group *laborers and rolling scaffolding* for the activity *Hang Drywall* one also has to use the same resource group for the activity *Install Insulation* and vice versa (Figure 14).

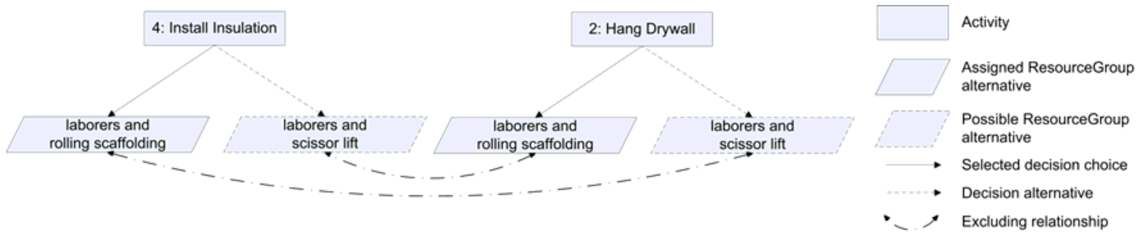

Figure 14: Interrelationships between decisions about resource group alternatives are modeled as exclusions. If *Hang Drywall* is performed by using *laborers and rolling scaffolding* the alternative *laborers and scissor lift* of the activity *Install Insulation* is automatically excluded and vice versa.

These decision alternatives and their interrelationships are encoded as described in the chapter Genetic Encoding of Decisions.

Resource Quantities

The resource quantity defines the amount of a resource used to perform a specific activity. This information is necessary to conclude about the activity duration. The relationships between resources are described in the resource group. We illustrate this on the case example.

To perform the activity *Hang Drywall* it is necessary that two *laborers* work together. This resource is the driving resource of the resource group *laborers and rolling scaffolding* and we shall call it the *master resource*. These two *laborers* need one *rolling scaffolding* to do their work properly. Consequently, they have a two to one relationship.

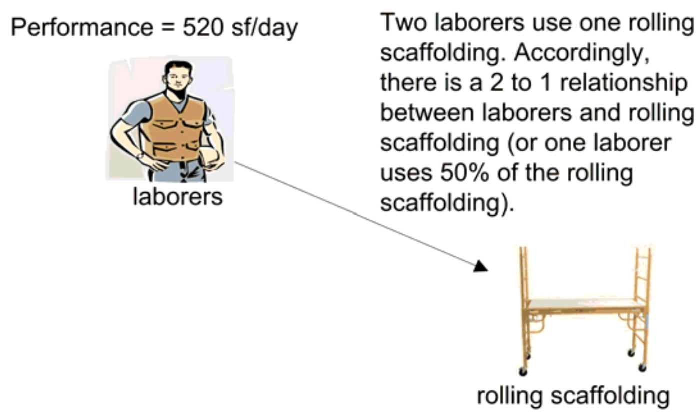

Figure 15: This figure illustrates the relationship between the *laborers* and the *rolling scaffolding*. There are two *laborers* necessary to perform the activity *Hang Drywall*. These two laborers together need one *rolling scaffolding* to do their work properly. Accordingly, one *laborer* occupies 50% of the *rolling scaffolding*.

From Staub-French et al. (2002) we know that this combination of resources can hang 520 sf of drywall per day. This results in a performance of 520 sf/day for this resource group. Since we know that there are 4,800 sf of drywall, we can conclude that the resource group needs 9.23 days (~10 days) to do the task.

From this information we can come up with three different equations (Figure 16) which describe the relationship between an activity and a resource group (A), between a resource group and a master resource (B), and between resources of a resource group (C).

- **Function A** defines the relationship between the resource group amount and the activity duration and vice versa. It defines the duration of an activity if one resource group entity is used. We decided to provide the possibility to specify an arbitrary function to describe the dependency between resource group and activity duration because an increase in resources does not necessarily lead to a proportional decrease in activity duration (Jenkins, 2006).

- **Function B** defines the relationship between the resource group amount and the *master resource* of the resource group. It defines how much of the master resource is used by one resource group entity. The master resource is the resource which is at the root of a resource tree. It is usually the resource which defines the performance of a resource group. There is only one master resource per resource group.
- **Function C** defines the relationship between parent and child resources. By definition, it defines how much of the child resource is used by one parent resource.

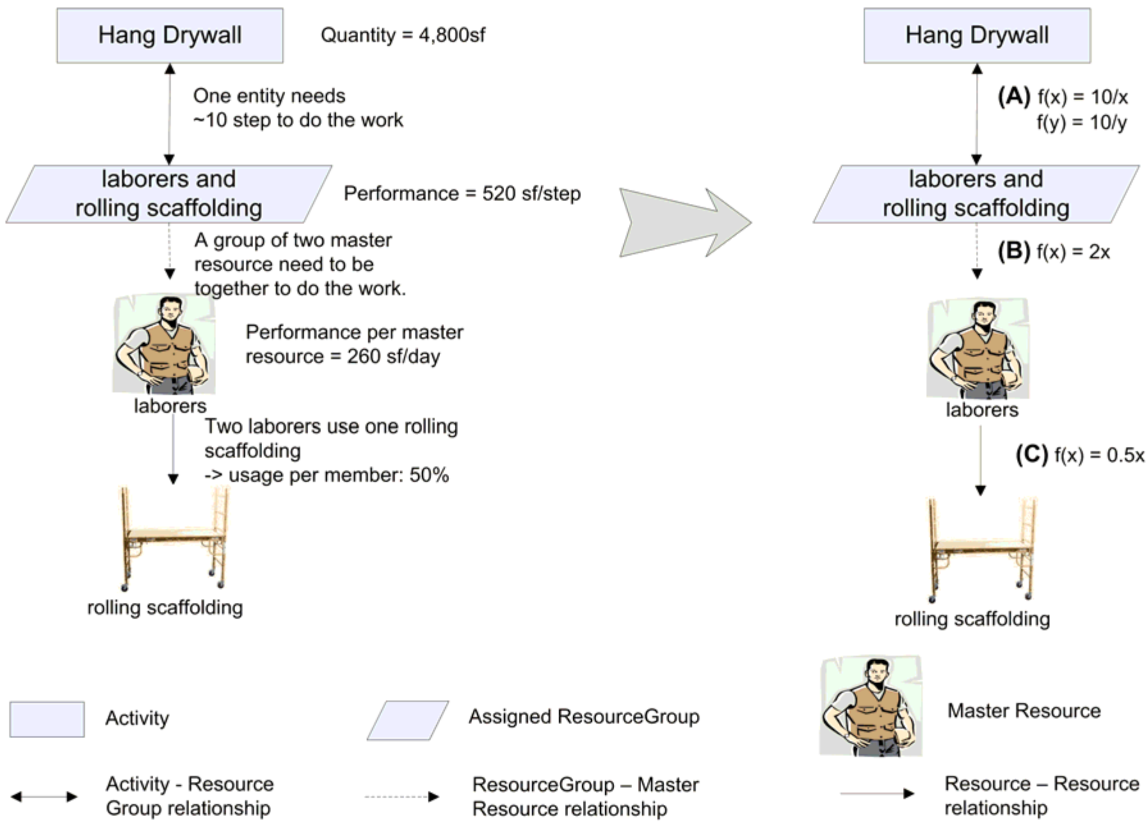

Figure 16: The left side of this figure illustrates how the relationships between activity, resource group, and resources are seen in the real world. The figure on the right shows how we convert these relationships into three distinct functions. Function A defines the relationship between the resource group amount and the activity duration (and vice versa). Function B describes the relationship between the resource group amount and the master resource of the resource group. Function C describes the relationship between the parent and child resources. All these functions are normalized for one entity.

The resource model as it is described above is very flexible and can model complex resource relationships which can, for instance, also contain material resources or resources which model space requirements (Figure 17).

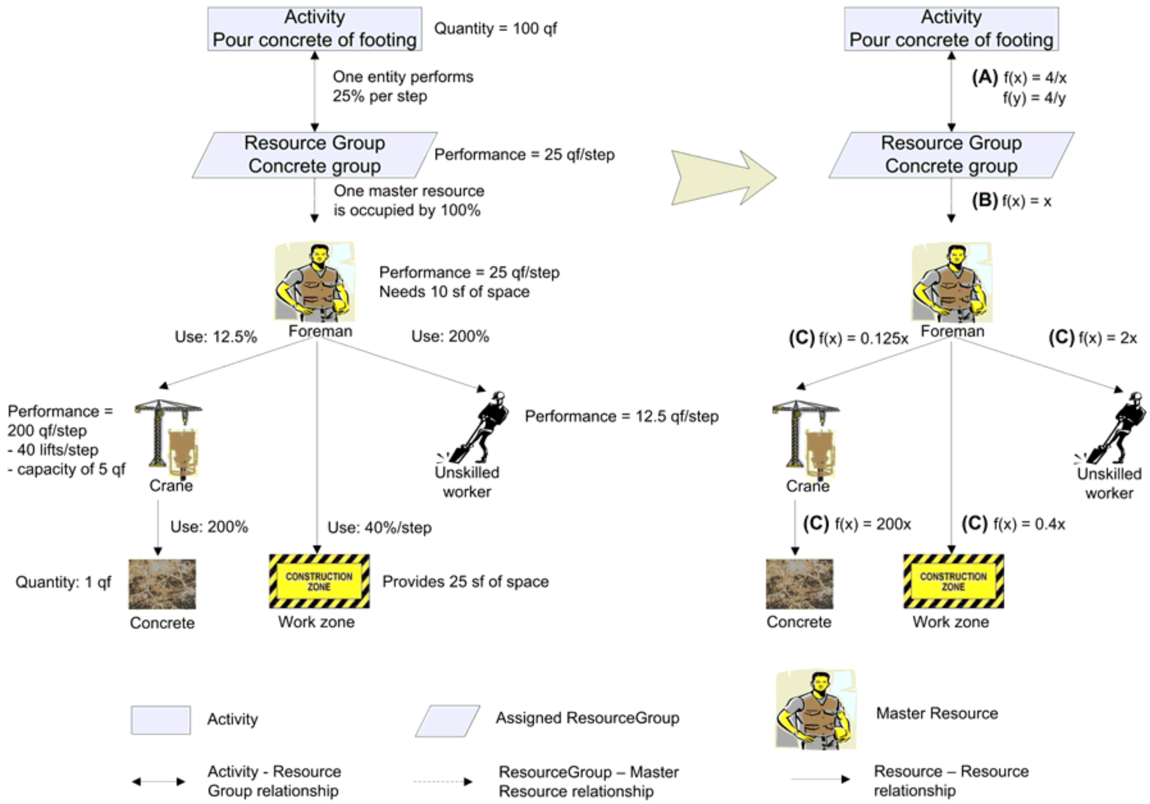

Figure 17: Resource model describing the relationship between a pour concrete activity and the resource group containing the necessary resources to perform this kind of activity.

Because we know the maximal available amount of each resource, we can calculate the minimal and maximal duration of each activity and its feasible durations in between. Consequently, we can choose between different duration decision alternatives for each activity.

Combining decision genotype with schedule genotype

The final genotype encoding used by GAPO contains the two arrays of the schedule optimization encoding (see chapter on GAPO Data Structure), one array for the decision about the assigned resource group and one array for the decision about the activity duration.

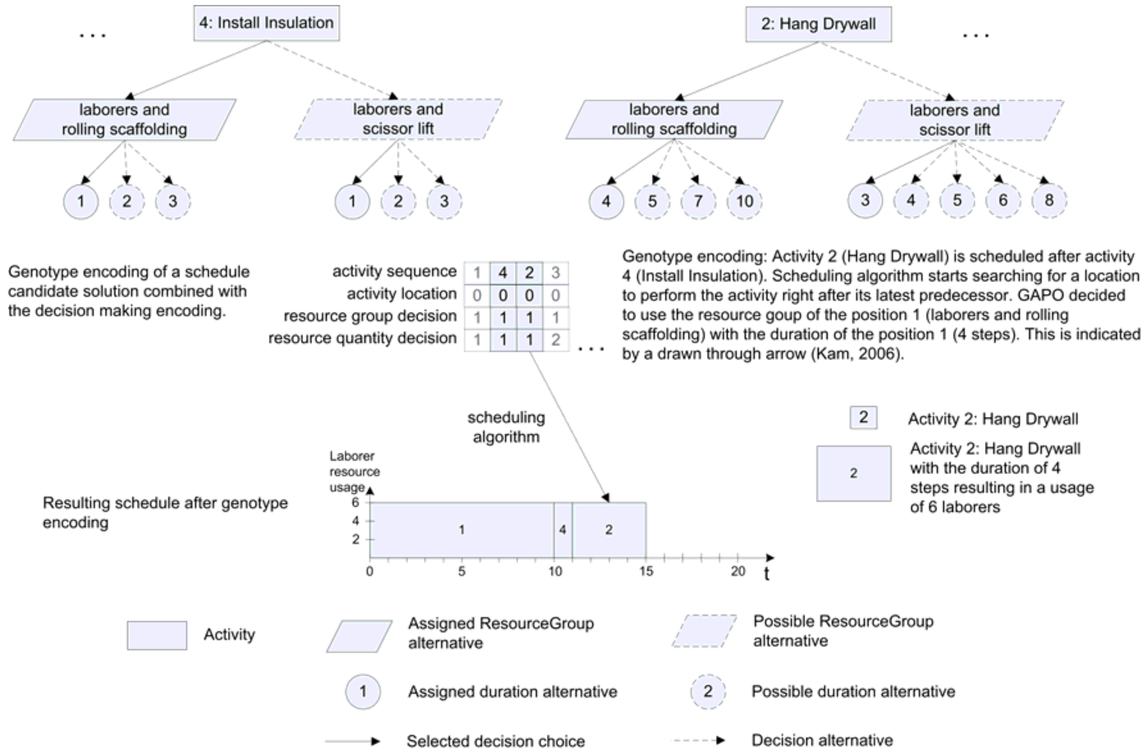

Figure 18: This figure illustrates how the schedule optimization genotype is combined with the decision genotype encoding for the decision about the resource group assignment and the decision about the activity duration. Through this combination, GAPO is able to concurrently expose the schedule and the decisions to an optimization process.

GAPO concurrently exposes these four arrays of the genotype encoding to an optimization process. The quality of the resulting schedule is then measured according to the preferences of the involved stakeholders. Consequently, GAPO will come up with the decision alternative combination which results in a Pareto optimal project schedule.

Decision and Schedule Quality

The stakeholders have the possibility to measure the quality of the schedule and consequently the quality of the underlying decisions by using different criteria which can be dynamically adapted to the needs of the stakeholders. Our quality measurement model allows a grouping of the quality measurement criteria according to the stakeholders' preferences. Depending on the stakeholders' priorities, they can also weigh the quality measurement functions as well as the groupings against each other (Figure 19). Specific aspects of the quality of a candidate schedule solution are measured by fitness functions. A fitness function is a particular type of objective function that quantifies the optimality of a candidate solution so that that particular solution can be ranked against all the other solutions[10].

[10] Wikipedia: http://en.wikipedia.org/wiki/Fitness_function (Last accessed: February 11th, 2007)

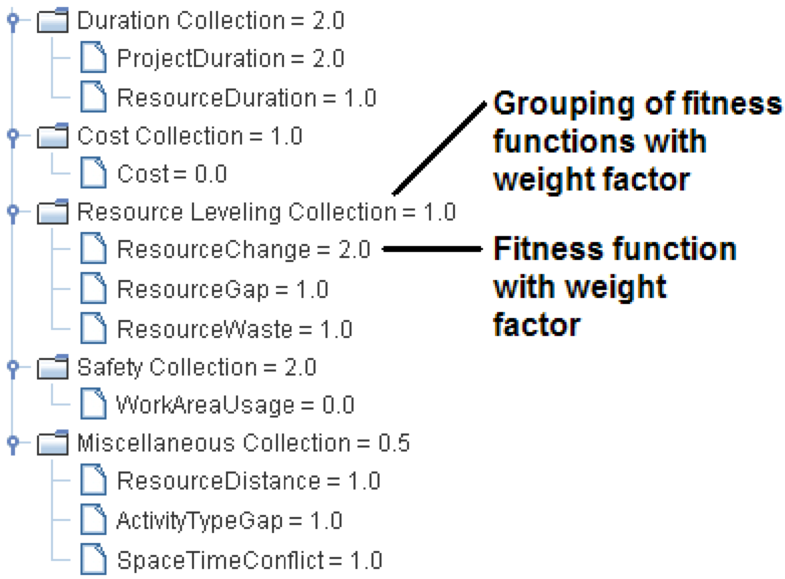

Figure 19: The quality of a schedule candidate solution can be measured by a weighted combination of different fitness functions and fitness function collections. This figure shows a possible grouping of several plausible fitness functions. The Duration Collection contains fitness functions that measure the duration of a schedule. In the particular quality measurement function illustrated in this figure, the ProjectDuration fitness function is weighed double the amount than the ResourceDuration fitness function and the whole collection together has a weight factor of 2. This means that the involved stakeholders prioritize the ProjectDuration fitness function twice the amount than the ResourceDuration fitness function and that they prioritize the Duration Collection twice the amount than, e.g., the Cost Collection. The Cost Collection contains a fitness function that measures the cost of a schedule. There is no other fitness function in the Cost Collection and therefore there is no need to apply a weight factor. The Resource Leveling Collection contains fitness functions which measure how well the resources of a schedule are leveled. The Safety Collection contains a fitness function that penalizes schedules which use work areas simultaneously. This is done to improve safety on the construction site. The fifth and last collection contains some additional functions.

The user can choose between five major types of fitness functions which make statements about the schedule duration, the cost, the resource leveling, safety considerations and some additional fitness functions.

Schedule duration

ProjectDuration: Measures the total project duration. A schedule is considered superior to another schedule if its duration is shorter.

ResourceDuration: Measures the total duration each resource is employed. A schedule is considered superior if these durations are as short as possible. It is possible to weigh the duration of each resource differently (e.g., to make the employment of an expensive crane as short as possible).

Cost

Cost: Integrates the amount of resources used during the project over time and multiplies this amount with a distinct resource cost factor.

Resource levelling

ResourceChange: Integrates changes in the amount of used resources and penalizes these changes. It is possible to weigh the changes of each resource differently.

ResouceGap: Penalizes durations where resources are idle between assignments. It is possible to weigh the idle time of each resource differently.

ResouceWaste: Identifies and penalizes resources which are not used to their full capacity. It is possible to weigh each resource differently.

Safety

WorkAreaUsage: Penalizes schedules which use work areas simultaneously.

Miscellaneous

ActivityTypeGap: Penalizes schedules where activities with the same type are not performed in one sequence. The more often the sequence is interrupted the higher the penalty. This prioritizes schedules where a work force can move in, do its work and leave without having to return again.

WorkAreaGap: Penalizes schedules where activities assigned to a work area are not performed in one sequence. The more often the sequence is interrupted the higher the penalty. This prioritizes schedules where work in one location of the construction site is finished before work forces move on to the next location.

ResourceDistance: Measures the distance resources have to cover between assignments. A schedule is considered superior to another schedule if the overall distance is shorter. It is possible to weigh the distance of each resource differently. If activities and consequently resources are assigned to 3D components, the distance resources have to cover between assignments is defined by the distance between the 3D components. If the activities are not assigned to 3D components, the distance is defined by the distance between work areas.

***TimeSpaceConflict*:** Measures the duration where resources assigned to a specific activity have space conflicts with other resources assigned to other activities. A schedule is considered superior to another schedule if the time space conflict duration is shorter. Space requirements are computed based on the 3D component an activity is assigned to and the resources used to perform the activity.

Results

Formalizing a decision scenario as described so far in this paper allows construction planners to optimize the schedule and resource allocations with GAPO. We will illustrate the process and results of this optimization process using the case example

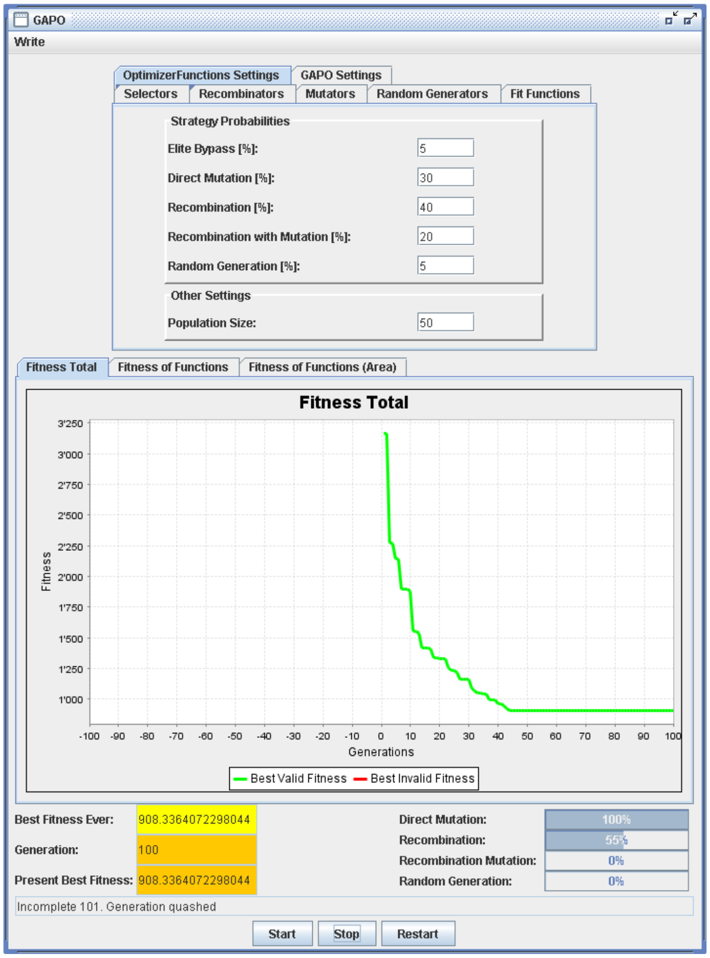

Figure 20: GUI to configure GAPO and run the optimization process. The top part of the GUI illustrates how the GAPO Evolution model is configured (see chapter on GAPO Evolution Model). The middle part shows a graph which describes the development of the best fitness value over time. The fact that the graph starts to level out indicates that a Pareto optimal solution has been found. The bottom part of the GUI shows the actual fitness value of the so far best found solution, the actual generation of the GA optimization process, and how many new solutions each major evolution strategy has produced in the current generation.

The first objective is to find those decisions which lead to the shortest project schedule as defined by the *ProjectDuration* fitness function. GAPO finds the shortest schedule by using the resource group *laborers and scissor lift* for all activities and making activities whose durations are as short as possible. Consequently, the maximal possible amount of each resource is used for most of the activities. This results in a schedule whose activities have to be done in sequence. According to the DBS, some activities can have durations of one or one and two days. These activities do not consume the maximum amount of resources. Therefore, if resource constraints allow it, GAPO appropriately schedules these activities in parallel (Figure 21) making the schedule duration as short as possible.

		Vorgangsname	Dauer	Anfang	Ende	Vorgänger
1		73: Install Metal Stud	10 Tage?	So 11.02.07	Di 20.02.07	
2		65: Install Metal Stud	10 Tage?	Mi 21.02.07	Fr 02.03.07	
3		57: Install Metal Stud	10 Tage?	Sa 03.03.07	Mo 12.03.07	
4		49: Install Metal Stud	10 Tage?	Di 13.03.07	Do 22.03.07	
5		41: Install Metal Stud	10 Tage?	Fr 23.03.07	So 01.04.07	
6		33: Install Metal Stud	10 Tage?	Mo 02.04.07	Mi 11.04.07	
7		1: Install Metal Stud	10 Tage?	Do 12.04.07	Sa 21.04.07	
8		9: Install Metal Stud	10 Tage?	So 22.04.07	Di 01.05.07	
9		17: Install Metal Stud	10 Tage?	Mi 02.05.07	Fr 11.05.07	
10		25: Install Metal Stud	10 Tage?	Sa 12.05.07	Mo 21.05.07	
11		28: Install Insulation	1 Tag?	Di 22.05.07	Di 22.05.07	10
12		12: Install Insulation	1 Tag?	Mi 23.05.07	Mi 23.05.07	8
13		4: Install Insulation	1 Tag?	Do 24.05.07	Do 24.05.07	7
14		20: Install Insulation	1 Tag?	Fr 25.05.07	Fr 25.05.07	9
15		60: Install Insulation	1 Tag?	Sa 26.05.07	Sa 26.05.07	3
16		68: Install Insulation	1 Tag?	So 27.05.07	So 27.05.07	2
17		76: Install Insulation	1 Tag?	Mo 28.05.07	Mo 28.05.07	1
18		52: Install Insulation	1 Tag?	Di 29.05.07	Di 29.05.07	4
19		36: Install Insulation	1 Tag?	Mi 30.05.07	Mi 30.05.07	6
20		44: Install Insulation	1 Tag?	Do 31.05.07	Do 31.05.07	5
21		50: Hang Drywall	3 Tage?	Fr 01.06.07	So 03.06.07	18
63		71: Frame Wall-Beam Inte	1 Tag?	Di 21.08.07	Di 21.08.07	58
64		78: Frame Openings	1 Tag?	Di 21.08.07	Di 21.08.07	62
65		79: Frame Wall-Beam Inte	1 Tag?	Di 21.08.07	Di 21.08.07	62
66		62: Frame Openings	1 Tag?	Mi 22.08.07	Mi 22.08.07	59
67		72: Apply Fire Caulking	1 Tag?	Mi 22.08.07	Mi 22.08.07	60;63
68		80: Apply Fire Caulking	1 Tag?	Mi 22.08.07	Mi 22.08.07	64;65
69		54: Frame Openings	1 Tag?	Do 23.08.07	Do 23.08.07	56
70		55: Frame Wall-Beam Inte	1 Tag?	Do 23.08.07	Do 23.08.07	56
71		64: Apply Fire Caulking	1 Tag?	Do 23.08.07	Do 23.08.07	66;61
72		40: Apply Fire Caulking	1 Tag?	Fr 24.08.07	Fr 24.08.07	53;55

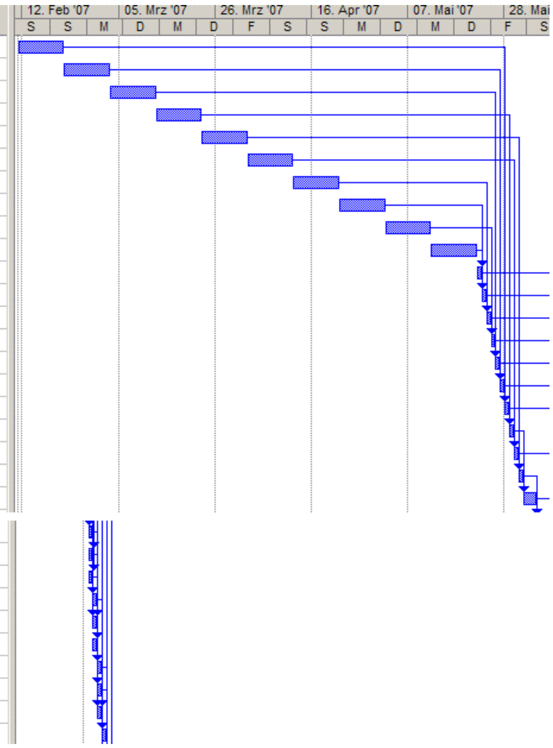

Figure 21: Resulting schedule if GAPO is configured to search for the decisions which lead to the shortest schedule as defined by the ProjectDuration fitness function.

Our second objective is to find the cheapest project schedule as defined by the *Cost* fitness function. GAPO finds a schedule which uses the resource group *laborers and scissor lift* for all activities. GAPO decides about the duration of the activities in such a way that it is possible to schedule the activities concurrently (Figure 22).

		Vorgangsname	Dauer	Anfang	Ende	Vorgänger
1		25: Install Metal Stud	28 Tage?	So 11.02.07	Sa 10.03.07	
2		33: Install Metal Stud	28 Tage?	So 11.02.07	Sa 10.03.07	
3		41: Install Metal Stud	28 Tage?	So 11.02.07	Sa 10.03.07	
4		17: Install Metal Stud	10 Tage?	So 11.03.07	Di 20.03.07	
5		9: Install Metal Stud	28 Tage?	Mi 21.03.07	Di 17.04.07	
6		65: Install Metal Stud	14 Tage?	Mi 21.03.07	Di 03.04.07	
7		73: Install Metal Stud	14 Tage?	Mi 04.04.07	Di 17.04.07	
8		12: Install Insulation	1 Tag?	Mi 18.04.07	Mi 18.04.07	5
9		36: Install Insulation	1 Tag?	Do 19.04.07	Do 19.04.07	2
10		49: Install Metal Stud	14 Tage?	Fr 20.04.07	Do 03.05.07	
11		52: Install Insulation	1 Tag?	Fr 04.05.07	Fr 04.05.07	10
12		28: Install Insulation	1 Tag?	Sa 05.05.07	Sa 05.05.07	1
13		50: Hang Drywall	4 Tage?	So 06.05.07	Mi 09.05.07	11
14		26: Hang Drywall	4 Tage?	Do 10.05.07	So 13.05.07	12
15		51: Apply Tape	11 Tage?	Do 10.05.07	So 20.05.07	13

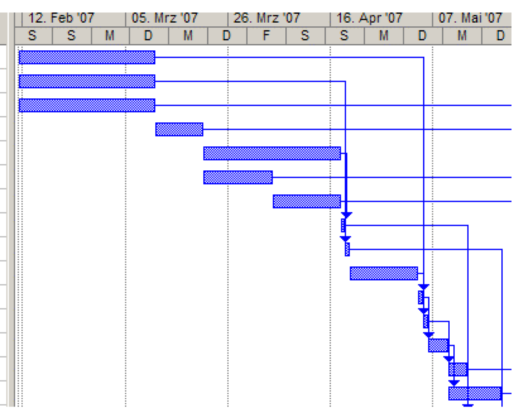

Figure 22: Resulting schedule if GAPO is configured to search for the decisions which lead to the cheapest schedule as defined by the Cost fitness function.

This result does not match our initial intuition. The cost of the resource *scissor lift* with $130 per day is much higher than the cost of the resource *rolling scaffolding* with $40 per day. However, the fact that the activities can be performed faster by using *scissor lifts* affects the overall cost of the schedule in two ways. First, the cost increase of using *scissor lifts* is smaller then the factor of four as one might expect from the cost ratio. Second, the employment duration of the resource *laborers* decreases by using *scissor lifts*. This results in a further reduction of the overall costs. This illustrates how GAPO's optimization method and exhaustive search for an optimal decision given the user's criteria can find solutions whose benefits are not immediately apparent.

A subsequent sensitivity analysis shows that the cost of *scissor lifts* would have to rise to approximately $145 per day to make it worthwhile to choose the cheaper but less effective *rolling scaffolding*.

Limitations

Changes in the process methods can also lead to changes in activities. Consider the example about the erection of a wall using a resource group consisting of builders or a resource group consisting of a crane operator and a crane which lifts a prefabricated wall in place. If this case is planned in more detail, one would have to add an activity which makes sure that the bricks and cement mix are ready when they are needed by the builders. This is not necessary for the resource group which uses the prefabricated wall. Our genotype encoding builds on the assumption that the underlying activity network (activities with precedence constraints) does not change. This allows preprocessing steps which help to speed up the overall optimization process. This makes it impossible to delete or replace existing activities or introduce new activities into the network. In order to support any kind of process method changes, it would be necessary to support changes in activities.

Currently, GAPO only supports activity networks with finish-start constraints. Project planners also use start-start, finish-finish, and start-finish constraints with different lag durations. This limitation will need to be eliminated for GAPO to become a tool that can be used in industry. However, this adaptation should be straightforward since it is possible to transform an activity network with any kind of constraint into an activity network having only finish-start constraints. This procedure implies that the durations of newly introduced activities are dependent on durations of other activities. Because GAPO can change resource amount assignments which results in activity duration changes, it becomes necessary to introduce a new kind of constraint which makes durations of activities dependant on durations of other activities.

Practical significance

GAPO needs approximately 30 to 40 seconds to find a Pareto optimal decision combination for the case example which leads to a Pareto optimal schedule which conforms to the quality measurement criteria defined by the involved stakeholders. This speed and the ability to dynamically adjust and combine the quality measurement criteria the decisions are based on shows the power of the scheduling decision optimization tool described in this paper. It allows stakeholders to determine good decisions quickly and negotiations or alternative exploration leading to the final decision can be done in real time, for instance during a meeting, such as Integrated Concurrent Engineering (ICE) sessions where short latency is extremely important (Chachere, 2004).

The decision genotype encoding as it is described in this research can be used to represent arbitrary decision problems. This makes it possible to reuse the described concept for decision optimization problems which focus on other problems than project planning.

Next Steps

Next to eliminating the limitations described in the chapter on Limitations we are also considering building the decision genotype encoding and the optimization algorithm into Kam's Decision Dashboard (Kam, 2006). This would enhance the Decision Dashboard's functionality and a quick optimization of arbitrary decisions would become possible.

The next level in automated project planning and automated decision making would be to include design alternatives in the optimization. Having the computer generate new design alternatives promises to be a very interesting further research topic. Koza et al. (2003) have already demonstrated that the computer is capable of generating new patentable innovations. Their findings might provide useful ideas about how computers could be used to generate new construction process designs.

Summary

This paper illustrates how a Genetic Algorithm is used to concurrently optimize schedule decision alternatives as well as the schedule itself. The decision making stakeholders can dynamically adapt the quality measurement criteria according to their preferences. This research focuses on decisions alternatives about resource quantities and resource group assignments to activities. Nonetheless, the decision genotype encoding is generalized in such a way that it could be reused to optimize arbitrary decision optimization problems. The overall performance of the optimization algorithm shows a short latency in acquiring a Pareto optimal decision alternative combination. This suggests that the algorithm could be utilized for decision optimization assignments where short latency is extremely important like meetings or ICE sessions.

References

Aalami, F.; Kunz, J.; Fischer, M. (1998): Model-Based Mechanisms for Automated Activity Sequencing. Center for Integrated Facility Engineering, Stanford University. Stanford, CA: Stanford University.

Bacchus, F.; van Run, P. (1995): *Dynamic Variable Ordering in CSPs*. Principles and Practice of Constraint Programming – CP-95.

Chachere, J., Kunz, J., Levitt, R. (2004). Observation, Theory, and Simulation of Integrated Concurrent Engineering: Grounded Theoretical Factors that Enable Radical Project Acceleration. Center for Integrated Facility Engineering, Stanford University. Stanford, CA: Stanford University.

Coley, D. A. (1999). *An Introduction to Genetic Algorithms for Scientists and Engineers.* Danvers, USA.

Ginsberg, M. (1993): *Dynamic Backtracking*. JAIR 1.

Gonçalves, J. F., Mendes, J.J.M., Resende, M. G. C.; (2002). *A hybrid genetic algorithm for the job shop scheduling problem.* AT&T Labs Research.

Jenkins, M. (2006): Guest speaker in Class CEE 241 Techniques of Project Planning and Control, Topic: Dealing with Uncertainties in Project Planning from a General Contractors Perspective, Department of Civil Engineering, Stanford University. Stanford 2006, USA.

Kam, C. (2005): *Dynamic Decision Breakdown Structure Ontology, Methodology, and Framework for Information Management in Support of Decision-Enabling Tasks in the Building Industry*. Civil and Environmental Engineering of Stanford University, USA.

KHosraviani, Bijan, Raymond E. Levitt and John R. Koza (2004): *Organization Design Optimization Using Genetic Programming* presented at the GECCO conference in Seattle, USA, June 2004 (can be downloaded from: http://crgp.stanford.edu/publications/conference_papers/BijanGecco2004.pdf).

Kondrak, G.; van Beek, P. (1995): *A Theoretical Evaluation of Selected Backtracking Algorithms*. Artificial Intelligence 89.

Koza J.R., Keane M.A, Streeter M.J., Mydlowec W., Yu J., Lanza G. (2003): *Genetic Programming IV: Routine Human-Competitive Machine Intelligence*, Kluwer Academic Publisher (2003)

Märki, F.; Suter, M. (2003): *Projektplanoptimierung mit Genetischen Algorithmen*. Diploma thesis, unpublished, University of Applied Sciences Aargau, Switzerland.

Märki, F.; Vogel, M.; Breit, M.; Fischer M. (2006): *Interactive Toolbox for 4D-Modeling*. Published at the Joint International Conference on Computing and Decision Making in Civil and Building Engineering 2006. Paper number IC-293, Montreal, Canada.

Prosser, P. (1993): *Hybrid Algorithms for the Constraint Satisfaction Problem*. Computational Intelligence 9(3).

Sengoku, H.; Yoshihara, I. (1998): *A Fast TSP Solver Using GA on JAVA*. Proceedings of the 3rd International Symposium on Artificial Life and Robotics (downloaded from: http://www.gcd.org/sengoku/theme.en.html, last access 10/18/2006)

Staub-French, S.; Fischer, M.; Kunz, J.; Paulson, B.; Ishii, K. (2002): *A formal process to create resource-loaded activities related to feature-based product models.* Civil and Environmental Engineering of Stanford University, USA.