SPE 112101

Rescheduling in a Collaborative Environment

Bård Henning Tvedt, Jan-Erik Nordtvedt, SPE, and Frédéric Verhelst, SPE, Epsis AS

This paper was prepared for presentation at the 2008 SPE Intelligent Energy Conference and Exhibition held in Amsterdam, The Netherlands, 25–27 February 2008.

Abstract

Scheduling and rescheduling strategies are used at several levels on a daily basis in the oil industry to among others have control on safety regulations, improve resource utilization and optimize profit. With the emerging integrated operations and focus on collaboration new work methods also become possible within scheduling.

Access to easily shareable information from multiple locations reduces the response time, which in turn make high performance decision support tools for rescheduling attractive. A rescheduling tool produces globally consistent schedule proposals based on information from the different disciplines involved in the planning activities. These proposals can be edited by the participants in the collaborative environment, and new feasible schedules are generated.

In this paper, several simplified scenarios have been executed to simulate and exemplify situations where the original schedule is deteriorated so that rescheduling must be applied to try to keep the schedule objective. In all scenarios our rescheduling strategies maintains the objectives of the original schedules. These scenarios are used to illustrate the potential of using collaborative environments in rescheduling processes.

Introduction

Large initiatives have been launched in the oil industry to implement integrated operations. An increasing number of examples show that these implementations lead to faster and better decision making (e.g., OLF (2007); Unneland and Hauser (2005); Ouimette and Oran (2006), Holst og Nystad (2007)). A key part of oil and gas exploitation is planning of operation and maintenance tasks or activities. More continuous data and information as well as changes in work processes towards more proactive decision-making is starting to influence planning and scheduling; this apply for both initial scheduling generation as well as scheduling during execution (scheduling during execution is hereafter referred to as rescheduling).

Today's demand for oil and gas means increased profit if projects are completed as early as possible. On the other hand, the high activity level reduces the availability of the necessary competence and equipment, thus increasing expenses. Optimizing schedules with regard to efficient resource utilization can therefore both increase production and reduce the related costs. The value potential of Integrated Operations on the Norwegian Continental Shelf was estimated to about 250 billion NOK or about 46 billion USD in 2005. About 12% of this amount is attributed to the reduction of operations and maintenance costs, i.e. about 30 billion NOK or 5.5 billion USD (OLF, 2006). These figures were recently re-estimated, adding an additional 20% to the value potential on the Norwegian Continental Shelf (OLF, 2007).

There is a broad diversity among the different assets within the E&P industry. This is reflected in the nature of the associated operational and maintenance activities. Regardless of the diversities, a common viewpoint within the industry is that scheduling of activities is complex, time consuming and that there is ample room for improvement. Asset location (onshore or offshore) is an important field characteristic. Onshore fields often have high numbers of wellheads that are spread over large geographic areas, whereas offshore fields have fewer wells and all the wellheads and installations located at one or just a few locations. For offshore assets limitations on storage capacities, challenges related to the supply chain or HSE issues related to hot permit work orders may be further complicating factors. Another important factor is the age of the field and the associated installations.

A complex scheduling problem contains several challenges. Schedules must be feasible meaning that all possible conflicts in the system must be resolved. A prerequisite is good communication lines, both between the different departments and contractors on an asset and between the schedulers and workers who execute the schedule. For rescheduling an extra factor is needed, namely a rapid response time.

In this paper we illustrate through simplified scenarios how a schedule's possible deterioration can be minimized by rescheduling strategies set in a collaborative environment.

Integrated scheduling

Scheduling is often defined as the allocation of scarce resources to activities over time (K. R. Baker, 1974). During execution a schedule may be rendered infeasible by unforeseen events like for instance critical maintenance. Rescheduling is the way a schedule is reworked so that it again is feasible. In both cases an objective can be added to the schedule generation strategy.

Scheduling challenges

The scheduling problems vary from asset to asset. They have all their specific considerations to take into account due to the differences in operating conditions, size and age of equipment. Not only do the problems themselves vary, but so do also the approaches the assets are taking.

The scheduling requirements differ also between disciplines within one asset. The various needs can give rise to different constraints on the problem, and several specialized disjoint systems are often in use at one field. An asset may therefore have several unrelated scheduling problems. Each of them have less complexity than an integrated asset schedule, but they lack the overall view of the entire problem.

In addition to their discipline dependent constraints there may also exist asset dependent ones. A good example is the bedding and transportation limitations on an offshore field. To ensure feasibility the schedules are typically coordinated during weekly or daily scheduling meetings. Often the schedules for the individual disciplines are available as print-outs or in individual scheduling software applications. This means that they cannot readily be shared among the other participants in the meeting and that a large part of the coordination has to be done manually.

The various discipline schedules may be optimized according to an objective. Several different objectives that may be contradictory are difficult to maintain manually. The need for rescheduling appears when new or previously undiscovered conflicts arise. In order to reduce idle time of crews and downtime of resources they must be reassigned to new activities, and the response time of the rescheduling process ought to be as fast as possible.

Facing the challenges

The first step to face these challenges is to improve control of all the constraints, especially the interdisciplinary ones. The second step is to increase collaboration, and examples on situations where this can be done are discussed in the next section.

Integration of schedules can either be done manually, automatically or as a combination. Manually produced schedules are time consuming and can in worst cases be unfeasible from the start. Automatically generated schedules are feasible to the extent the model is correct, but a model will only be a simplified representation of reality. Combining the two will present the schedulers with a feasible schedule which can be edited, and ensure that the final schedule still is feasible.

If several departments do their scheduling integrated the constraints are an intrinsic part of the problem. When the constraints are modeled as part of the problem all solutions are guarantied to satisfy them. The complexity of the problem increases, and a concise model may be harder to produce. There is also a risk that by elevating the scheduling process to a higher level the discipline dependent expert knowledge is reduced. Another approach is to add a scheduling software on top of the already existent applications. A drawback is that the intuitive understanding of a discipline's schedule may be lost because it is externally constrained. An integrated solution relies heavily on a consensus of modeling aspects. Optimization of a schedule can be done based on user defined goals. Often the different goals can be contradictory. Other challenging modeling aspects can be the implementation of best practice or customs. It is straightforward to enforce a temporal constraint such that activity A must end before activity B can start, but an undocumented best practice (e.g., this is a task that one specific crew member does best or prefers to do) as a constraint is cumbersome to model.

Regardless of approach a generic framework for modeling activities and constraints is beneficial. Constraint Satisfaction Programming is a framework well suited to incorporate different constraints, making it possible to integrate the discipline's varying needs (Baptiste et. al., 2001). The problem model is also explicitly separated from the problem solving strategy.

Constraint based scheduling (CBS) is a subset of the constraint satisfaction framework. A model in CBS consists of activities, resources and constraints. An activity contains variables like start times, durations and end times. Constraints can be added to both these variables, and to the different resources that activities use. A solution to the problem is an instantiation of when the activities are being executed so that all temporal and resource constraints are satisfied; see Appendix A for more information about the framework. In this work ILOG's CP is used (ILOG, 2007). It is a sophisticated C++ library consisting of classes for modeling, solving and optimization of constraint satisfaction problems.

Saga used constraint satisfaction to solve well activity scheduling already in 1995 (Hasle et. al., 1995). The project showed promising results – even without optimization techniques. The constraint satisfaction framework has since then developed and optimization of user defined objectives is now possible. Recently several other projects in the industry have published their work in the area, among them the drilling and petroleum engineering program at Santos Ltd (Horton and Dedigama, 2006), Stuart Petroleum's approach to delivering exploration, development, and production work programs (MacDougall and Saman, 2006) and Saudi Aramco's rig scheduling (Irgens and Lavenue, 2007). They all show promising results and indicate the flexibility of the constraint satisfaction framework.

Rescheduling in collaborative environments

Coordination of the individual schedules (or, each discipline's schedule) is a collaborative effort. Ouimette and Oran (2006) describe how a collaborative environment is used for visualization of the different schedules. Possible conflicts were detected and the necessary decisions as to how to resolve them were made. This project was piloted at an onshore brown field. The use of a visualization environment (see, e.g., Figure 1) showed that the decision making was improved.

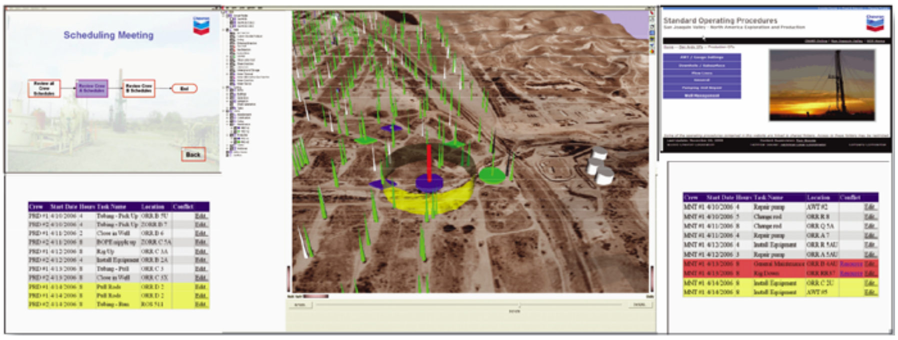

Figure 1: Example of a visualization environment showing conflicts between different schedules.

The scheduling meeting may also be extended by involving participants from different locations. A typical example is the offshore Brage field, where people offshore are remotely participating in a scheduling meeting (Holst and Nystad, 2007). At Brage the integrated scheduling is done onshore, but offshore needs to approve all HSE and production-critical work-orders.

Until now the collaboration environment has mainly been used for its visualization capabilities, i.e. for sharing a common understanding of the current plan. A natural extension is to include decision support tools to ease the decision making in the planning phase. A decision support tool for rescheduling includes asset dependent constraints and provides schedule suggestions based on the information from the discipline dependent systems. The suggestions are also dependent on the objective, normally one or several KPIs, used when solving the problem. A feasible schedule suggestion can now be visualized and edited according to the participant's expertise. A rescheduling tool ensures that the constraints in the system are satisfied, and facilitate the final step is to update the respective discipline's systems.

The creation and maintenance of integrated and dynamical schedules is a continuously decision process. Figure 2 is an illustration of the schematic scheduling process. The first stage is to model the problem. This is demanding, but is only required once. If changes happen in the real problem it may be enough to update the model since the constraint satisfaction programming framework separates the modeling from the solving phase.

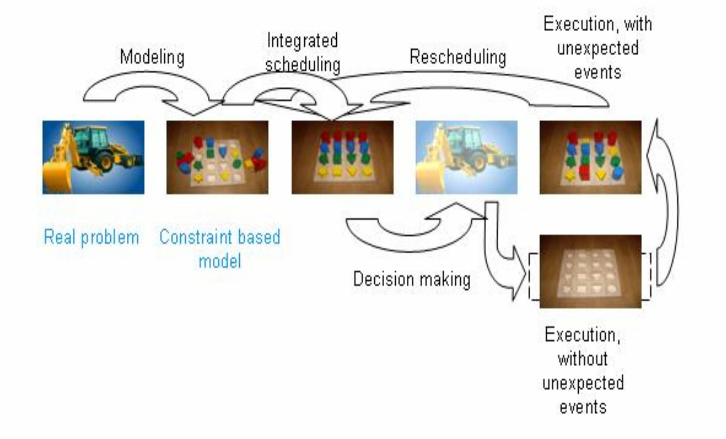

Figure 2: Dynamic scheduling work process.

At a given interval, for instance weekly, an integrated scheduling is performed. The disciplines meet as participants in a collaborative environment as shown in Figure 3. Implementing their combined expert knowledge of the entire system in the decision making process is bringing the executable schedule as close to the real problem as possible.

In a real life situation unexpected events do occur, and if the deterioration of the schedule is large enough a rescheduling must be performed in order to try and keep the objective. The participants in the collaborative environment are still representatives from the different disciplines. However, no one knows at what time an unexpected event are going to occur, and a collaborative environment allows people to participate from different locations, for instance through a pocket computer. After a quick rescheduling the new schedules are shared, and the changes are communicated to the involved people.

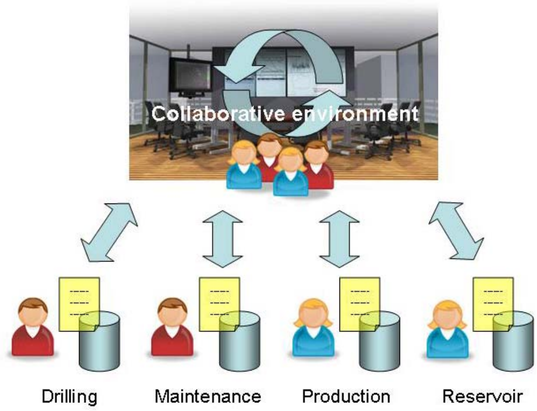

Figure 3: Rescheduling in a collaborative environment.

Deterioration of a schedule can be caused by new activities that require immediate attention, work progress delays, changes in resource availability, and missing information like activity dependencies. In the next section the effect of critical activities and missing dependencies are simulated. Rescheduling can also be used to analyze what if scenarios. What if an activity is delayed? What if a resource is suddenly unavailable? The use of collaborative environments and integrated scheduling strategies enables also sharing of resources between disciplines.

In all the cases collaboration is important. The right people need the right information at the right time. To produce a good as possible schedule people from all areas that are dependent on the schedule ought to be involved. Such a procedure is also essential to ensure that the schedule is properly anchored in the organization. It will ease schedule execution if all parts of the organization have the appropriate buy-in and ownership.

Scheduling scenarios

Problem description

An artificial simplified problem was created for generating weekly schedules. It consists of activities located in a backlog. The goal is to schedule these activities as optimal as possible given different objectives. Extra activities and dependencies are then added to simulate the need for rescheduling. The rescheduling strategies use the same objectives and try to minimize the deterioration of the schedule.

The backlog contains 100 activities, and they have all an associated value between 100 and 1000. Activities with duration shorter than 10 hours are modeled as non-preemptive, i.e., they must be finished at the same day as they start. All activities have a release time at zero hours and no deadline. Each activity requires one crew and is executed at one location. There are 20 locations and 5 different crews. The locations are modeled as unary resources meaning that only one activity can be executed at each location simultaneously. The crews are similarly modeled as unary resources, but with the additional constraint that they are available 10 hours per day. All crews are working the same interval, and weekends off are not included. For simplicity it is assumed that the locations are geographically close and that traveling time between activities are implemented in the time span of the activities.

An arbitrary number of temporal or relational constraints can be added between the activities. The dependencies used in the scenarios are shown in Figure 4. Constraints are imposed between the activities of the same color, i.e., activity A2 is dependent on A1, and A52 is dependent on A51.

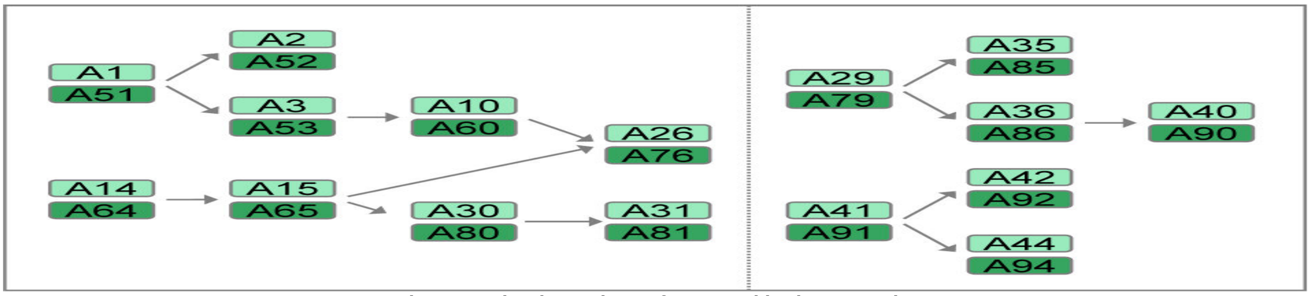

Figure 4: The dependency form used in the scenario.

Two different objectives have been used. One objective is to minimize the latest end time of all the activities, i.e., minimizing makespan, another is to maximize the value of the activities performed during the week. The makespan is a well known optimization criteria, and if C_i denotes the completion time of i^{th} activity it can be minimized according to

$$\min(F) = \min(C_{\max}), \quad \text{(eq. 1)}$$

where $C_{max} = max(C_i)$. This criterion is effective since the upper bound is directly propagated at the end time of each activity. It returns however an earliest end time of the entire backlog and therefore not necessarily the most favorable activities within a weekly schedule. For this an objective working on the respective values of the activities has been employed. An activity's value can for instance be related to profit or HMS issues. Maximizing the value V is done according to

$$\max(V) = \sum_i \left(V(A_i)\left(C(A_i) \leq end(week)\right)\right) \quad \text{(eq. 2)}$$

where $C(A_i) \leq end(week)$ is a Boolean constraint returning unity if the activity is executed before the end of the week and zero otherwise. The propagation of this objective is much more complex, because it must be considered at each step in the search tree. Regardless of the objective a scheduling problem is NP hard, i.e., there are no known algorithm that solves the problem to optimality in polynomial time. Also, finding a global optimal solution is time consuming. An iterative process has been employed to search for local optimal solutions instead. The objectives are modeled as decision variables for instance $V \in [0, \sum V(A_i)]$. If feasible solutions exist, the bounds on the decision variable are updated. The new bounds are used to search for an improved solution. An upper limit of 30 seconds is set as the search length for each iteration. All simulations have terminated with 8 or less iterations. All solutions are therefore produced within 4 minutes.

Scenarios Performed

Three different scenarios have been performed. The first two illustrate situations where rescheduling is needed while the third consider how increased collaboration in itself can increase efficiency.

During the execution of a weekly schedule there is a high probability that unforeseen events will take place. A compressor might for instance shut down and require immediate attention. Scheduled activities must be postponed in order to fix the problem, and the result may be that conflicts arise in the remainder of the schedule. Such a deterioration lead among other things to idle crews. This is particularly unfortunate if e.g., in an offshore situation the crew is only available for a limited time before leaving the installation. Rescheduling is performed in order to reduce the idle time and maintain the schedules' objective. In Scenario #1 (Critical Activity) one critical activity has been added at the beginning of the week. Using collaboration the schedule can be adjusted to make room for the new activity, and to alter it to minimize deterioration. In Scenario #2 (Dependencies), an extra dependency has been added between two activities that are performed at the same location. The dependency may have been overlooked because the activities are performed by different disciplines, but a collaborative environment are well suited to detect interdisciplinary relations, and proper decisions can be made to avoid conflicts.

The final scenario (Scenario #3; Alternative Resource) benefits from the increased integration and communication between the disciplines. As mentioned earlier problems arise when interdisciplinary constraints on resources are not taken into account. With these issues under control an increased sharing of resources can be considered. At for instance an onshore field both maintenance and well operations have crews operating backhoes. These crews can be shared between the disciplines. Two of the five crews have been modeled as alternative resources. This means that during schedule generation the related activities are assigned to one of the two crews. In other words the collaboration is not only helping resolving existing conflicts, but it can help increasing efficiency by turning sharing of resources between disciplines a viable option,

Results

The original schedule solved with minimum makespan is shown in Figure 5. Activity names are indicated by A# above each activity, and the number below refer to location, for example, A51_20 means activity number 51 to be executed at location 20. The activities A3 and A19 are the only ones with duration greater than 10 hours. The colors indicate from light to darker values in the range [0, 300], (300, 700], and (700, 1000] respectively. The minimum makespan is 244 hours.

Figure 6 show the original solution generated with the objective to maximum value within a seven day week. Total value of all activities in the backlog is normalized to 1, and the value returned within the week is 77.5 % of the total. Several activities with relative small values can be scheduled before one activity with larger value if their combined value is greater. The two objectives produce very different schedule proposals.

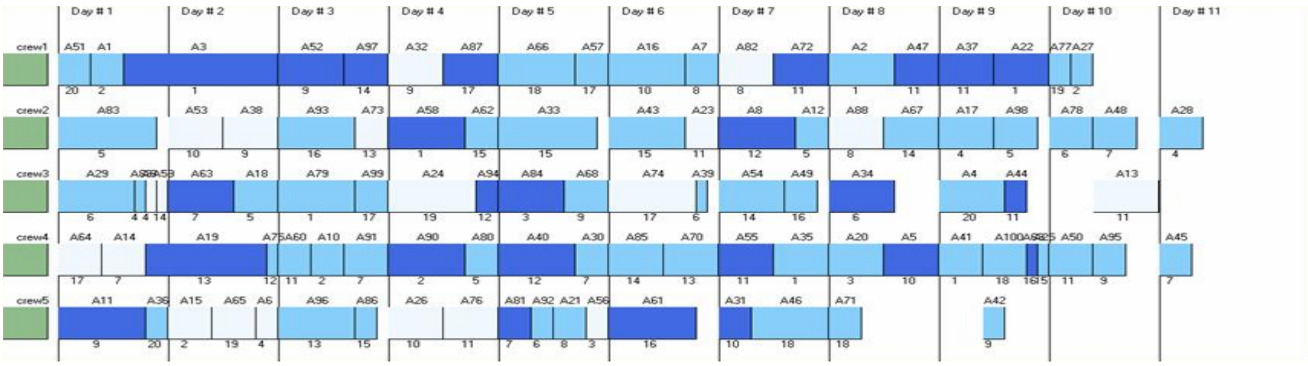

Figure 5: Original schedule with the minimum makespan objective.

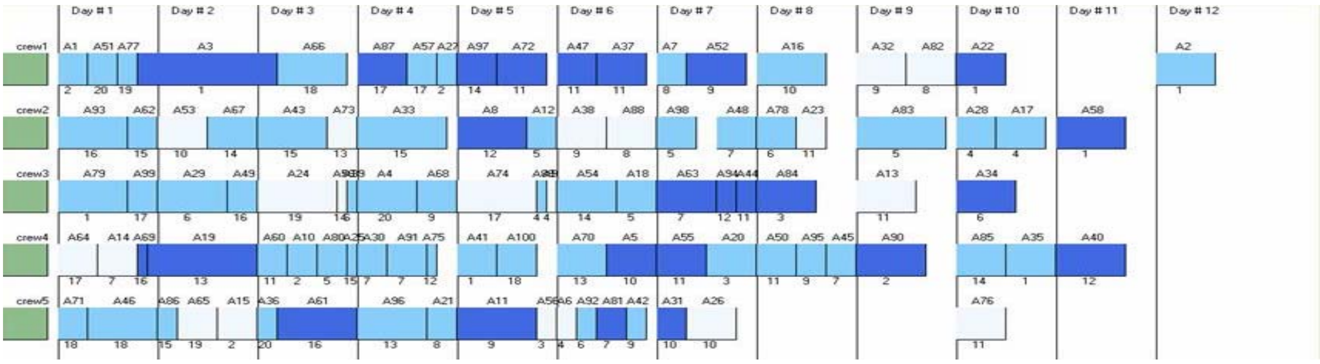

Figure 6: Original schedule with the max value objective.

It is important to emphasis that the schedules produced are proposals which can and probably must be edited by people with expert knowledge about the problem. A model will always be a representation of reality, and may not take into account complex constraints for instance related to best practice. The addition of expert knowledge may both improve a schedule and ease its execution.

Scenario #1: Critical activity

A critical activity is inserted in the interval 0 to 10. It takes place at location 1 and must be performed by crew 3. A rescheduling of the new situation with minimum makespan is shown in Figure 7. Figure 8 shows the schedule with the new critical activity but without any other changes – less those changes needed to resolve the conflicts. The conflicts introduced by the critical activity have been resolved by postponing activities whenever needed. We refer to such a rescheduling as a new schedule without *interaction* in the following. Schedule supervisors will of course make alterations in order to keep the schedule objective as well as possible, but it illustrates how the constraints affect the schedule during changes.

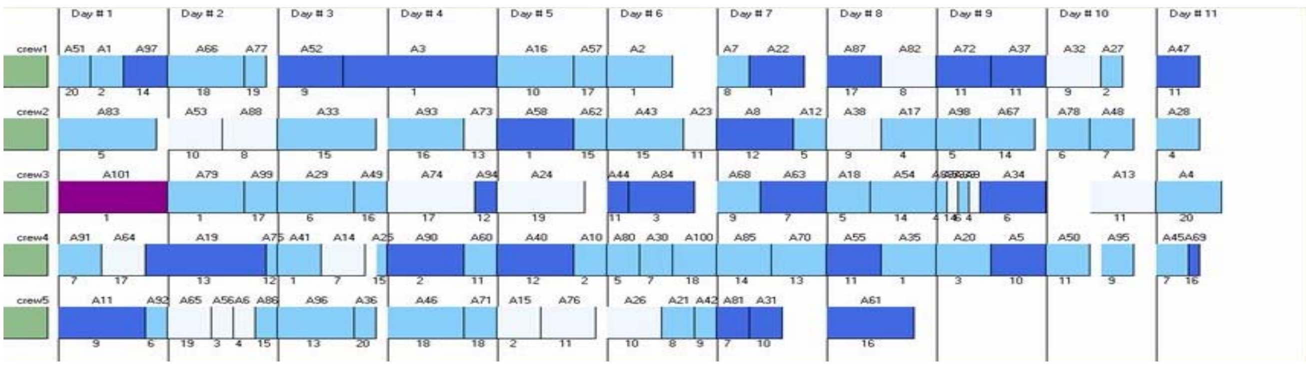

Figure 7: Schedule after inclusion of critical activity A101 with the min makespan objective.

The solution with minimum makespan is now 246 hours. This is two hours more than the original solution. Without any interactions the schedule finishes one day later with a makespan of 268 hours. By introducing one extra activity to a fairly simple scenario with few constraints the effects on the schedule is large. Even though it is only crew 3 that have to perform an extra activity all crews are affected and if no interactions are made crew 2 will be idle for an entire day.

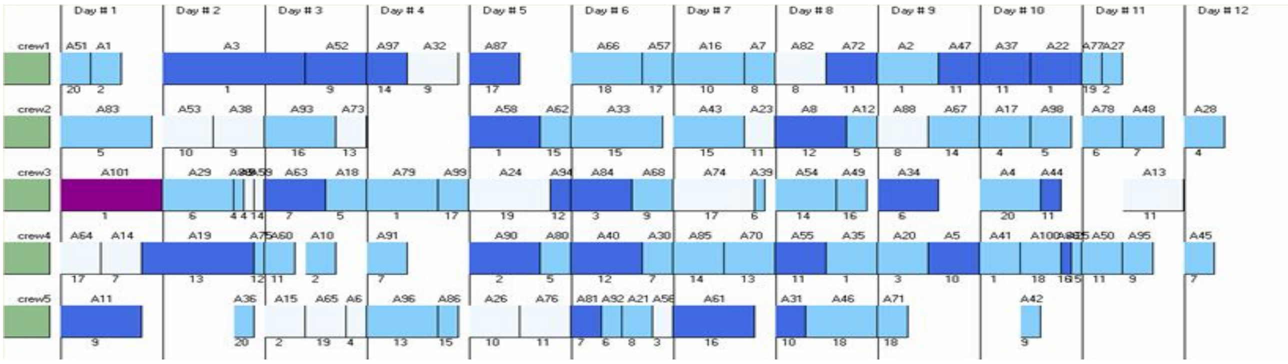

Figure 8: Schedule after inclusion of critical activity A101 without interaction for the original schedule with minimum makespan.

Corresponding results for the maximum value objective are presented in Figures 9 and 10. The total value of the activities, including the critical activity's value of 1100, is now 77.3% of the total, which is only slightly less than the original schedule. Without interaction the value is reduced to 66.9%. **In this case 4 out of the 5 crews are affected in the first seven days. To avoid idle crews the schedules must be reworked during the first half of day one.**

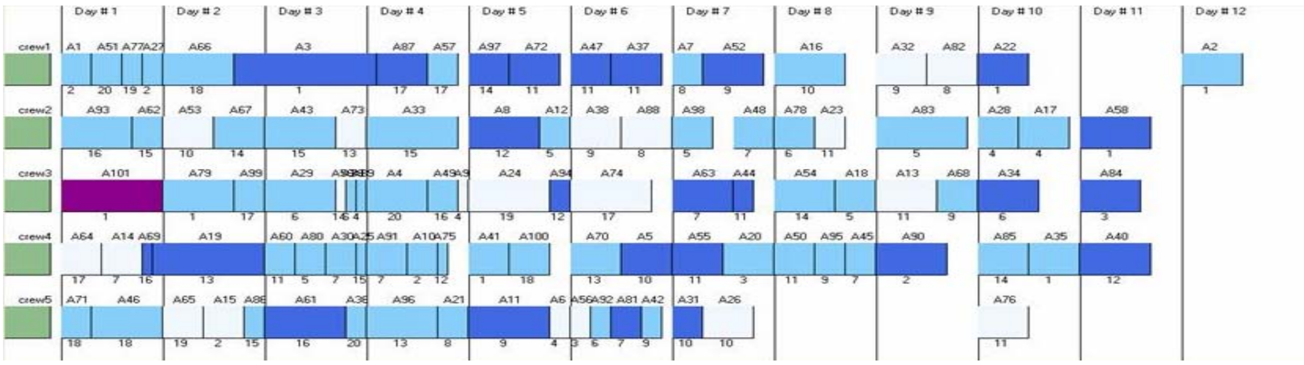

Figure 9: Schedule after inclusion of critical activity A101 with the max value objective.

Both objectives are altering the schedule proposals considerably, and they may be additionally changed by people with expert knowledge. This means that communication of the changes in the schedule to the people who are executing it is of great importance. Collaborative environments are well suited for these changes, because the new schedules can be easily shared among the participants.

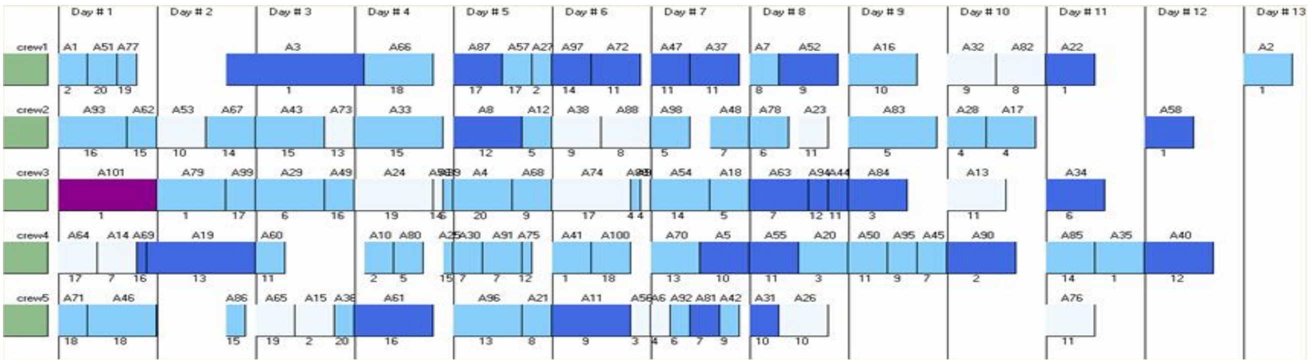

Figure 10: Schedule after inclusion of critical activity A101 without interaction for the original schedule with max value.

Scenario #2: Dependencies

As a result of a collaboration session on the plan one can detect dependencies between activities that was not known or existing during planning of the activity. This can be dependencies that one discipline didn't know existed or new dependencies that has emerged late and not included in the plan. In our case, activity A1 is scheduled to be performed before A15 at location 2. In this scenario an extra dependency stating that activity A15 must be finished before A1 can start is added. The rescheduling results in the case of minimum makespan are shown in Figures 11 and 12.

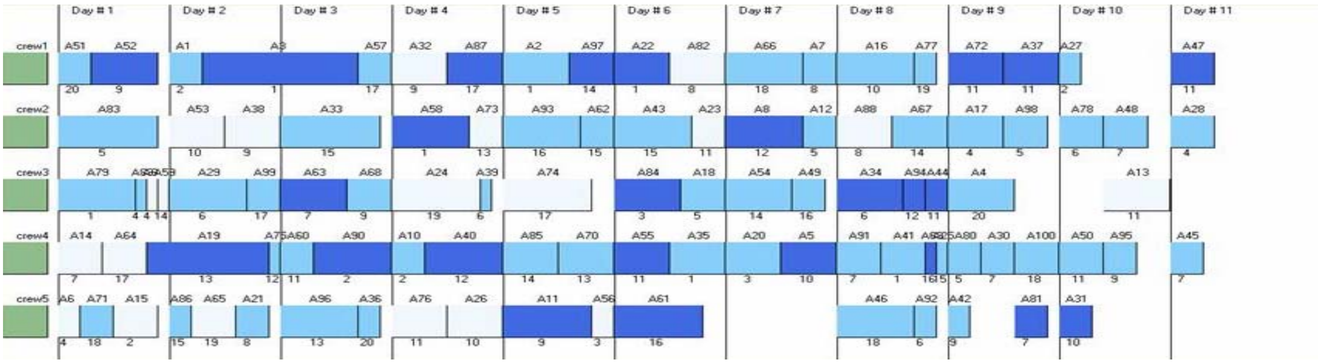

Figure 11: Schedule after inclusion of dependency A15 precedes A1 with the minimum makespan objective.

In this case the minimum makespan objective manages to keep the makespan at 244 hours. , while without interactions show conflicts touching all the crews. The makespan in this case is 272 hours. The effect of the extra dependency is apparent at the first two days for crew 1 where it becomes idle. In order to avoid this, a rescheduling must be performed before activity A51 is finished, i.e. 3 hours are available for rescheduling and communicating the changes to the crews. The postponing of crew 1's activities affect the four other crews heavily between day 3 and 4.

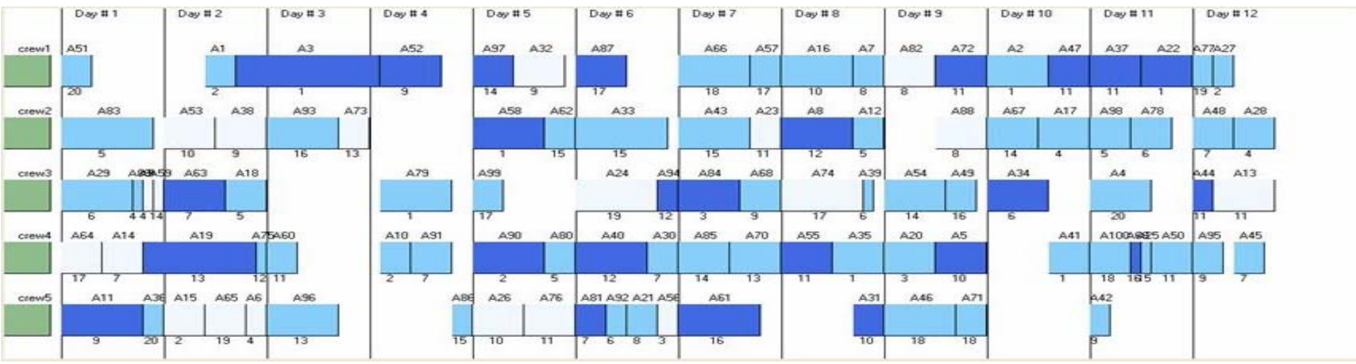

Figure 12: Schedule after inclusion of dependency A15 precedes A1 without interaction for the original schedule with minimum makespan.

The results for the maximum value objective are shown in Figures 13 and 14. The value of the new schedule is 77.5% of the total. In other words the schedule objective is maintained. If no interactions are made the value is reduced to 65.6%. Again it is clear that small changes can have a large impact on the total schedule. Rescheduling must be done immediately since A1 is the first scheduled activity for crew 1.

If other activities or dependencies had been added to the scenarios other changes would have been made. Some of them may have caused fewer disturbances others again would have produced more. By employing a sound rescheduling strategy set in a collaborative environment the asset will most likely be better equipped to deal with unexpected events. It is also possible to run different "what if" scenarios. What if the duration of an activity is longer than expected? Or what if the contractor that is supposed to perform an activity cannot make it in time?

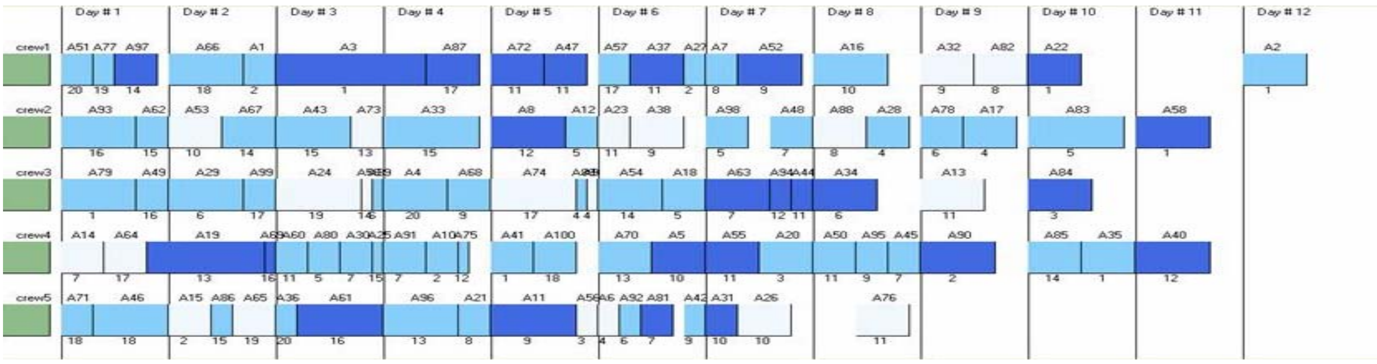

Figure 13: Schedule after inclusion of dependency A15 precedes A1 with the maximum value objective.

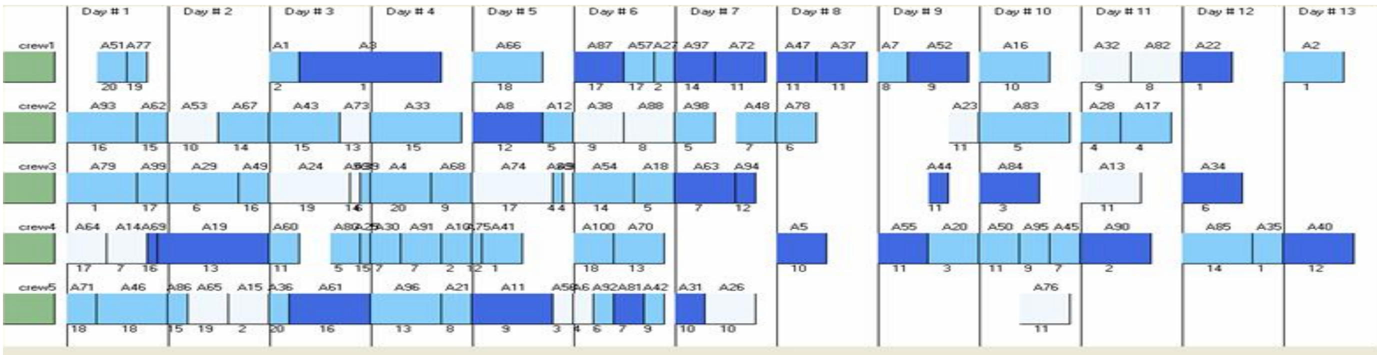

Figure 14: Schedule after inclusion of dependency A15 precedes A1 without interaction for the original schedule with maximum value.

Scenario #3: Alternative resources

If an activity requires one resource from a pool of identical resources this can be modeled as alternative resource constraints. Instead of assigning one fixed resource the solution strategy can choose from the set. In cases where scheduling is done for each discipline splitting the problem into parts meant that they required each separate set of the same resource. Improved control of interdisciplinary constraints and increased communication make sharing of resources between disciplines possible.

In our case it is assumed that crew 2 and crew 5 belong to different disciplines and that they can perform the same activities. The impact on the original schedule with minimum makespan is shown in Figure 15. The makespan of the activity is down to 226, and is at this point limited by the work load on crew 4. Due to collaboration the entire schedule of all the activities in the backlog can be finished one day earlier than originally anticipated.

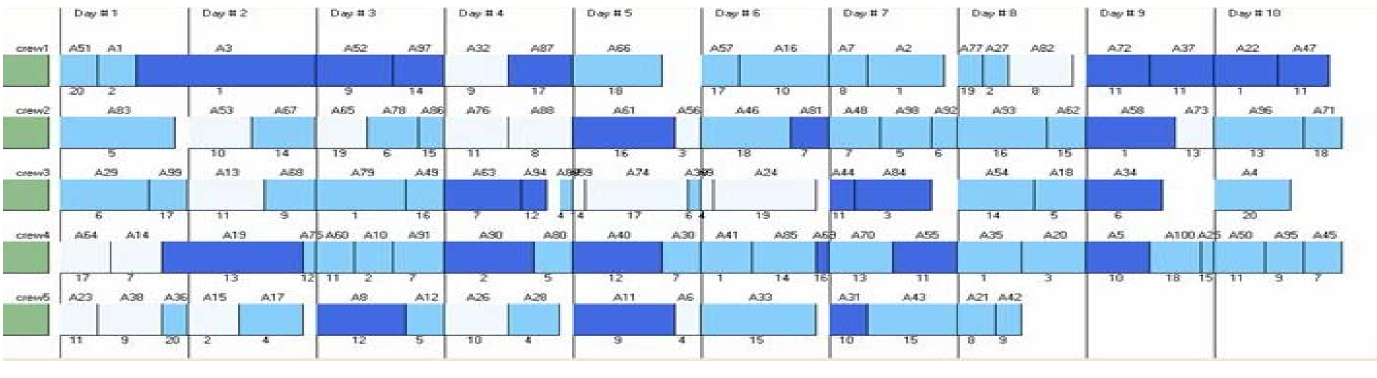

Figure 15: Schedule with min makespan and alternative resources crews 2 and 5.

Conclusions

In this paper we have discussed how a collaborative environment can be used within rescheduling. A simplified model has been used to create three different scenarios in order to simulate rescheduling. In the first two scenarios, adding critical activities and adding dependencies, the schedule was deteriorated with approximately 8%. Our rescheduling strategies removed these deteriorations in all cases. The third scenario represented increased efficiency due to increased flexibility regarding sharing of resources. In that case, an original 11 day schedule can be finished in 10. All the rescheduling (i.e., minimization / maximization solutions) are produced within 4 minutes.

In order to capitalize on these improvements collaboration is important. The value added by collaboration can be seen in six areas:

1) An effective handling of unexpected events.
2) Easy detection of missing schedule information
3) Added flexibility due to possible increased resource sharing
4) Schedule weaknesses can be examined through what if simulation
5) Changes are effectively communicated to the schedule executors
6) Firmer anchoring of the schedule in the organization

Acknowledgement

We would like to thank the Research Council of Norway for their contribution through the Petromaks project "Decision-Support Tool for Real-time Reservoir and Production Management" (162165/210).

References

K. R. Baker. "Introduction to Sequencing and Scheduling." John Wiley and Sons, 1974.

P. Baptiste, C. Le Pape and W. Nuijten. "Constraint-Based Scheduling, Applying Constraint Programming to Scheduling Problems." Kluwer Academic Pblishers, 2001.

M. R. Garey and D. S. Johnson. "Computers and Intractability, A Guide to the Theory of NP-Completeness." W. H. Freeman and Company, 1979.

G. Hasle, R.C. Haut, B. S. Johansen and T. S. Ølberg, "Well Activity Scheduling – An Application of Constraint Reasoning." Proc. of Practial Applications of Constraint Technology (PACT97), London, UK, 1997.

B.Ø. Holst and Nystad. "Oil & Gas offshore/onshore Integrated Operations – introducing the Brage 2010+ project." Joint 8th Annual IEEE Conference on Human Factors and Power Plants, 2007.

G. Horton and T. Dedigama. "Drilling and Petroleum Engineering Program and Project Management." SPE 104062, presented at the 2006 SPE Asia Pacific Oil & Gas Conference and Exhibition held in Adelaide, Australia, Sept. 11-13, 2006.

ILOG. "ILOG Optimization Documentation." 2007.

M. Irgens and W. L. Lavenue. "Use of Advanced Optimization Techniques to Manage a Complex Drilling Schedule." SPE 110805, presented at the 2007 SPE Annual Technical Conference and Exhibition held in Anaheim, CA, USA, Nov. 11-14, 2007.

R. Kowalsky. "Algorithm = Logic + Control." Communications of the ACM, 22(7):424-436, 1979

V. Kumar. "Algorithms for Constraint Satisfaction Problems: A Survey." AI Magazine, 13(1):32-44, 1992.

C. Le Pape. "Constraint-Based Scheduling: A tutorial." http://www.math.unipd.it/~frossi/cp-school/lepape.pdf.

I. MacDougall and K. Saman. "A Constraint Management Based Planning Approach to Delivering Exploration, Development and Production Work Programmes." SPE 101117, presented at the 2006 SPE Asia Pacific Oil & Gas Conference and Exhibition held in Adelaide, Australia, Sept. 11-13, 2006.

OLF. "Potential value of Integrated Operations on the Norwegian Shelf." 2006.

OLF. November 23rd press release, 2007

J. Ouimette and K. Oran. "Implementing Chevron's i-field at the San Ardo, California, Asset." SPE 99548, presented at the SPE 2006 Intelligent Energy Conference and Exhibition held in Amsterdam, the Netherlands, April 11-13, 2006.

T. Unneland and M., Hauser. "real-Time Asset Management: From Vision to Engagement – an Operator's Experience." SPE 96390, presented at the 2005 SPE Annual Technical Conference and Exhibition held in Dallas, Texas, USA, Oct. 9-12, 2005

Appendix A: Constraint Based Scheduling

This appendix gives a brief introduction to the elements of constraint based scheduling mentioned in this paper.

Constraint Satisfaction Problem (CSP)

A CSP is defined as a finite set of variables $V=\{X_1,...,X_n\}$, a set of domains $D=\{D_1,...,D_n\}$ where the ith domain is associated with X_i, and a set of constraints defined on the domains of a subsequence of V. A constraint is satisfied if there exists an instantiation $d_i \in D_i$ for all variables related to the constraint. A solution to a CSP is an instantiation of all variables so that all constraints are satisfied.

Le Pape defines three main principles on which constraint satisfaction programming is based. Namely:

- clearly separation of problem definition and solving algorithms
- a solution strategy divided into a deductive process and a decision process
- the deductive process performed locally and incrementally

In a CSP modeling is clearly separated from the solution strategy. The distinction between the logical representation of constraints and the control of their use is in accordance with the equation Kowalski (1979) stated for logical programming: Algorithm = Logic + Control. The solution strategy is again divided into a deductive method, referred to as constraint propagation, and a decision making search procedure. Constraint propagation makes active use of the presence of constraints in the deduction of a solution, and not only as a final check for solution feasibility. However, according to Garey and Johnson (1979) a general CSP is NP complete, i.e. there exists no known algorithm for solving the problem in polynomial time, and in most cases pure deductions alone will not be able to generate a solution. By decision making the search procedure adds constraints to the problem. If a contradiction is detected the search must backtrack and undo previous decisions looking for a solution.

The constraint propagation process shall be as local and incremental as possible. The locality principle (Steele, 1980) states that each constraint is propagated independently of the existence or non – existence of other constraints. Incrementality means that new variables and constraints can be added at any time, without re-computing all the consequences of the new constraint set.

Constraint Based Scheduling (CBS)

Most scheduling problems can according to Kumar, 1992, be represented as instances of the constraint satisfaction problem. A CBS problem consists of sets of activities, resources and constraints.

Activities

The set of activities $A = \{A_1 ... A_n\}$ is the tasks that are ready for scheduling. An activity contains information about when it is possible for it to be executed. This information is stored in three decision variables, namely start time $s(A_i)$, end time $e(A_i)$ and processing time $p(A_i)$. The domains of the start and end variables are given by the activity's release r_i, deadline d_i, latest start time lst_i and earliest end time eet_i and defined as $D(s(A_i)) = [r_i, lst_i]$ and $D(e(A_i)) = [eet_i, d_i]$. Activities can be separated into non – preemptive and preemptive. The non – preemptive ones cannot be interrupted, while preemptive activities can be put on hold for other activities or lack of resource availability. The relations between the different variables are illustrated in Figure A-1.

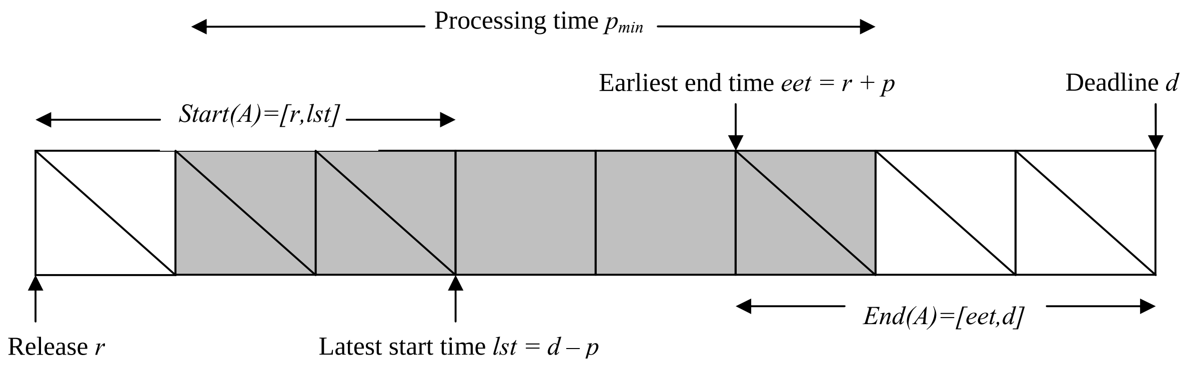

Figure A-1: Relation between time aspects of a non preemptive activity

Resources

The set of resources $R = \{R_1 ... R_n\}$ describe necessary elements that activities need in order to be executed. The activities can require, produce or consume resources. In a feasible solution of a CBS the resource requirements must at all time be within the limits of the resource capacities. Resource constraints based on the different kinds of resources ensures that these requirements are satisfied.

Unary resources
A unary resource has capacity 1 meaning that if an activity requires the resource at time *t* no other activity can use it at that time.

Discrete resources
A discrete resource has a capacity *n*. Several activities can use the resource simultaneously as long as the resource capacity is not exceeded. Discrete resources are a generalization of unary resources.

Reservoir resources
A reservoir resource has at time *t* an amount in store. Activities can in addition to require the resource during for instance execution also produce or consume the reservoir. The reservoir must at all times be within the limits of minimum and maximum capacity.

Constraints

In a CBS there are two kinds of constraints defining when the activities can be executed, namely temporal and resource constraints.

Temporal Constraints
Relational or temporal constraints are expressed as linear constraints between the time domains of activities. They can for example state that one activity must end before another can start. They are also used to state the relationship between the time domains of an activity, i.e. $s(A_i) + p(A_i) = e(A_i)$.

Resource Constraints
Resource constraints are posted between the activities in need of a resource. These constraints are the elements of the CBS ensuring that the resource capacities are not excided. Constraints related to one type of resource can be created in several ways with different propagation method related to them. Among these are time table propagation, disjunctive constraint propagation and edgefinding propagation.

Constraint propagation
An important part of solving a constraint satisfaction problem is constraint propagation. Enforcing the constraints at an early state of the solving process adding new constraints or removing values from the domains can efficiently reduce the number of feasible solutions. A solution to a CBS is globally consistent. Enforcing global consistency by constraint propagation takes exponential time and is a NP hard problem. A natural approach has been to develop local consistency techniques like for instance arc consistency. It is defined (Baptiste et al., 2001) as

> *"Given a constraint c over n variables $x_1,\ldots,x_n$ and a domain $d(x_i)$ for each variable x_i, c is said to be "arc-consistent" if and only if for any variable x_i and any value v_i in $d(x_i)$, there exist values $v_1,\ldots v_{i-1},v_{i+1},\ldots v_n$ in $d(x_1),\ldots,d(x_{i-1}),d(x_{i+1}),\ldots,d(x_n)$ such that $c(v_1,\ldots,v_n)$ holds."*

Propagation techniques are dependent on whether or not an activity is preemptive or not, and how the constraints are created. In the following three different propagation methods for constraints on a unary resource are described for non-preemptive activities (Baptist et al., 2001).

Time table constraint
A time table is an explicit data structure that maintains information about resource availability over time. The propagation consists primarily of maintaining arc consistency on

$$\sum_i X(A_i,t) \le 1 \qquad \text{(Eq. A-1)}$$

where $X(A_i,t)$ is an activity dependent variable equal 1 if and only if the activity is executed at time *t.* This lead to the following updates of activity domains and timetable.

$$\begin{aligned} & eet_i > lst_i \Rightarrow \forall t \in [lst_i, eet_i) : X(A_i,t) := 1 \\ & [X(A_i,t) = 0] \wedge [t < eet_i] \Rightarrow [s(A_i) > t] \\ & [X(A_i,t) = 0] \wedge [lst_i < t] \Rightarrow [e(A_i) \le t] \end{aligned} \qquad \text{(Eq. A-2)}$$

It is one of the most popular propagation techniques for discrete resources today. This is probably due to its fairly low complexity and simple implementation. Unfortunately the time intervals must be sufficiently small, so that $eet_i > lst_i$, before it starts to propagate.

Disjunctive constraints

A disjunctive constraint is a relational constraint between two activities that requires the same resource. If *n* activities require a resource the resource constraint can be implemented as *n(n – 1)/2* disjunctive constraints. In most cases the propagation consists of maintaining arc-consistency on the formula

$$\left[e(A_i) \le s(A_j)\right] \vee \left[e(A_j) \le s(A_i)\right] \quad \text{(Eq. A-3)}$$

These constraints may propagate more than the time table constraint, but the number of constraints increases rapidly with *n*.

Edge Finding constraints

Given a chosen set of activities **Ω** edge finding tries to determine whether or not an activity $A_i \in \Omega$ must be scheduled first or last in **Ω**. If new orderings are detected new time bounds can be deduced. The deductions are made according to the following formulas

$$\begin{aligned}
&\forall\Omega, \forall A_i \notin \Omega, \left[d_{\Omega\cup\{A_i\}} - r_\Omega < p_\Omega + p_i\right] \Rightarrow \left[A_i \prec \Omega\right] \\
&\forall\Omega, \forall A_i \notin \Omega, \left[d_\Omega - r_{\Omega\cup\{A_i\}} < p_\Omega + p_i\right] \Rightarrow \left[A_i \succ \Omega\right] \\
&\forall\Omega, \forall A_i \notin \Omega, \left[A_i \prec \Omega\right] \Rightarrow \left[e(A_i) \le \min_{\emptyset \ne \Omega' \subseteq \Omega}\left(d_{\Omega'} - p_{\Omega'}\right)\right] \\
&\forall\Omega, \forall A_i \notin \Omega, \left[A_i \succ \Omega\right] \Rightarrow \left[s(A_i) \ge \max_{\emptyset \ne \Omega' \subseteq \Omega}\left(r_{\Omega'} - p_{\Omega'}\right)\right]
\end{aligned} \quad \text{(Eq. A-4)}$$

where r_Ω, d_Ω and p_Ω denotes the earliest start time, latest end time and sum of the minimal processing times of the activities in **Ω**, respectively.

Search

The solution of a constraint satisfaction problem is based on constraint propagation and search. Constraint propagation is a purely deductive process where all certain information is taken into consideration either by reducing a decision variables domain or by posting new constraints. A scheduling problem is seldom unambiguous and constraint propagation is therefore not sufficient to find a solution. The solution generation must be helped by a decision making process, i.e. search. A search creates a search tree to traverse as the search proceeds. After each decision making, which can be viewed as adding constraints, the propagation is performed to prune the search tree as much as possible, hence reducing the number of solutions at the earliest possible stage. If a contradiction is detected i.e. a situation occurs where constraints are not satisfied, the search must backtrack and try other decisions.