

AGILE PROJECT SCHEDULING IN SOFTWARE DEVELOPMENT

Eric Newby[†]

Study Group Participants

Eric Newby, Byron Jacobs[†], Raymond Phillips[†],
Dario Fanucchi[†], Asha Taylor[†], Lady Kokela[†],
Jesal Kika[†], Nadine Padiyachi[†]

Industry Representatives

Aderemi Adewumi[‡], Brendon McDonald[‡]

Abstract

Agile project management is replacing traditional project management in many software development projects. The bulk of the project scheduling literature was developed to be applied within a traditional project management framework. In this report we develop a project scheduling procedure more appropriate to an agile project management environment. Specifically we develop a stochastic, proactive, preemptive project scheduling procedure. The structure of the procedure is such that a reactive component can be added to the procedure, if required. The scheduling procedure presented here is a modification of an existing stochastic, proactive scheduling procedure due to Van de Vonder et al. (2008). The modification allows activities to be preempted.

1 Introduction

Project scheduling is the problem of finding the optimal way to schedule a number of different activities which have durations, resource requirements and

[†]School of Computational and Applied Mathematics, University of the Witwatersrand, Wits 2050, South Africa (Eric.Newby@students.wits.ac.za).

[‡]School of Mathematics, Statistics and Computer Science, University of KwaZulu-Natal, Private Bag X01, Scottsville, Pietermaritzburg 3209 (Brendclyde@gmail.com).

are subject to given precedence relations [5]. Project scheduling is one component of the multidisciplinary field of project management. Project managers try to ensure that projects are completed on time and within the project budget through careful planning, scheduling and control of project activities [5]. In the last decade a new project management philosophy has been developed; agile project management. Agile project management is based on a number of principles, the full details of which are not necessary for this report. The interested reader is referred to [6, 7] and the references therein. For this report it suffices to know that agile project management is more flexible and adaptable than traditional project management and that incremental releases of products are sometimes required [6]. Many software development companies are starting to adopt an agile approach to project management. The study group was asked to develop a project scheduling procedure that could be used in an agile project management environment in software development.

More specifically the study group was asked to develop a stochastic, proactive/reactive, preemptive scheduling procedure. Stochastic project scheduling involves the use of stochastic rather than deterministic values for the project activity durations, this allows us to take into account the uncertainty in the estimation of activity durations which is almost always present in real world situations. Preemptive project scheduling allows activities to be interrupted before they are completed and then finished at a later time during the project execution. Proactive scheduling involves the creation of a project schedule which is protected against variations in the activity durations by the insertion of time buffers after certain activities in the schedule. Reactive scheduling involves the development of a procedure to be used during project execution to adjust the schedule if it is disrupted by delays in the completion time of some of the activities. In addition it was required that the procedure should be applicable in the case where limited resources are available; this is known as resource constrained project scheduling. Extensive further information on these concepts as well as a good general overview of project scheduling can be found in Demeulemeester and Herroelen [5].

A scheduling procedure with the required properties is developed by adapting an existing stochastic, proactive scheduling procedure [11] in such a way that the modified approach can handle preemption. A specific reactive procedure has not been developed, rather the proposed method is developed in such a way that a reactive component can easily be added to the scheduling procedure. Specifically, our procedure generates a predicted project schedule which can be used to determine when the disruptions in the schedule have become large enough for the reactive procedure to be applied.

The rest of this report is organised as follows. In Section 2 we discuss project scheduling procedures that take stochastic activity durations into account. In Section 3 we discuss the inclusion of preemption into a stochastic project scheduling procedure. In Section 4 we discuss the validity of some of

the assumptions made in the report as well as possible ways of making our procedure more agile. Concluding remarks are made in Section 5.

2 Stochastic Proactive Project Scheduling

We represent our project network $G(N, A)$ as an activity-on-node graph with dummy start and end activities. N is the set of nodes, the nodes represent the different activities in the project. A is the set of edges and represents the precedence relations between the activities. We assume that project activity i has a known stochastic duration distribution \mathbf{d}_i . Each activity requires a fixed amount of one or more finite renewable resources during each time period of its duration. Since we are scheduling with stochastic durations and resource constraints this is known as the stochastic resource constrained project scheduling problem (SRCPSP). We want to solve the SRCPSP by finding a proactive schedule in such a way that we leave room for a reactive procedure to be added to our method. Now most of the work that has been done on solving the SRCPSP involves minimising the expected project makespan. Instead of determining a deterministic schedule the SRCPSP is solved by finding a scheduling policy. The scheduling policy is used during the execution of the project which is now treated as a multi-stage decision process. At each decision point during project execution the scheduling policy is used to determine which activities should be started next [9, 10]. The main problem with this solution method, in the context of the problem considered in this report, is that it does not allow the creation of a predictive baseline schedule. This will make it very difficult to add a reactive component to the model as we will not be able to determine when enough has gone wrong during project execution for the application of the reactive procedure to become necessary.

Recently a new approach to solving the SRCPSP has been developed by Van de Vonder et al. [11] and Deblaere et al. [4]. Instead of minimising the expected makespan they minimise a stability objective function given by

$$\sum_i w_i E(|\mathbf{s}_i - s_i|), \quad (1)$$

where w_i is a weight assigned to the i th activity, s_i is the predicted starting time of the i th activity and \mathbf{s}_i is the realised starting time of activity i . The weights w_i are a measure of the cost of a delay in the completion time of the i th activity. Suggested methods for assigning these weights are given in section 4. In addition to the usual precedence and resource constraints (1) is also subject to a due date constraint which forces s_n to be less than or equal to some deterministic due date δ_n . This due date is generally set to 1.3 times the minimum makespan of the deterministic version of the project. The deterministic version of the project is the scheduling problem obtained

by taking the means of \mathbf{d}_i as deterministic durations. Hopefully, minimising (1) creates a project schedule which is unlikely to deviate from its predicted completion time and in which individual activities which would be expensive to delay are also unlikely to deviate from their predicted starting times.

The methods described in Deblaere et al. [4] and Van de Vonder et al. [11] minimise (1) by producing both a scheduling policy and a proactive predicted baseline schedule. The fact that a predicted baseline schedule can be generated makes it possible to add a reactive component to the procedure. We recommend using the STC (starting time criticality) algorithm described in Van de Vonder et al. [11] to solve the problem. A brief summary of this algorithm, all the details of which are taken from Van de Vonder et al. [11], is now given. The algorithm involves two separate stages; in the first stage an initial resource feasible schedule is found by solving the deterministic version of the project. One of the most efficient methods currently available for solving this problem is the combined crossover algorithm developed by Debels and Vanhoucke[3]. In the second stage of the procedure buffers are inserted into the schedule while keeping it resource feasible and ensuring that it satisfies the due date constraint. When inserting the buffers two factors are taken into account. Firstly, the standard deviation of an activity as well as the standard deviation of all activities preceding that activity in the schedule. Secondly, the weight of an activity as well as the weights of all activities preceding and succeeding that activity in the schedule. The reasons for considering these factors during buffer insertion should be obvious. Some measure is needed to quantify these two factors; the measure used in this algorithm is the starting time criticality (STC). The STC of activity i is defined as

$$\text{STC}_i = w_i P(\mathbf{s}_i > s_i), \quad (2)$$

where $P(\mathbf{s}_i > s_i)$ denotes the probability that activity i cannot be started at its scheduled starting time. Now before implementing the algorithm a resource flow network $G'(N, R)$ on the initial deterministic schedule needs to be determined. The resource flow network is a graph with the same nodes N as the project network $G(N, A)$ but in which the edges represent possible resource flow rather than precedence relations. The resource flow network is not unique and the network used in this case is the one found using the algorithm given in Artigues et al. [1]. The STC algorithm inserts buffers into the initial schedule using the resource flow network to ensure that the buffered schedule remains resource feasible. Details of the STC algorithm are given in Algorithm 1.

A method to calculate the probabilities $P(\mathbf{s}_i > s_i)$ is still required. The probabilities cannot be calculated directly so some approximation is required. Define $k(i, j)$ as the event that activity j disturbs the starting time of activity i . The probability of $k(i, j)$ occurring is given by

$$P(k(i, j)) = P(\mathbf{s}_j + \mathbf{d}_j + \text{LPL}(i, j) > s_i), \quad (3)$$

Algorithm 1 The STC algorithm

```

stop = 0
while stop = 0 do
  Calculate all  $STC_i$ 
   $Obj = \sum_i STC_i$ 
  Sort activities by decreasing  $STC_i$ 
  while no improved schedule found do
    Take the next activity  $j$  from the sorted list
    if  $STC_j = 0$  then
      stop = 1
    else
      Increase the starting time of activity  $j$  by one unit
      Increase the starting time of the direct and transitive successors
        of activity  $j$  in  $G$  and  $G'$  by one time unit
      Calculate the new values of  $STC_i$ 
       $ObjNew = \sum_i STC_i$ 
      if  $(ObjNew < Obj)$  and  $(s_n \leq \delta_n)$  then
        Improved schedule found
      else
        Restore the previous schedule
      end if
    end if
  end while
end while

```

where $LPL(i, j)$ is the sum of the expected durations of all the activities on the longest path between activity j and activity i in the graph $G(N, A \cup R)$. $P(\mathbf{s}_i > s_i)$ is now given by

$$P(\mathbf{s}_i > s_i) = P\left(\bigcup_{i \in T(A \cup R)} k(i, j)\right), \quad (4)$$

where $T(A \cup R)$ is defined as the set of all direct and transitive predecessors of activity i in G and G' . Two assumptions are made that allow $P(\mathbf{s}_i > s_i)$ to be approximated. Firstly, it is assumed that activity j starts at its predicted starting time when we calculate $P(k(i, j))$. It is also assumed that only one activity at a time delays the starting time of activity i . The second assumption allows (4) to be rewritten as follows

$$P(\mathbf{s}_i > s_i) = \sum_{i \in T(A \cup R)} P(k(i, j)), \quad (5)$$

substituting in (3) gives

$$P(\mathbf{s}_i > s_i) = \sum_{i \in T(A \cup R)} P(\mathbf{s}_j + \mathbf{d}_j + LPL(i, j) > s_i). \quad (6)$$

From the first assumption it is clear that $\mathbf{s}_j = s_j$ so the following approximation for $P(\mathbf{s}_i > s_i)$ is obtained

$$P(\mathbf{s}_i > s_i) = \sum_{i \in T(A \cup R)} P(\mathbf{d}_j > s_i - s_j - LPL(i, j)). \quad (7)$$

The values of s_i , s_j , $LPL(i, j)$ and the distribution of \mathbf{d}_j are all known or easily calculated so (7) can be used to calculate the value of STC_i for every activity i .

This completes the description of the STC algorithm; further details, computational results and a number of alternative algorithms can be found in Deblaere et al. [4] and Van de Vonder et al.[11].

3 Preemption in a Stochastic Environment

Allowing preemption in a deterministic environment can be done in a trivial way by splitting each node into d separate nodes with duration 1 where d is the duration of the original node. This results in a new project network with the same form as the original network so the methods developed to solve non-preemptive RCPSP can be applied to the new preemptive network. We want to find a method to modify a stochastic project graph in such a way that it

allows preemption in the project while still retaining the same general form. We will then be able to apply the STC algorithm described in section 2 to our modified network.

In this report we shall only consider the case where each activity can be interrupted at most once. Hopefully the ideas developed here can be generalised to allow an arbitrary number of preemptions. To achieve this we need some procedure that can be used to split one node with a known duration distribution into two separate nodes which each have their own known duration distributions. Let the original node represent activity j and let its stochastic duration be given by d_j . Label the two new nodes θ and ϕ for the first and second nodes respectively and let their stochastic durations be given by d_θ and d_ϕ . Clearly we must have $d_j = d_\theta + d_\phi$. Now let $P_j(t)$ be the probability that activity j is completed at time t , let $P_\theta(t)$ be the probability that activity θ is completed at time t and let $P_\phi(t)$ be the probability that activity ϕ is completed at time t . We need to calculate $P_\theta(t)$ and $P_\phi(t)$. It is difficult to find these quantities in a meaningful way using only the information given above. Accordingly, we add additional structure to the problem by defining the interruption probability $P_I(t)$ as the probability that activity j is interrupted at time t .

Using this information the duration distributions of activities θ and ϕ can be found using probability diagrams. The duration distribution for activity θ is given by

$$P_\theta(t) = \left(\tilde{P}_j(t) + \tilde{P}_I(t) - \tilde{P}_I(t)\tilde{P}_j(t) \right) \Xi(t),$$

where $\tilde{P}_j(t)$, $\tilde{P}_I(t)$ and $\Xi(t)$ are defined as follows

$$\begin{aligned} \tilde{P}_j(t) &= \frac{P_j(t)}{\sum_{m=t}^n P_j(m)}, \\ \tilde{P}_I(t) &= \frac{P_I(t)}{1 - \sum_{m=1}^{t-1} P_I(m)}, \\ \Xi(t) &= \prod_{m=0}^{t-1} \left[\left(1 - \tilde{P}_j(m) \right) \left(1 - \tilde{P}_I(m) \right) \right], \end{aligned}$$

where n is the number of time periods over which the distributions are defined. The duration distribution for activity ϕ is given by

$$P_\phi(t) = \begin{cases} \sum_{k=1}^n \tilde{P}_j(k)\Xi(k), & t = 0, \\ \sum_{k=1}^{n-t} \left\{ \Xi(k) \left(1 - \tilde{P}_j(k) \right) \tilde{P}_I(k)\Upsilon(k,t)\tilde{P}_j(k+t) \right\}, & t > 0, \end{cases}$$

where $\Upsilon(k, t)$ is defined as

$$\Upsilon(k, t) = \prod_{r=0}^{t-1} \left[1 - \tilde{P}_j(r+k) \right].$$

Using the distributions, P_θ and P_ϕ , we can modify our original graph in such a way that it allows pre-emption while retaining the same form as the original graph. Combining this with the stochastic, proactive procedure described in Section 2 we obtain a stochastic, proactive, preemptive project scheduling procedure, as required.

4 Comments on Assumptions and Agility

It has been assumed that the project managers have access to the following quantities:

1. The duration distributions of the activities.
2. The weights w_i used to define (1).
3. The interruption probability.

We now give a brief discussion of possible scheduling procedures that can be applied if this information is not available. Firstly, to perform any project scheduling we must have some information about the durations of the activities. If duration distributions are not available we must either have access to fuzzy durations or deterministic durations. Project scheduling using deterministic durations is well developed and extensive further information can be found in Demeulemeester and Herroelen [5]. Projects with fuzzy durations can be solved by using a fuzzy variant of the critical path method [8, 13] or by using heuristics to minimise the fuzzy makespan of the project [2, 12].

The activity weights w_i may be difficult to assign. One possibility is to make the weights proportional to the real world cost of delaying the activities; for example delaying the start of an activity might require storage space for materials to be rented for a longer period. The activities on the critical path of the deterministic version of the project might also be assigned a higher weight than the other activities. The dummy end activity is generally assigned a value of $\lfloor 10w_{\text{avg}} \rfloor$, where w_{avg} is the average weight of the non-dummy activities [11]. If the weights cannot be assigned the problem can be solved by generating a scheduling policy [9, 10]. However, as noted in Section 2, this will make the addition of a reactive component to the procedure much more difficult.

There are currently no methods in the literature for including preemption in a project with stochastic activity durations. Accordingly, if the interruption

probability is not available there are three options available to the project manager. Firstly, preemption can be ignored and a stochastic, proactive schedule can be found using the method in Section 2. Secondly, the stochastic nature of the activity durations could be ignored, allowing the deterministic preemptive methods described in [5] to be applied. Another option is to develop a new method of including preemption in a stochastic environment. However, it might be difficult to achieve this without changing the structure of the project network.

We now give a brief discussion of possible modifications that could be made to the scheduling procedure developed in this report to make the procedure more applicable when scheduling agile projects. Firstly, it should be possible to allow activities to be cancelled as well as allowing for preemption. This could be done by defining a cancellation probability analogous to the interruption probability defined in Section 3. It might also be necessary to include stochastic resource availabilities in the schedule. In some cases this can be done by varying the duration distribution. If this is not the case it will be a non-trivial matter to include stochastic resource availabilities in our model, this is an area of future research. Agile software development often requires the delivery of portions of the software to the client during project execution. This could be included in our procedure by inserting a dummy node in the project network as a direct successor to all of the activities required to make a portion of the software available. These dummy activities could be given a relatively high weight to ensure that the software portions are delivered close to their predicted completion times.

5 Conclusion

We have developed a stochastic, proactive, preemptive project scheduling procedure. This procedure can be used to assist project managers in constructing agile schedules for software development. The scheduling method is structured in such a way that it should be easy to add a reactive procedure to the method. In addition we suggest a number of ways in which our procedure could be modified to make it more consistent with agile scheduling principles. We also give a number of alternative scheduling procedures that can be applied if there is not enough information available to allow our method to be applied.

References

- [1] Artigues, C., Michelon, P. and Reusser, S. Insertion techniques for static and dynamic resource-constrained project scheduling. *European Journal of Operational Research*, **149**, (2003), 249-267.

- [2] Bhaskar, T., Pal, M.N. and Pal, A.K. A heuristic method for R-CPSP with fuzzy activity times. *European Journal of Operational Research*, **208**, (2011), 57-66.
- [3] Debels, D. and Vanhoucke, M. Future research avenues for resource-constrained project scheduling: search space restriction or neighbourhood search extension. In: *Proceedings of the 10th International Workshop on Project Management and Scheduling*, 2006, pp 110-113.
- [4] Deblaere, F., Demeulemeester, E.L. and Herroelen, W.S. Proactive policies for the stochastic resource-constrained project scheduling problem. *European Journal of Operational Research*, **214**, (2011), 308-316.
- [5] Demeulemeester, E.L. and Herroelen, W.S. *Project Scheduling: A Research Handbook*. Kluwer Academic Publishers, first edition, 2002.
- [6] Highsmith, J. *Agile Project Management: Creating Innovative Products*. Addison-Wesley, first edition, 2004.
- [7] Lee, J. and Hur, S.J. Agile approach to manage projects in ubiquitous multi-project environment. In: *Proceedings of the 5th International Conference on Ubiquitous Information Technologies and Applications*, 2010, pp 1-5.
- [8] Long, L.D. and Ohsato, A. Fuzzy critical chain method for project scheduling under resource constraints and uncertainty. *International Journal of Project Management*, **26**, (2008), 688-698.
- [9] Möhring, R.H., Radermacher, F.J. and Weiss, G. Stochastic scheduling problems I: General strategies. *Zeitschrift für Operations Research*, **28**, (1984), 193-260.
- [10] Stork, F. *Stochastic Resource-Constrained Project Scheduling*. PhD thesis, Technical University of Berlin, School of Mathematics and Natural Sciences, 2001.
- [11] Van de Vonder, S., Demeulemeester, E.L. and Herroelen, W.S. Proactive heuristic procedures for robust project scheduling: An experimental analysis. *European Journal of Operational Research*, **189**, (2008), 723-733.
- [12] Wang, H., Lin, D. and Li, M. A genetic algorithm for solving fuzzy resource-constrained scheduling. In: *Advances in Natural Computation, Lecture Notes in Computer Science*, Volume 3612, (2005), pp 171-180. Editors Wang, K., Chen, K. and Ong, Y.S., Springer.
- [13] Zhao, Z.-Y., You, W.-Y. and Lv, Q.-L. Applications of fuzzy critical chain method in project scheduling. In: *Fourth International Conference on Natural Computation*, Volume 6, 2008, pp 473-477.