Project Scheduling in Software Development

Eric Newby, Raymond Phillips, Dario Fanucchi, Byron Jacobs, Asha Tailor, Lady Kokela, Jesal Kika, Nadine Padiyachi

University of the Witwatersrand

January 13, 2012
Overview of the problem

- Schedule a project with stochastic activity durations and preemption. The schedule should be proactive.
- Design a heuristic project scheduling algorithm.
- Create mocked project with 20 activities.
- Deterministic methods and reactive scheduling were not considered due to time constraints.
Review of RCPSP

(a) RCPSP graph

(b) RCPSP schedule
Reformulating the problem

- Including proactivity and stochastic duration.
- Need to generate a predicted baseline schedule to leave room for reactive scheduling.
- Normal stochastic methods generate a scheduling policy.
- Baseline schedule needs to be proactive.
- Instead of minimizing the makespan, minimize

\[ \sum_j w_j E(|s_j - s_j|) \]

- Include due date constraint.
Reformulating the problem

- To include preemption, need to be able to leave an activity and return later.
- Achieve this without changing the form of the graph.
- Only allow one preemption per activity.
Reformulating the problem

- Assume that the probability of being interrupted can be defined.
- Distribution for the duration of X can be found analytically

\[ P_X(j) = \prod_{i=1}^{j-1} (1 - P_F(i) - P_I(i))(P_F(j) + P_I(j)) \]

where

\[ P_F(i) = \frac{P_A(i)}{\sum_{k=i}^{n} P_A(k)} \]

- Distribution for the duration of Y can be found by simulation.
Created project with 20 activities and 3 resource types.
The Mocked Project

Using the means as deterministic durations we get the following schedule.
The Heuristic Algorithm

- Recommend using STC algorithm (Stijin Van de Vonder et al, 2008).
- Add a due date $\delta_n$ to our problem.
- Start from an initial unbuffered schedule and iteratively create intermediate schedules by adding a time buffer in front of that activity that needs it the most.
- Which activity ”needs it most” is determined using a quantity, called stc, which depends on both the activities weight and its duration distribution, $stc_j = w_j \times P(s_j > s_j)$.
- Once the buffer is inserted the schedule is updated.
- To ensure resource feasiblity we construct a resource flow network for our initial schedule.
To ensure resource feasibility we construct a resource flow network for our initial schedule.
Resource Flow Network

We obtain the following flow network for our mocked project.
The Heuristic Algorithm

- When we add a buffer of one to an activity we also add to the starting times of all the direct and transitive successors of the activity.
- Keeps the schedule resource and precedence feasible.
- If this new schedule is an improvement and obeys the due date constraint we update the schedule.
- We repeat this process until we encounter any $stc_j = 0$.
- The current best schedule is an approximation of the optimal schedule.
The following graph shows the starting times obtained for our mocked project using the heuristic algorithm.
We have assumed we know the following:

1. Stochastic activity duration
2. Activity weights
3. Interruption probability

If we don’t have stochastic activity durations we must either have fuzzy durations or deterministic durations.

Deterministic proactive preemption is well developed e.g. critical path method.

Fuzzy projects can be solved by converting to a deterministic schedule or using GA to minimise the fuzzy makespan.

The activity weights $w_j$ may be difficult to assign.
Applicability to Industry

- Use weights proportional to real world costs of delay and/or weight activities on critical path more heavily.
- If $w_i$ not available may have to use normal SRCPSP, makes reactive scheduling difficult.
- There should be a number of ways to quantify preemption in a stochastic environment.
- Could be difficult to keep a graph of the required form.
- Currently no literature on stochastic preemption, no readily available alternatives.
There are a number of ways the schedule algorithm could be made more agile.

Define a cancelling probability for each node to allow for cancelling of activities.

There may be difficulties combining interruption and cancellation probabilities.

Partial completion can be included trivially using intermediate dummy nodes.

Variable resources can be modelled with variable activity durations. If this is difficult to do the model will have to be expanded.

Large changes in the schedule during execution should be included in a reactive procedure.
We have developed a mathematical formulation for a stochastic, preemptive resource constrained network.

We propose a heuristic method to create a proactive schedule for projects formulated in this way.

The heuristic was implemented and used to solve a mocked project with twenty activities.

Once a reactive component has been added this is sufficiently agile.

Given various alternative methods that could be applied if information is insufficient to apply our model.

Discussed various ways in which the model could be made more agile if desired.