AdM III: Scheduling and Project Planning

Steffen Suerbier
Benjamin Labonte

October 25, 2012

Abstract

Lecture notes taken during the 2012/2013 lecture "Scheduling and Project Planning" by Prof. Dr. Rolf H. Möhring.

Contents

1 Projects and Partial Orders

What makes a project?

- activities or jobs $j \in V = \{1, 2, \cdots, n\}$
- job data
  - processing time/duration (deterministic/random)
  - resource consumption
  - processing cost
- project data
  - available resources
  - limited budget
- project rules for carrying out the project
  - temporal conditions
  - resource conditions
  - setups

Precedence constraints, i.e. job j has to wait for i define a partial ordering on V. We write $i < j$ if j must wait for i.

Definition 1.1 (Partial Ordering). *A (strict) partial order is a binary relation $<$ over a set V which is*

(i) assymmetric: $i < j \Rightarrow j \not< i$

(ii) transitive: $i < j,\ j < k \Rightarrow i < k$

Definition 1.2 (Transitive Reduction of $<$, Transitive closure). *The transitive reduction of the finite partial order $(V, <)$ is the digraph $G = (V, E)$ with the edges*

$$E = \{(i, j) \mid i, j \in V,\ i < j,\ \nexists k : i < k < j\}.$$

In the context of scheduling this is also called activity-on-node diagram. For any graph $G = (V, E)$ the edge set defined by

$$E^{trans} = \{(i, j) \mid \exists \textit{finite sequence } i = i_0, \cdots, i_k = j : (i_j, i_{j+1}) \in E\}$$

denotes the transitive closure of E.

Example 1.3 (Bridge Construction Project).

Definition 1.4.

$$\begin{aligned} Pred_G(i) &= \{j \in V \mid j < i\} && \textit{set of predecessors} \\ Suc_G(i) &= \{j \in V \mid j > i\} && \textit{set of successors} \\ ImPred_G(i) &= \{j \in V \mid (j,i) \in E\} && \textit{set of predecessors} \\ ImSuc_G(i) &= \{j \in V \mid (i,j) \in E\} && \textit{set of predecessors} \end{aligned}$$

We call an element maximal if it has no successors, minimal if it has no predecessors. If there is only one maximal (minimal) element it is called greatest (smallest). i, j *are comparable* ($i \sim_G j$) *if* $i < j$ *or* $j > i$ *otherwise they are incomparable* ($i \parallel_G j$)*.*

Lemma 1.5. *Let* $(V, <)$ *be a finite partial order and* (V, E) *be its transitive reduction. Then* E *is the smallest (under* $\leq$*) binary relation, s.t.* $E^{trans} = <$.

Proof. Show that $E^{trans} = <$. Let $i < j$ and $(i, j) \notin E$. Then there is a $k \in V$ s.t. $i < k < j$. As $<$ is acyclic and finite, by iterating, we obtain a finite sequence $i = i_0 \to \cdots \to j$. □

Exercise 0. Show that E^{trans} exists.

Remark 1.6. Lemma 1.5 does not hold for relations with cycles or infinite ground sets.

Exercise 1. Formulate an algorithm for constructing the transitive closure of a digraph.

Exercise 2. Formulate an algorithm for constructing the transitive reduction of a partial order or a directed acyclic graph.

Exercise 3. Prove an $\mathcal{O}(n^3)$ upper bound on the running times of the algorithms.