A Stochastic Scheduling Model for Software Projects

Frank Padberg
Fakultät für Informatik
Universität Karlsruhe
Germany
Characteristics of Software Projects

- activity durations not precisely known
- hard to track the progress
- strong feedback between activities
- frequent rework and delays

*need a scheduling model tailored to software projects*
Solution Approach

- model software project as Markov decision process
- apply stochastic dynamic programming
- use simulation to cut down computing times
Model Properties from OR Perspective

- constrained, non-identical, renewable resources
- preemptive scheduling (interrupts, reassignments)
- stochastic activity times
- activity time depends on feedback
- activity time depends on scheduling strategy
Feedback and Rework

- feedback causes interrupts and leads to rework
- impact depends on component coupling

© Dr. Frank Padberg 2002
Phases

- time is discrete: time slices, *phases*
- phase ends when component gets finished
- phase ends when design change occurs
- scheduling occurs only at end of phases
State

- progress vector: net development time for each component

- rework vector: rework time still left for each component

- countdown: time left until deadline expires

- example: ((2, 4, 3); (0, 1, 2); 10)
Action

- **action vector**: for each team the number of the component to work on

- **example**: \((1, 3)\)

- components not showing in the action vector are stopped (or finished)

- team is stopped if entry is -1
Strategy

- action $a$ may depend on current state $\zeta$
- strategy $\pi$ specifies for each state which action to take:
  \[ a = \pi(\zeta) \]
- actions must be admissible
Partial Sample Project Tree

\[
\begin{align*}
\alpha &: (1, 3) \\
\eta &: ((\infty, 4, 5); (0, 1, 0); 6) \\
\zeta &: ((2, 4, 3); (0, 1, 2); 10)
\end{align*}
\]

next state

scheduling

current state

© Dr. Frank Padberg 2002
Base Probabilities

$$P(D^i_k(t))$$ – team $i$ finishes after a net time of $t$ slices when working on component $k$

- example:

- measure net development times in various projects
Dependency Degrees

\[ \alpha ( K, X ) \] – probability that set \( X \) of components must be changed if design problems occur in set \( K \) of components

- example: \( \alpha ( \{ 2 \}, \{ 1, 2 \} ) = 0.25 \)

- strength of coupling between components
Stochastic Optimization Problem

- find a scheduling strategy which has minimal expected project cost in the Markov decision model for software projects
- optimal strategy depends on the input data
Thank You!