DM204, 2010
SCHEDULING, TIMETABLING AND ROUTING

Lecture 22
Resource-Constrained Project Scheduling Model

Marco Chiarandini

Department of Mathematics & Computer Science
University of Southern Denmark

Outline

Course Overview

- ✔ Problem Introduction
  - ✔ Scheduling classification
  - ✔ Scheduling complexity
  - ✔ RCPSP
- ✔ General Methods
  - ✔ Integer Programming
  - ✔ Constraint Programming
  - ✔ Heuristics
  - ✔ Dynamic Programming
  - ✔ Branch and Bound
- ✔ Scheduling Models
  - ✔ Single Machine
  - ✔ Parallel Machine and Flow Shop
  - ✔ Job Shop
  - Resource-Constrained Project Scheduling
- Timetabling
  - Reservations and Education
  - University Timetabling
  - Crew Scheduling
  - Public Transports
- Vechicle Routing
  - Capacited Models
  - Time Windows models
  - Rich Models

Outline

RCPS Model

Resource Constrained Project Scheduling Model

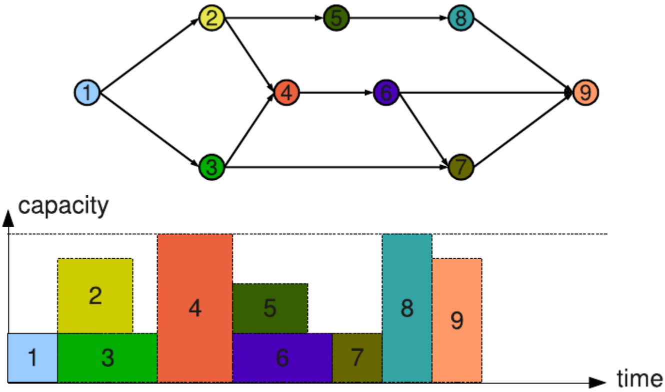
Given:

- activities (jobs) $j = 1, \ldots, n$
- renewable resources $i = 1, \ldots, m$
- amount of resources available R_i
- processing times p_j
- amount of resource used r_{ij}
- precedence constraints $j \rightarrow k$

Further generalizations

- Time dependent resource profile $R_i(t)$ given by (t_i^μ, R_i^μ) where $0 = t_i^1 < t_i^2 < \ldots < t_i^{m_i} = T$
- Multiple modes for an activity j processing time and use of resource depends on its mode m: p_{jm}, r_{jkm}.

Example



Modeling

Case 1

- A contractor has to complete n activities.
- The duration of activity j is p_j
- each activity requires a crew of size W_j.
- The activities are not subject to precedence constraints.
- The contractor has W workers at his disposal
- his objective is to complete all n activities in minimum time.

Modeling

Case 2

- Exams in a college may have different duration.
- The exams have to be held in a gym with W seats.
- The enrollment in course j is W_j and
- all W_j students have to take the exam at the same time.
- The goal is to develop a timetable that schedules all n exams in minimum time.
- Consider both the cases in which each student has to attend a single exam as well as the situation in which a student can attend more than one exam.

Mathematical Model

$$
\begin{aligned}
\text{min} \quad & \max_{j=1}^{n} \{S_j + p_j\} \\
\text{s.t.} \quad & S_j \geq S_i + p_i, \qquad j = 1, \ldots, n, \forall (i,j) \in A \\
& \sum_{j \in J(t)} r_{jk} \leq R_k, \qquad k = 1, \ldots m, t = 1 \ldots, T \\
& S_j \geq 0, \qquad j = 1, \ldots, n \\
\text{where} \quad & J(t) = \{j = 1, \ldots, n | S_j \leq t \leq S_j + p_j\}.
\end{aligned}
$$

Preprocessing: Temporal Analysis

- Precedence network must be acyclic

Preprocessing: constraint propagation

1. conjunctions $i \rightarrow j$ $S_i + p_i \leq S_j$
   [precedence constrains]

2. parallelity constraints $i \,||\, j$ $S_i + p_i \geq S_j$ and $S_j + p_j \geq S_i$
   [time windows $[r_j, d_j]$,$[r_l, d_l]$ and $p_l + p_j > \max\{d_l, d_j\} - \min\{r_l, r_j\}$]

3. disjunctions $i - j$ $S_i + p_i \leq S_j$ or $S_j + p_j \leq S_i$
   [resource constraints: $r_{jk} + r_{lk} > R_k$]

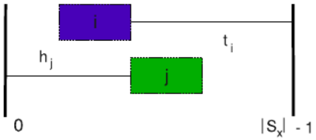
N. Strengthenings: symmetric triples, etc.

Preprocessing: Temporal Analysis

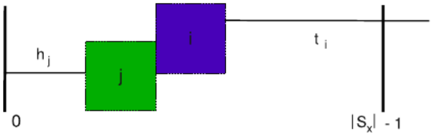
- When a schedule S_x with makespan $|S_x|$ is found, we are only interested in solutions S_x with $|S_x| = |S_x| - 1$.
- Precedence augmentation adds new precedence relations which must be satisfied for the makespan to improve.
- Heads r_j and Tails $q_j \Leftarrow$ Longest paths $\Leftarrow$ Topological ordering (deadlines d_j can be obtained as $UB - q_j$)

Let i, j be a pair of activities. A precedence relation is added between i and j if one of the following holds:

- $h_j + q_i \geq |S_x| - 1$



- $h_j + p_j + p_i + q_i > |S_x| - 1 \quad \wedge \quad \exists k = 1, \ldots, m : r_{ik} + r_{jk} > R_k$



Solutions

Task: Find a schedule indicating the starting time of each activity

- All solution methods restrict the search to feasible schedules, S, S'
- Types of schedules
  - Local left shift (LLS): $S \rightarrow S'$ with $S'_j < S_j$ and $S'_l = S_l$ for all $l \neq j$.
  - Global left shift (GLS): LLS passing through infeasible schedule
  - Semi active schedule: no LLS possible
  - Active schedule: no GLS possible
  - Non-delay schedule: no GLS and LLS possible even with preemption
- If regular objectives $\Longrightarrow$ exists an optimum which is active

Hence:

- Schedule not given by start times S_i
  - space too large $O(T^n)$
  - difficult to check feasibility
- Sequence (list, permutation) of activities $\pi = (j_1, \ldots, j_n)$
- π determines the order of activities to be passed to a schedule generation scheme

Schedule Generation Schemes

Given a sequence of activity, SGS determine the starting times of each activity

Serial schedule generation scheme (SSGS)

n stages, S_λ scheduled jobs, E_λ eligible jobs

Step 1 Select next from E_λ and schedule at earliest.

Step 2 Update E_λ and $R_k(\tau)$.
If E_λ is empty then STOP,
else go to Step 1.

Parallel schedule generation scheme (PSGS) (Time sweep)

stage λ at time t_λ

S_λ (finished activities), A_λ (activities not yet finished), E_λ (eligible activities)

Step 1 In each stage select maximal resource-feasible subset of eligible activities in E_λ and schedule it at t_λ.

Step 2 Update E_λ, A_λ and $R_k(\tau)$.
If E_λ is empty then STOP,

else move to $t_{\lambda+1} = \min \left\{ \min_{j \in A_\lambda} C_j, \min_{\substack{k=1,\ldots,r \\ i \in m_k}} t_i^\mu \right\}$

and go to Step 1.

- If constant resource, it generates non-delay schedules
- Search space of PSGS is smaller than SSGS

Possible uses:

- Forward
- Backward
- Bidirectional
- Forward-backward improvement (justification techniques)
  [V. Valls, F. Ballestín and S. Quintanilla. Justification and RCPSP: A technique that pays. EJOR, 165:375-386, 2005]

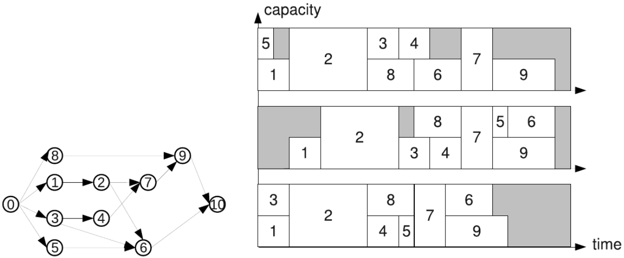


Fig. from [D. Debels, R. Leus, and M. Vanhoucke. A hybrid scatter search/electromagnetism meta-heuristic for project scheduling. EJOR, 169(2):638Â653, 2006]

Dispatching Rules

Determines the sequence of activities to pass to the schedule generation scheme

- activity based
- network based
- path based
- resource based

Static vs Dynamic

Local Search

All typical neighborhood operators can be used:

- Swap
- Interchange
- Insert

reduced to only those moves compatible with precedence constraints

Genetic Algorithms

Recombination operator:

- One point crossover
- Two point crossover
- Uniform crossover

Implementations compatible with precedence constraints