DM865 – Spring 2018
Heuristics and Approximation Algorithms

Resource Constrained Project Scheduling

Marco Chiarandini

Department of Mathematics & Computer Science
University of Southern Denmark

Outline

Outline

1. Resource Constrained Project Scheduling Model
2. Preprocessing
3. Heuristics

RCPSP

Resource Constrained Project Scheduling Model

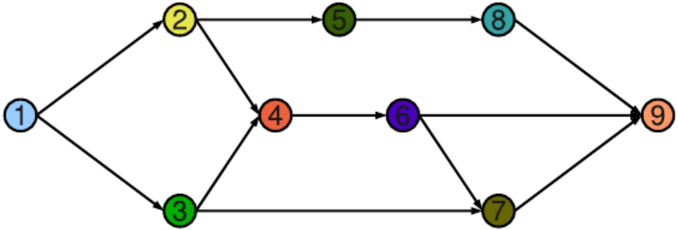
Given:

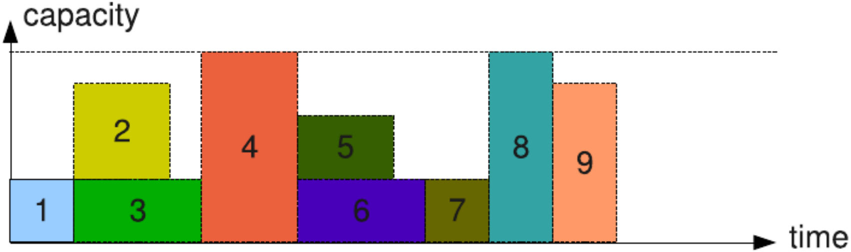
- activities (jobs) $j = 1, \ldots, n$
- renewable resources $i = 1, \ldots, m$
- amount of resources available R_i
- processing times p_j
- amount of resource used r_{ij}
- precedence constraints $j \to k$

Further generalizations

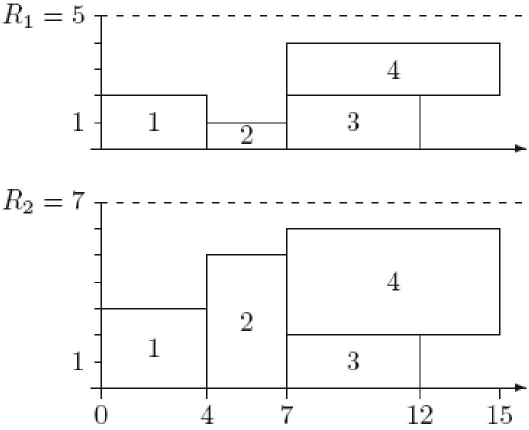
- Time dependent resource profile $R_i(t)$
  given by (t_i^μ, R_i^μ) where $0 = t_i^1 < t_i^2 < \ldots < t_i^{m_i} = T$
  Disjunctive resource, if $R_k(t) = \{0, 1\}$; cumulative resource, otherwise
- Multiple modes for an activity j
  processing time and use of resource depends on its mode m: p_{jm},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

An Example

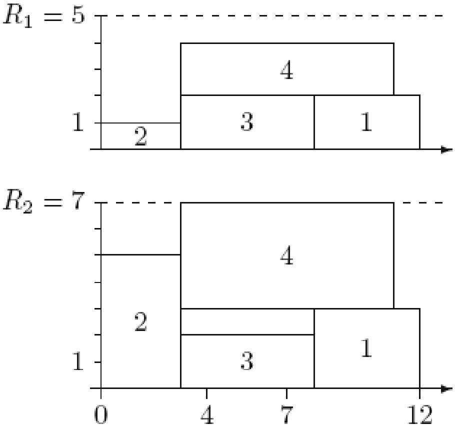




An Example



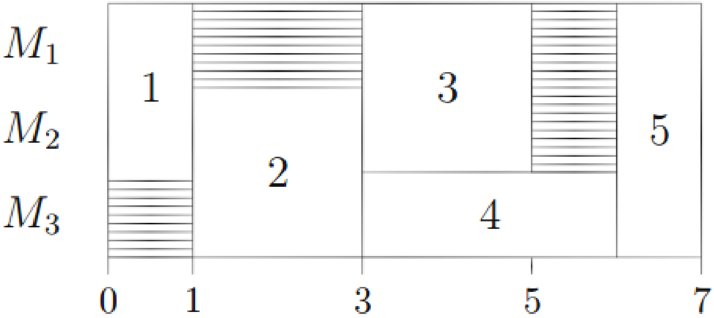
(a) A feasible schedule



(b) An optimal schedule

Multi-processor Task Scheduling

j	1	2	3	4	5
μ_j	$\{M_1, M_2\}$	$\{M_2, M_3\}$	$\{M_1, M_2\}$	$\{M_3\}$	$\{M_1, M_2, M_3\}$
p_j	1	2	2	3	1



Equivalent to a RCPSP with $r = m$ and $R_k = 1$ for $k = 1..m$

Modeling

Assignment 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.

Assignment 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.

Assignment 3

- In a basic high-school timetabling problem we are given m classes $c_1, \dots, c_m$,
- h teachers $a_1, \dots, a_h$ and
- T teaching periods $t_1, \dots, t_T$.
- Furthermore, we have lectures $i = l_1, \dots, l_n$.
- Associated with each lecture is a unique teacher and a unique class.
- A teacher a_j may be available only in certain teaching periods.
- The corresponding timetabling problem is to assign the lectures to the teaching periods such that
  - each class has at most one lecture in any time period
  - each teacher has at most one lecture in any time period,
  - each teacher has only to teach in time periods where he is available.

Assignment 4

- A set of jobs $J_1, \dots, J_g$ are to be processed by auditors $A_1, \dots, A_m$.
- Job J_l consists of n_l tasks ($l = 1, \dots, g$).
- There are precedence constraints $i_1 \rightarrow i_2$ between tasks i_1, i_2 of the same job.
- Each job J_l has a release time r_l, a due date d_l and a weight w_l.
- Each task must be processed by exactly one auditor. If task i is processed by auditor A_k, then its processing time is p_{ik}.
- Auditor A_k is available during disjoint time intervals $[s_k^\nu, l_k^\nu]$ ($\nu = 1, \dots, m$) with $l_k^\nu < s_k^\nu$ for $\nu = 1, \dots, m_k - 1$.
- Furthermore, the total working time of A_k is bounded from below by H_k^- and from above by H_k^+ with $H_k^- \leq H_k^+$ ($k = 1, \dots, m$).
- We have to find an assignment $\alpha(i)$ for each task $i = 1, \dots, n := \sum_{l=1}^{g} n_l$ to an auditor $A_{\alpha(i)}$ such that
    - each task is processed without preemption in a time window of the assigned auditor
    - the total workload of A_k is bounded by H_k^- and H_k^k for $k = 1, \dots, m$.
    - the precedence constraints are satisfied,
    - all tasks of J_l do not start before time r_l, and
    - the total weighted tardiness $\sum_{l=1}^{g} w_l T_l$ is minimized.

Mathematical Model

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

Outline

1. Resource Constrained Project Scheduling Model
2. Preprocessing
3. Heuristics

Preprocessing: Temporal Analysis

- Precedence network must be acyclic

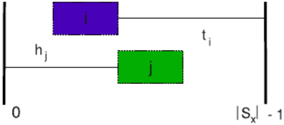
Preprocessing: constraint propagation

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: $r_{jk} + r_{lk} > R_k$]

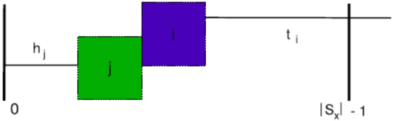
N. Strengthenings: symmetric triples, etc.

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

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



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



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

Outline

1. Resource Constrained Project Scheduling Model
2. Preprocessing
3. Heuristics

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.

```
Procedure Serial Schedule Generation Scheme
 1. Let $E_1$ be the set of all activities without predecessor;
 2. FOR $\lambda := 1$ TO $n$ DO
 3.   Choose an activity $j \in E_\lambda$;
 4.   $t := \max_{i \to j \in A} \{S_i + p_i\}$;
 5.   WHILE a resource $k$ with $r_{jk} > R_k(\tau)$ for some time
      $\tau \in \{t+1, \ldots, t+p_j\}$ exists DO
 6.     Calculate the smallest time $t_k^\mu > t$ such that $j$
        can be scheduled in the interval $[t_k^\mu, t_k^\mu + p_j[$ if
        only resource $k$ is considered and set $t := t_k^\mu$;
 7.   ENDWHILE
 8.   Schedule $j$ in the interval $[S_j, C_j[ := [t, t+p_j[$;
 9.   Update the current resource profiles by setting
      $R_k(\tau) := R_k(\tau) - r_{jk}$ for $k = 1, \ldots, r$; $\tau \in \{t+1, \ldots, t+p_j\}$;
10.   Let $E_{\lambda+1} := E_\lambda \setminus \{j\}$ and add to $E_{\lambda+1}$ all successors
      $i \notin E_\lambda$ of $j$ for which all predecessors are scheduled;
11. ENDFOR
```

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

and go to Step 1.

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

```
Procedure Parallel Schedule Generation Scheme
$\lambda := 1$; $t_1 := 0$; $A_1 := \emptyset$;
Let $E_1$ be the set of all activities $i$ without predecessor
and $r_{ik} \leq R_k(\tau)$ for $k = 1, \ldots, r$ and all $\tau \in \{1, \ldots, p_i\}$;
WHILE not all activities are scheduled DO
  WHILE $E_\lambda \neq \emptyset$ DO
    Choose an activity $j \in E_\lambda$;
    Schedule $j$ in the interval $[S_j, C_j[ := [t_\lambda, t_\lambda + p_j[$;
    Update the current resource profiles by setting
    $R_k(\tau) := R_k(\tau) - r_{jk}$ for $k = 1, \ldots, r$; $\tau \in \{t_\lambda + 1, \ldots, t_\lambda + p_j\}$;
    Add $j$ to $A_\lambda$ and update the set $E_\lambda$ by eliminating
    $j$ and all activities $i \in E_\lambda$ with $r_{ik} > R_k(\tau)$ for some
    resource $k$ and a time $\tau \in \{t_\lambda + 1, t_\lambda + p_i\}$;
  ENDWHILE
  Let $t_{\lambda+1}$ be the minimum of the smallest value $t_k^\mu > t_\lambda$
  and $\min_{i \in A_\lambda} \{S_i + p_i\}$;
  $\lambda := \lambda + 1$;
  Calculate the new sets $A_\lambda$ and $E_\lambda$;
ENDWHILE
```

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]

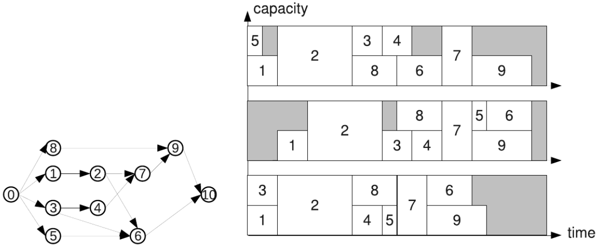


Fig. from [D. Debels, R. Leus, and M. Vanhoucke. A hybrid scatter search/electromagnetism meta-heuristic for project scheduling. EJOR, 169(2):638Â653, 2006]

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

Ant Colony

```
Ant algorithm RCPSP
REPEAT
    FOR $k := 1$ TO $m$ DO
        FOR $i := 1$ TO $n$ DO
            Choose an unscheduled eligible activity
            $j \in V$ for position $i$ with probability
            $p_{ij}^k = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum\limits_{l \in V^k} [\tau_{il}]^\alpha [\eta_{il}]^\beta}$;
        ENDFOR
    ENDFOR
    Calculate the makespans $C^k$ of the schedules
    constructed by the ants $k = 1, \ldots, m$;
    Determine the best makespan $C^* = \min\limits_{k=1}^{m}\{C^k\}$ and a
    corresponding list $L^*$;
    FOR ALL activities $j \in V$ and their corresponding
    positions $i$ in $L^*$ DO
        $\tau_{ij} := (1 - \varrho)\tau_{ij} + \varrho \frac{1}{2C^*}$;
UNTIL a stopping condition is satisfied
```