Mixed-integer linear programming for project scheduling with resource-unit related constraints

Norbert Trautmann
University of Bern, Switzerland

Scheduling Seminar, www.schedulingseminar.com
December 4, 2024

Thanks to all contributors





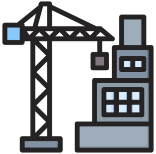




N. Ackermann
T. Bigler
M. Gnägi
N. Klein
T. Rihm
N. Saner
J. Stadler
L. Trotter
A. Zimmermann

Overview: RCPSP

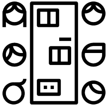
- Resource-Constrained Project Scheduling Problem (RCPSP): devise a schedule for execution of project activities such that
  - project duration (i.e., time-to-market) is minimized,
  - completion-start precedence between given pairs of activities is respected, and
  - at no time total demand of the in-progress activities exceeds the available capacity of the various resource types required for the execution of the activities
- In many applications: resource types represent pools of teams of people with specific skills or equipment units
- Well-known MILP formulations of the RCPSP
  - Discrete-time and continuous-time models
  - In general, consideration of resource-unit related constraints



All images: Flaticon.com

Overview: RCPSP variants considered here

- Novel RCPSP variants: additional resource-unit related constraints
  1. Multi-site resource-constrained project scheduling
     - Particular site must be selected for execution of each activity
     - Some resource units available only at a particular site
     - Other resource units can be moved between sites, requiring some transportation time
  2. Workload balancing in resource-constrained project scheduling
     - Foster team productivity and cohesion by balancing workload across team



- For both variants: continuous-time assignment-based MILP formulation

All images: Flaticon.com

Outline

Part I: MILP formulations of the RCPSP

Part II: Multi-site resource-constrained project scheduling

Part III: Workload balancing in resource-constrained project scheduling

Part IV: Conclusions

Outline

Part I: MILP formulations of the RCPSP

Part II: Multi-site resource-constrained project scheduling

Part III: Workload balancing in resource-constrained project scheduling

Part IV: Conclusions

ELSEVIER
Computers & Industrial Engineering
Volume 129, March 2019, Pages 346-353

Two continuous-time assignment-based models for the multi-mode resource-constrained project scheduling problem

Mario Gnägi, Tom Rihm, Adrian Zimmermann, Norbert Trautmann

Show more

Add to Mendeley Share Cite

https://doi.org/10.1016/j.cie.2019.01.033

Outline I: MILP Formulations

Modeling approaches

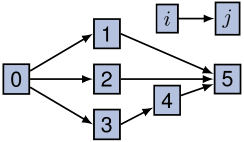
Continuous-time assignment-based MILP formulation

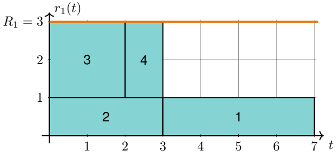
Computational results

MILP models from the literature

Illustrative example

- Project activities $1, \ldots, 4$
- Single resource type, capacity 3





Discrete-time models (DT, DDT)

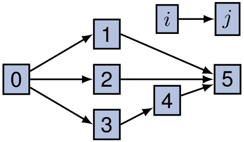
- Planning horizon divided into intervals of equal-length
- Binary variables per time interval and activity

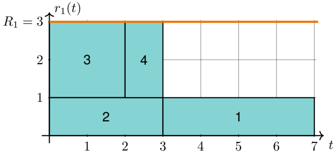
i	0	1	2	3	4	5	6	7	8	9
0										
1										
2										
3										
4										
5										

MILP models from the literature

Illustrative example

- Project activities $1, \ldots, 4$
- Single resource type, capacity 3





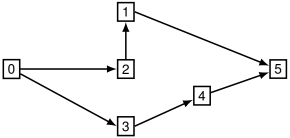
Discrete-time models (DT, DDT)

- Planning horizon divided into intervals of equal-length
- Binary variables per time interval and activity

i	0	1	2	3	4	5	6	7	8	9
0										
1										
2										
3										
4										
5										

Resource-flow model (FCT)

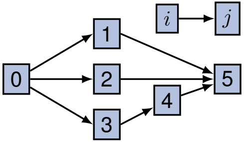
- Continuous start time variables
- Resource flows between activities

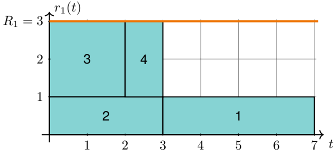


MILP models from the literature

Illustrative example

- Project activities $1, \dots, 4$
- Single resource type, capacity 3





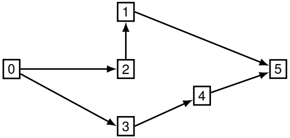
Discrete-time models (DT, DDT)

- Planning horizon divided into intervals of equal-length
- Binary variables per time interval and activity

i	0	1	2	3	4	5	6	7	8	9
0										
1										
2										
3										
4										
5										

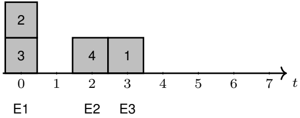
Resource-flow model (FCT)

- Continuous start time variables
- Resource flows between activities



On/off event-based model (OOE)

- Assign activities and start times to events



Outline I: MILP formulations

Modeling approaches

Continuous-time assignment-based MILP formulation

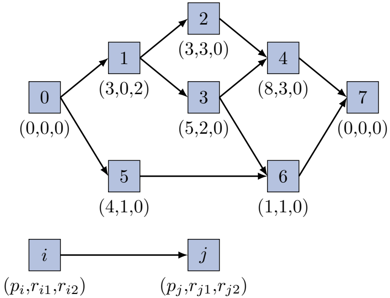
Computational results

Planning situation: RCPSP

Given:

- Set of activities $V = \{0, 1, \ldots, n, n+1\}$
- Activity $i \in V$: duration $p_i \geq 0$
- Set of precedence relations $E \subset V \times V$ among activities
- Set of resource types R; for each resource type $k \in R$
  - Resource capacity R_k
  - Required number of units r_{ik} for executing activity $i \in V$

Illustrative Example



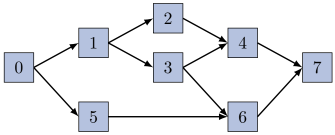
Resource capacities: $R_1 = 4$, $R_2 = 2$

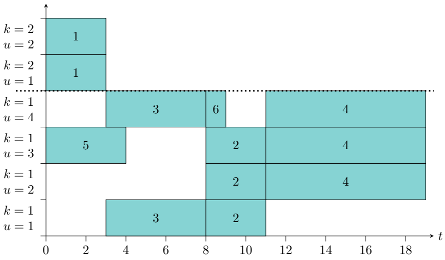
Planning situation: RCPSP

Sought: activity start times so that

- project duration minimized
- all precedence relations considered
- resource capacity never exceeded

Illustrative Example





CTAB formulation: notation

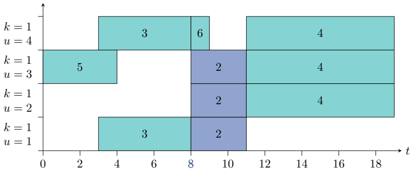
S_i Start time of activity i

r_{ik}^u $\begin{cases} = 1, \text{ if activity } i \text{ is assigned to unit } u \\ \text{of resource } k \\ = 0, \text{ otherwise} \end{cases}$

y_{ij} $\begin{cases} = 1, \text{ if activity } i \text{ must be completed} \\ \text{before the start of } j \\ = 0, \text{ otherwise} \end{cases}$

TE Transitive closure of E
T Planning horizon
ES_i Earliest possible start time of activity $i \in V$
LS_i Latest possible start time of activity $i \in V$

Illustrative example



CTAB formulation: notation

S_i Start time of activity i

r_{ik}^u $\begin{cases} = 1, & \text{if activity } i \text{ is assigned to unit } u \text{ of resource } k \\ = 0, & \text{otherwise} \end{cases}$

y_{ij} $\begin{cases} = 1, & \text{if activity } i \text{ must be completed before the start of } j \\ = 0, & \text{otherwise} \end{cases}$

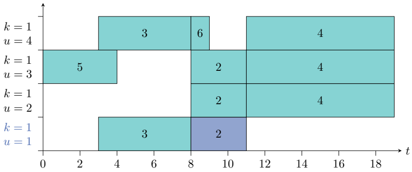
TE Transitive closure of E

T Planning horizon

ES_i Earliest possible start time of activity $i \in V$

LS_i Latest possible start time of activity $i \in V$

Illustrative example



CTAB formulation: notation

S_i Start time of activity i

r_{ik}^u $\begin{cases} = 1, \text{ if activity } i \text{ is assigned to unit } u \text{ of resource } k \\ = 0, \text{ otherwise} \end{cases}$

y_{ij} $\begin{cases} = 1, \text{ if activity } i \text{ must be completed before the start of } j \\ = 0, \text{ otherwise} \end{cases}$

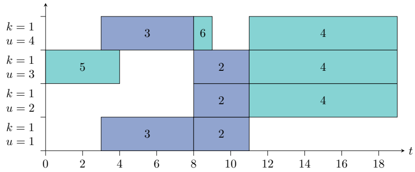
TE Transitive closure of E

T Planning horizon

ES_i Earliest possible start time of activity $i \in V$

LS_i Latest possible start time of activity $i \in V$

Illustrative example



CTAB formulation: notation

S_i Start time of activity i

$$r_{ik}^{u} \begin{cases} = 1, & \text{if activity } i \text{ is assigned to unit } u \text{ of resource } k \\ = 0, & \text{otherwise} \end{cases}$$

$$y_{ij} \begin{cases} = 1, & \text{if activity } i \text{ must be completed before the start of } j \\ = 0, & \text{otherwise} \end{cases}$$

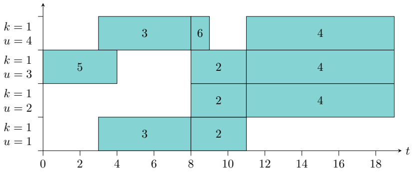
TE Transitive closure of E

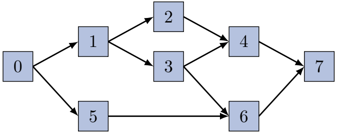
T Planning horizon

ES_i Earliest possible start time of activity $i \in V$

LS_i Latest possible start time of activity $i \in V$

Illustrative example





CTAB model formulation

Min. S_{n+1}

$$\sum_{u=1}^{R_k} r_{ik}^u = r_{ik} \quad (i \in V;\ k \in R) \tag{1}$$

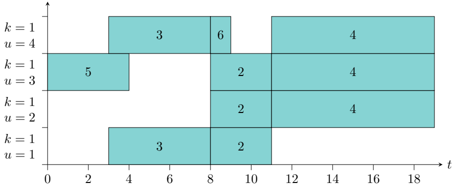
$$r_{ik}^u + r_{jk}^u \leq 1 + y_{ij} + y_{ji} \qquad (i,j \in V; k \in R; \; u = 1,\ldots,R_k : i < j, (i,j) \notin TE) \tag{2}$$

$$S_i + p_i \leq S_j \quad ((i,j) \in E) \tag{3}$$

$$S_i + p_i \leq S_j + T(1 - y_{ij}) \quad (i,j \in V : i \neq j, (i,j) \notin TE) \tag{4}$$

$$y_{ij} + y_{ji} \leq 1 \quad (i,j \in V : i \neq j, (i,j) \notin TE) \tag{5}$$

$$ES_i \leq S_i \leq LS_i \quad (i \in V) \tag{6}$$



(OF) objective is to minimize the project makespan

CTAB model formulation

Min. S_{n+1}

$$\sum_{u=1}^{R_k} r_{ik}^u = r_{ik} \quad (i \in V;\ k \in R) \tag{1}$$

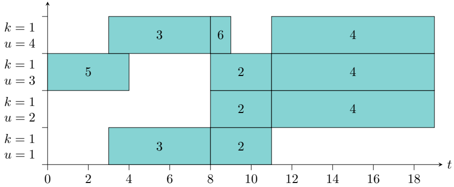
$$r_{ik}^u + r_{jk}^u \leq 1 + y_{ij} + y_{ji} \qquad (i,j \in V; k \in R; \; u = 1,\ldots,R_k : i < j, (i,j) \notin TE) \tag{2}$$

$$S_i + p_i \leq S_j \quad ((i,j) \in E) \tag{3}$$

$$S_i + p_i \leq S_j + T(1 - y_{ij}) \quad (i,j \in V : i \neq j, (i,j) \notin TE) \tag{4}$$

$$y_{ij} + y_{ji} \leq 1 \quad (i,j \in V : i \neq j, (i,j) \notin TE) \tag{5}$$

$$ES_i \leq S_i \leq LS_i \quad (i \in V) \tag{6}$$



(1) number of units of resource type k assigned to activity i must match the required number of units

CTAB model formulation

Min. S_{n+1}

$$\sum_{u=1}^{R_k} r_{ik}^u = r_{ik} \quad (i \in V;\ k \in R) \tag{1}$$

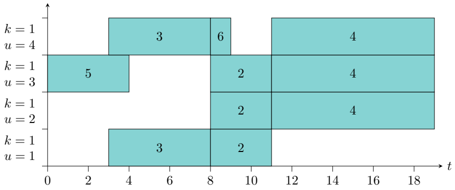
$$r_{ik}^u + r_{jk}^u \leq 1 + y_{ij} + y_{ji} \qquad (i,j \in V; k \in R; \quad u = 1,\dots,R_k : i < j, (i,j) \notin TE) \tag{2}$$

$$S_i + p_i \leq S_j \quad ((i,j) \in E) \tag{3}$$

$$S_i + p_i \leq S_j + T(1 - y_{ij}) \quad (i,j \in V : i \neq j, (i,j) \notin TE) \tag{4}$$

$$y_{ij} + y_{ji} \leq 1 \quad (i,j \in V : i \neq j, (i,j) \notin TE) \tag{5}$$

$$ES_i \leq S_i \leq LS_i \quad (i \in V) \tag{6}$$



(2) if the same resource unit is assigned to two activities i and j, then a sequencing is enforced between these two activities

CTAB model formulation

Min. S_{n+1}

$$\sum_{u=1}^{R_k} r_{ik}^u = r_{ik} \quad (i \in V;\ k \in R) \tag{1}$$

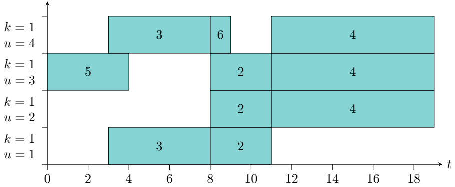
$$r_{ik}^u + r_{jk}^u \leq 1 + y_{ij} + y_{ji} \qquad (i,j \in V; k \in R; \\ u = 1,\dots,R_k : i < j, (i,j) \notin TE) \tag{2}$$

$$S_i + p_i \leq S_j \quad ((i,j) \in E) \tag{3}$$

$$S_i + p_i \leq S_j + T(1 - y_{ij}) \\ (i,j \in V : i \neq j, (i,j) \notin TE) \tag{4}$$

$$y_{ij} + y_{ji} \leq 1 \quad (i,j \in V : i \neq j, (i,j) \notin TE) \tag{5}$$

$$ES_i \leq S_i \leq LS_i \quad (i \in V) \tag{6}$$



(3) precedence relations

CTAB model formulation

Min. S_{n+1}

$$\sum_{u=1}^{R_k} r_{ik}^u = r_{ik} \quad (i \in V;\ k \in R) \qquad (1)$$

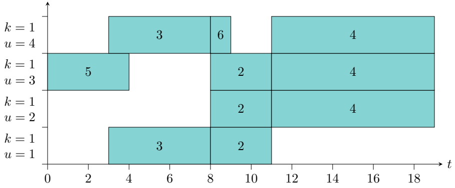
$$r_{ik}^u + r_{jk}^u \leq 1 + y_{ij} + y_{ji} \qquad (i, j \in V; k \in R; \; u = 1, \ldots, R_k : i < j, (i,j) \notin TE) \qquad (2)$$

$$S_i + p_i \leq S_j \quad ((i,j) \in E) \qquad (3)$$

$$S_i + p_i \leq S_j + T(1 - y_{ij}) \qquad (i, j \in V : i \neq j, (i,j) \notin TE) \qquad (4)$$

$$y_{ij} + y_{ji} \leq 1 \quad (i, j \in V : i \neq j, (i,j) \notin TE) \qquad (5)$$

$$ES_i \leq S_i \leq LS_i \quad (i \in V) \qquad (6)$$



(4) link of the the start time variables to the sequencing variables

CTAB model formulation

Min. S_{n+1}

$$\sum_{u=1}^{R_k} r_{ik}^u = r_{ik} \quad (i \in V;\ k \in R) \qquad (1)$$

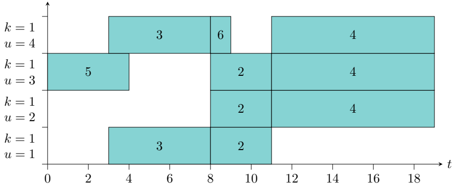
$$r_{ik}^u + r_{jk}^u \leq 1 + y_{ij} + y_{ji} \qquad (i,j \in V; k \in R; \; u = 1,\dots,R_k : i < j, (i,j) \notin TE) \qquad (2)$$

$$S_i + p_i \leq S_j \quad ((i,j) \in E) \qquad (3)$$

$$S_i + p_i \leq S_j + T(1 - y_{ij}) \quad (i,j \in V : i \neq j, (i,j) \notin TE) \qquad (4)$$

$$y_{ij} + y_{ji} \leq 1 \quad (i,j \in V : i \neq j, (i,j) \notin TE) \qquad (5)$$

$$ES_i \leq S_i \leq LS_i \quad (i \in V) \qquad (6)$$



(5) either activity i precedes j, j precedes i, or i and j are processed in parallel

CTAB model formulation

Min. S_{n+1}

$$\sum_{u=1}^{R_k} r_{ik}^u = r_{ik} \quad (i \in V;\ k \in R) \tag{1}$$

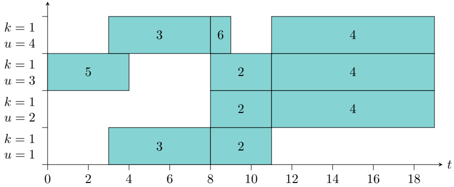
$$r_{ik}^u + r_{jk}^u \leq 1 + y_{ij} + y_{ji} \qquad (i,j \in V; k \in R; \\ u = 1, \ldots, R_k : i < j, (i,j) \notin TE) \tag{2}$$

$$S_i + p_i \leq S_j \quad ((i,j) \in E) \tag{3}$$

$$S_i + p_i \leq S_j + T(1 - y_{ij}) \\ (i,j \in V : i \neq j, (i,j) \notin TE) \tag{4}$$

$$y_{ij} + y_{ji} \leq 1 \quad (i,j \in V : i \neq j, (i,j) \notin TE) \tag{5}$$

$$ES_i \leq S_i \leq LS_i \quad (i \in V) \tag{6}$$



(6) each activity start between its earliest and latest start times

Outline I: MILP Formulations

Modeling approaches

Continuous-time assignment-based MILP formulation

Computational results

Experimental design

- Models compared:

Abbrev.	DT/CT	Source
DT	DT	Pritsker et al. (1969)
DDT	DT	Christofides et al. (1987)
FCT	CT	Artigues et al. (2003)
OOE	CT	Kone et al. (2011)
SEQ	CT	Klein et al. (2024)
CTAB	CT	Gnägi et al. (2018)
CTAB_EXT	CT	Gnägi et al. (2018)

- Models implemented in Python 3.8
- Gurobi 9.1.2, limited to 2 threads
- Intel(R) CPU 3.10GHz, 128 GB RAM
- Time limit per instance: 500 sec.
- Test set J30 (PSPLIB; Kolisch & Sprecher, 1996)
  - 30 activities
  - 4 renewable resources
  - 480 instances

Computational results

Model	# Feas	# Opt	# Best	$\text{Gap}^{\text{UB}^*-\text{CPM}}$	Time
DT	**480**	443	462	13.59%	54.16
DDT	479	437	451	13.66%	67.89
FCT	**480**	458	473	13.45%	40.21
OOE	**480**	0	302	15.50%	501.08
SEQ	**480**	**471**	**478**	**13.43%**	**15.28**
CTAB	**480**	376	426	14.06%	120.77
CTAB_EXT	**480**	422	454	13.60%	71.45

Outline

Part I: MILP formulations of the RCPSP

Part II: Multi-site resource-constrained project scheduling

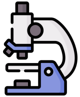
Part III: Workload balancing in resource-constrained project scheduling

Part IV: Conclusions

Multi-site resource-constrained project scheduling

Subject

- Project distributed among multiple sites
  - Alternative sites for the execution of the activities
  - Some resource units mobile, others non-mobile
  - Transportation times between sites
- Objective: minimize project duration (NP-hard problem)
- Sample applications
  - Pooling of personnel in health care or R&D (cf. Laurent et al. 2017)
  - Distributed make-to-order production in supply chains



Contribution

- CT MILP formulation
- Matheuristic based on continuous-time MILP formulation

All images: Flaticon.com

Outline II: Multi-site project scheduling

Planning situation

CTAB-based MILP formulation

Relax-optimize-and-fix matheuristic

Computational results

Planning problem

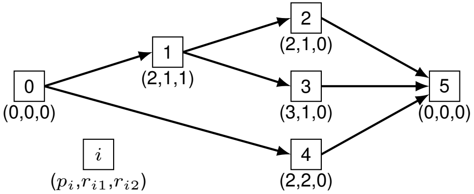
Given:

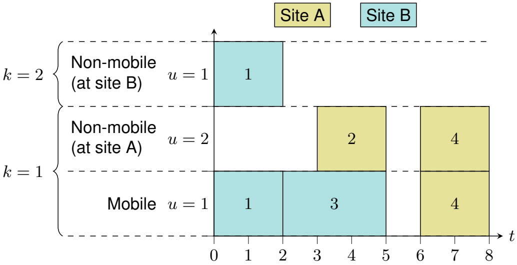
- Set of activities $V = \{0, 1, \ldots, n, n+1\}$ with duration p_i $(i \in V)$
- Set of precedence relations $E \subseteq V \times V$ among activities
- Set of sites L; transportation time $\delta_{ll'}$ between sites $l, l' \in L \times L$
- Set of resource types R; for each resource type $k \in R$
  - Available number of units R_k
  - Required number of units r_{ik} for executing activity $i \in V$
  - Indicator M_{ku} for unit $u \in \{1, \ldots, R_k\}$: $= 1$ mobile; $= 0$ else
  - Site loc_{ku} of non-mobile unit $u \in \{1, \ldots, R_k\}$

Sought: start time and site for each activity s.t.

- project duration is minimal,
- all precedence relationships are taken into account,
- resource usage never exceeds the prescribed resource availabilities, and
- transportation times between sites are taken into account

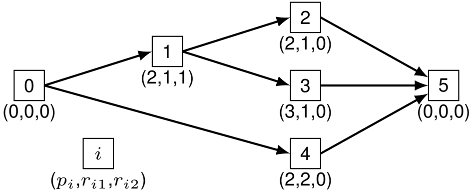
Planning problem: illustrative example

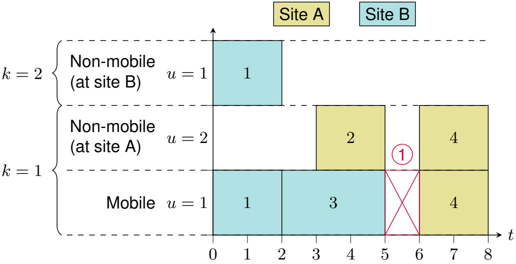




- $V = \{0, 1, \ldots, 4, 5\}$
- $R = \{1, 2\}$, $R_1 = 2$, $R_2 = 1$
- $L = \{\mathsf{A}, \mathsf{B}\}$, $\delta_{\mathsf{AB}} = \delta_{\mathsf{BA}} = 1$

Planning problem: illustrative example

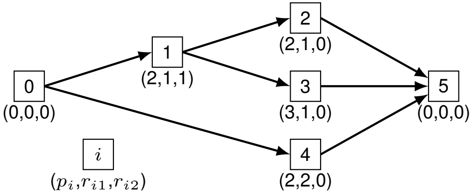


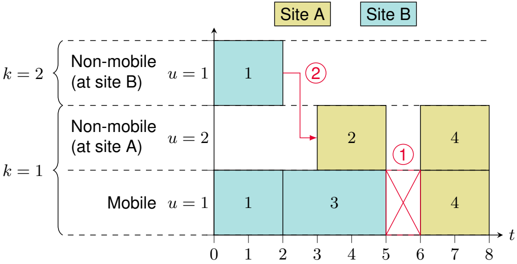


① Time for moving mobile ressource unit from site B to site A

- $V = \{0, 1, \ldots, 4, 5\}$
- $R = \{1, 2\}$, $R_1 = 2$, $R_2 = 1$
- $L = \{\text{A}, \text{B}\}$, $\delta_{\text{AB}} = \delta_{\text{BA}} = 1$

Planning problem: illustrative example





- $V = \{0, 1, \ldots, 4, 5\}$
- $R = \{1, 2\}$, $R_1 = 2$, $R_2 = 1$
- $L = \{\mathsf{A}, \mathsf{B}\}$, $\delta_{\mathsf{AB}} = \delta_{\mathsf{BA}} = 1$

(1) Time for moving mobile ressource unit from site B to site A

(2) Time for moving output of activity 1 to activity 2, i.e., from site B to site A

Outline II: Multi-site project scheduling

Planning situation

CTAB-based MILP formulation

Relax-optimize-and-fix matheuristic

Computational results

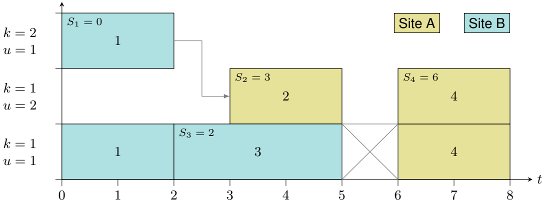
MILP model: decision variables

S_i Start time of activity i

s_{il} $\begin{cases} = 1, & \text{if activity } i \text{ is executed at site } l \\ = 0, & \text{otherwise} \end{cases}$

r_{ik}^u $\begin{cases} = 1, & \text{if activity } i \text{ is assigned to unit } u \text{ of resource type } k \\ = 0, & \text{otherwise} \end{cases}$

y_{ij} $\begin{cases} = 1, & \text{if activity } i \text{ must be completed before the start of activity } j \\ = 0, & \text{otherwise} \end{cases}$



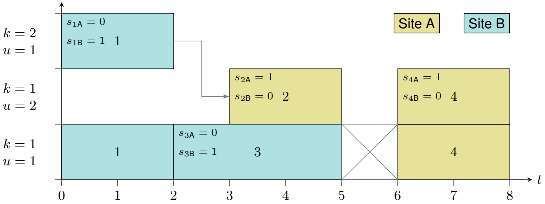
MILP model: decision variables

S_i Start time of activity i

s_{il} $\begin{cases} = 1, & \text{if activity } i \text{ is executed at site } l \\ = 0, & \text{otherwise} \end{cases}$

r_{ik}^{u} $\begin{cases} = 1, & \text{if activity } i \text{ is assigned to unit } u \text{ of resource type } k \\ = 0, & \text{otherwise} \end{cases}$

y_{ij} $\begin{cases} = 1, & \text{if activity } i \text{ must be completed before the start of activity } j \\ = 0, & \text{otherwise} \end{cases}$



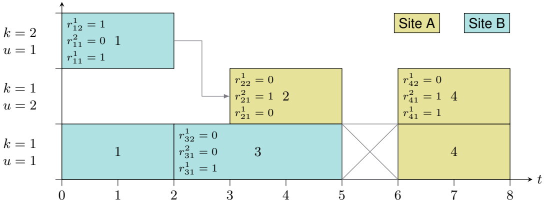
MILP model: decision variables

S_i Start time of activity i

$$s_{il} \begin{cases} = 1, & \text{if activity } i \text{ is executed at site } l \\ = 0, & \text{otherwise} \end{cases}$$

$$r_{ik}^u \begin{cases} = 1, & \text{if activity } i \text{ is assigned to unit } u \text{ of resource type } k \\ = 0, & \text{otherwise} \end{cases}$$

$$y_{ij} \begin{cases} = 1, & \text{if activity } i \text{ must be completed before the start of activity } j \\ = 0, & \text{otherwise} \end{cases}$$



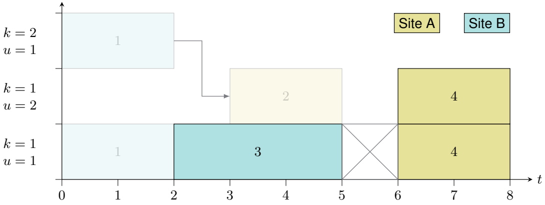
MILP model: decision variables

S_i Start time of activity i

$$s_{il} \begin{cases} = 1, & \text{if activity } i \text{ is executed at site } l \\ = 0, & \text{otherwise} \end{cases}$$

$$r_{ik}^{u} \begin{cases} = 1, & \text{if activity } i \text{ is assigned to unit } u \\ & \text{of resource type } k \\ = 0, & \text{otherwise} \end{cases}$$

$$y_{ij} \begin{cases} = 1, & \text{if activity } i \text{ must be completed} \\ & \text{before the start of activity } j \\ = 0, & \text{otherwise} \end{cases}$$



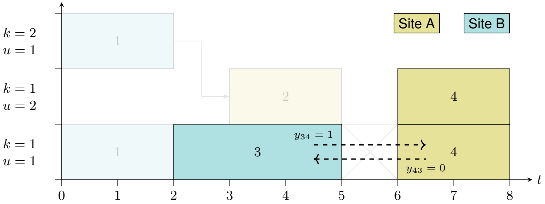
MILP model: decision variables

S_i Start time of activity i

s_{il} $\begin{cases} = 1, & \text{if activity } i \text{ is executed at site } l \\ = 0, & \text{otherwise} \end{cases}$

r_{ik}^u $\begin{cases} = 1, & \text{if activity } i \text{ is assigned to unit } u \text{ of resource type } k \\ = 0, & \text{otherwise} \end{cases}$

y_{ij} $\begin{cases} = 1, & \text{if activity } i \text{ must be completed before the start of activity } j \\ = 0, & \text{otherwise} \end{cases}$



MILP model: notation

V	Set of activities ($V = \{0, 1, \ldots, n+1\}$)	L	Set of sites
$\dot{V}$	Set of real activities ($\dot{V} = \{1, 2, \ldots, n\}$)	$\delta_{ll'}$	Transportation time between sites $l, l' \in L \times L$
E	Set of precedence relations	loc_{ku}	Site for non-mobile unit $u \in \{1, \ldots, R_k\}$ of resource type $k \in R$
TE	Transitive closure of E	M_{ku}	$\begin{cases} = 1, & \text{if unit } u \in \{1, \ldots, R_k\} \text{ of resource type } k \in R \text{ is mobile} \\ = 0, & \text{otherwise} \end{cases}$
R	Set of resource types	δ^{max}	Longest transportation time between all pairs of sites
p_i	Duration of activity $i \in V$		
R_k	Available number of units of resource type $k \in R$		
r_{ik}	Required number of units of resource type $k \in R$ for executing activity $i \in V$		

MILP model

Minimize project duration:

Min. S_{n+1}

Each real activity is executed at exactly one site:

$$\sum_{l \in L} s_{il} = 1 \quad (i \in \dot{V}) \tag{7}$$

For each resource type, required number of units are assigned:

$$\sum_{u=1}^{R_k} r_{ik}^{u} = r_{ik} \quad (i \in \dot{V};\ k \in R : r_{ik} > 0) \tag{8}$$

Precedence relations among real activities:

$$S_i + p_i + (s_{il} + s_{jl'} - 1)\delta_{ll'} \leq S_j \quad (i, j \in \dot{V} \times \dot{V} :\ (i, j) \in E;\ l, l' \in L \times L) \tag{9}$$

MILP model

Minimize project duration:

Min. S_{n+1}

Each real activity is executed at exactly one site:

$$\sum_{l \in L} s_{il} = 1 \quad (i \in \dot{V}) \tag{7}$$

For each resource type, required number of units are assigned:

$$\sum_{u=1}^{R_k} r_{ik}^u = r_{ik} \quad (i \in \dot{V};\ k \in R : r_{ik} > 0) \tag{8}$$

Precedence relations among real activities:

$$S_i + p_i + (s_{il} + s_{jl'} - 1)\delta_{ll'} \leq S_j \quad (i, j \in \dot{V} \times \dot{V} :\ (i, j) \in E;\ l, l' \in L \times L) \tag{9}$$

MILP model

Minimize project duration:

Min. S_{n+1}

Each real activity is executed at exactly one site:

$$\sum_{l \in L} s_{il} = 1 \quad (i \in \dot{V}) \tag{7}$$

For each resource type, required number of units are assigned:

$$\sum_{u=1}^{R_k} r_{ik}^u = r_{ik} \quad (i \in \dot{V};\ k \in R : r_{ik} > 0) \tag{8}$$

Precedence relations among real activities:

$$S_i + p_i + (s_{il} + s_{jl'} - 1)\delta_{ll'} \leq S_j \quad (i, j \in \dot{V} \times \dot{V} :\ (i, j) \in E;\ l, l' \in L \times L) \tag{9}$$

MILP model

Minimize project duration:

Min. S_{n+1}

Each real activity is executed at exactly one site:

$$\sum_{l \in L} s_{il} = 1 \quad (i \in \dot{V}) \tag{7}$$

For each resource type, required number of units are assigned:

$$\sum_{u=1}^{R_k} r_{ik}^u = r_{ik} \quad (i \in \dot{V};\ k \in R : r_{ik} > 0) \tag{8}$$

Precedence relations among real activities:

$$S_i + p_i + (s_{il} + s_{jl'} - 1)\delta_{ll'} \leq S_j \quad (i, j \in \dot{V} \times \dot{V} :\ (i, j) \in E;\ l, l' \in L \times L) \tag{9}$$

MILP model

Real activities must be processed sequentially if assigned to at least one common resource unit:

$$r_{ik}^u + r_{jk}^u \leq y_{ij} + y_{ji} + 1$$
$$(i, j \in \dot{V} \times \dot{V};\ k \in R;\ u \in \{1, \ldots, R_k\} : i < j,\ (i,j) \notin TE,\ r_{ik} > 0,\ r_{jk} > 0) \quad (10)$$

$$S_i + p_i + (s_{il} + s_{jl'} - 1)\delta_{ll'} \leq S_j + (\sum_{i \in V} p_i + n\delta^{max})(1 - y_{ij})$$
$$(i, j \in \dot{V} \times \dot{V} :\ i \neq j,\ (i,j) \notin TE;\ l, l' \in L \times L) \quad (11)$$

No real activity completed after project completion:

$$S_i + p_i \leq S_{n+1} \quad (i \in \dot{V} \cup \{0\}) \quad (12)$$

Fixed site-assignments of non-mobile resource units:

$$r_{ik}^u \leq s_{i,loc_{ku}}\ (i \in \dot{V};\ k \in R;\ u = 1, \ldots, R_k :\ M_{ku} = 0, r_{ik} > 0) \quad (13)$$

MILP model

Real activities must be processed sequentially if assigned to at least one common resource unit:

$$r_{ik}^u + r_{jk}^u \leq y_{ij} + y_{ji} + 1$$
$$(i,j \in \dot{V} \times \dot{V};\ k \in R;\ u \in \{1,\dots,R_k\} : i < j,\ (i,j) \notin TE,\ r_{ik} > 0,\ r_{jk} > 0) \quad (10)$$

$$S_i + p_i + (s_{il} + s_{jl'} - 1)\delta_{ll'} \leq S_j + (\sum_{i \in V} p_i + n\delta^{max})(1 - y_{ij})$$
$$(i,j \in \dot{V} \times \dot{V} :\ i \neq j,\ (i,j) \notin TE;\ l,l' \in L \times L) \quad (11)$$

No real activity completed after project completion:

$$S_i + p_i \leq S_{n+1} \quad (i \in \dot{V} \cup \{0\}) \quad (12)$$

Fixed site-assignments of non-mobile resource units:

$$r_{ik}^u \leq s_{i,loc_{ku}}\ (i \in \dot{V};\ k \in R;\ u = 1,\dots,R_k :\ M_{ku} = 0, r_{ik} > 0) \quad (13)$$

MILP model

Real activities must be processed sequentially if assigned to at least one common resource unit:

$$r_{ik}^u + r_{jk}^u \leq y_{ij} + y_{ji} + 1$$
$$(i,j \in \dot{V} \times \dot{V};\ k \in R;\ u \in \{1,\ldots,R_k\} : i < j,\ (i,j) \notin TE,\ r_{ik} > 0,\ r_{jk} > 0) \quad (10)$$

$$S_i + p_i + (s_{il} + s_{jl'} - 1)\delta_{ll'} \leq S_j + (\sum_{i \in V} p_i + n\delta^{max})(1 - y_{ij})$$
$$(i,j \in \dot{V} \times \dot{V} :\ i \neq j,\ (i,j) \notin TE;\ l,l' \in L \times L) \quad (11)$$

No real activity completed after project completion:

$$S_i + p_i \leq S_{n+1} \quad (i \in \dot{V} \cup \{0\}) \quad (12)$$

Fixed site-assignments of non-mobile resource units:

$$r_{ik}^u \leq s_{i,loc_{ku}} \ (i \in \dot{V};\ k \in R;\ u = 1,\ldots,R_k :\ M_{ku} = 0, r_{ik} > 0) \quad (13)$$

Outline II: Multi-site project scheduling

Planning situation

CTAB-based MILP formulation

Relax-optimize-and-fix matheuristic

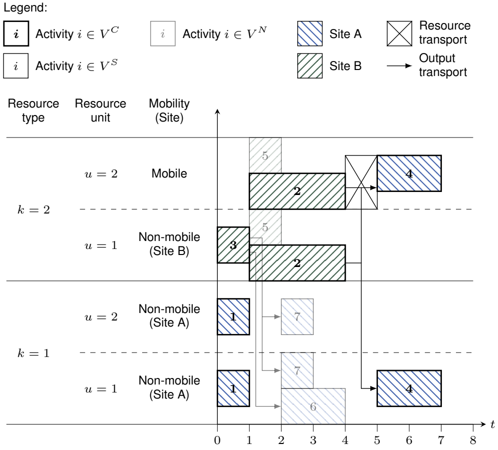
Computational results

Relax-optimize-and-fix matheuristic

Main idea: iteratively schedule a subset of activities by solving a relaxation of MILP model

Overview

1) Apply priority rule and select subset of c activities with highest priorities

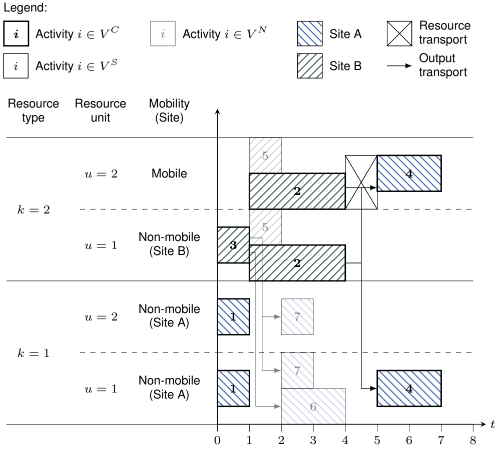


Relax-optimize-and-fix matheuristic

Main idea: iteratively schedule a subset of activities by solving a relaxation of MILP model

Overview

1) Apply priority rule and select subset of c activities with highest priorities
2) **Relax** binary sequencing variables for non-selected activities

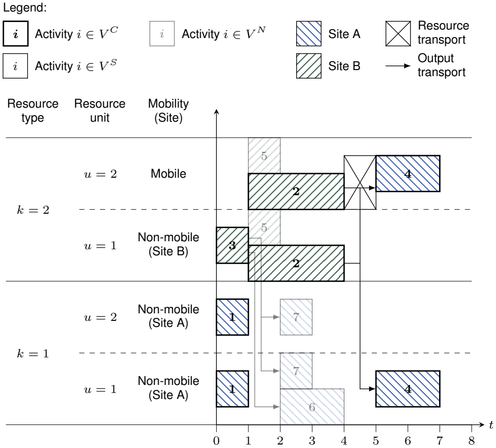


Relax-optimize-and-fix matheuristic

Main idea: iteratively schedule a subset of activities by solving a relaxation of MILP model

Overview

1) Apply priority rule and select subset of c activities with highest priorities
2) **Relax** binary sequencing variables for non-selected activities
3) **Optimize** resulting relaxation of MILP

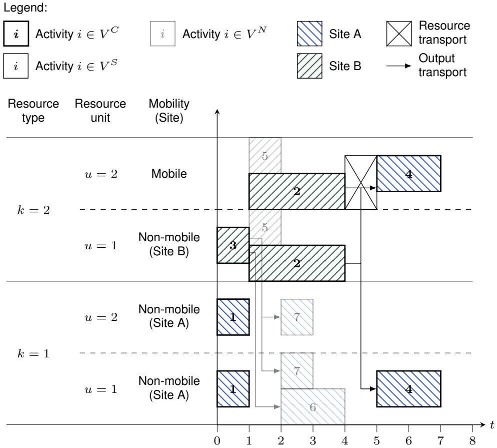


Relax-optimize-and-fix matheuristic

Main idea: iteratively schedule a subset of activities by solving a relaxation of MILP model

Overview

1) Apply priority rule and select subset of c activities with highest priorities
2) **Relax** binary sequencing variables for non-selected activities
3) **Optimize** resulting relaxation of MILP
4) **Fix** values of sequencing variables for $s \leq c$ activities with highest priorities

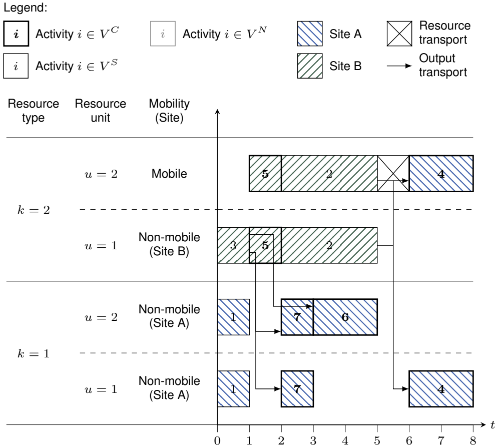


Relax-optimize-and-fix matheuristic

Main idea: iteratively schedule a subset of activities by solving a relaxation of MILP model

Overview

1) Apply priority rule and select subset of c activities with highest priorities
2) **Relax** binary sequencing variables for non-selected activities
3) **Optimize** resulting relaxation of MILP
4) **Fix** values of sequencing variables for $s \leq c$ activities with highest priorities
5) Select s activities not scheduled yet, impose binary sequencing variables for them and go to **3)**; if all activities scheduled, stop



Outline II: Multi-site project scheduling

Planning situation

CTAB-based MILP formulation

Relax-optimize-and-fix matheuristic

Computational results

Computational results: experimental design

- Analyzed exact approaches
  - CT: novel continuous-time model
  - DT: discrete-time model of Laurent et al. (2017)
- Analyzed heuristic approaches:
  - MH: novel matheuristic
  - LS, SA, ILS LS and ILS SA: four metaheuristics of Laurent et al. (2017)

- Test sets MSj30 and MSj60
  - Generated by Laurent et al. (2017)
  - Adapting well-known single-site RCPSP instances j30 and j60 (Kolisch & Sprecher 1996)
  - $1{,}920$ instances: $n = \{30, 60\}$ activities and $|L| = \{2, 3\}$ sites
- HP workstation: Intel Xeon CPU with 2.20GHz, 128 GB RAM
- Implementation in Python 3.7
- Gurobi 9.1 as solver; default settings

Computational results: exact approaches

All MSj30 instances

#Act	#Sites	Model	#Feas	#Opt	Gap^{CP} (%)	CPU (s)
30	2	CT	**480**	**327**	**25.86**	**112.00**
		DT	455	272	34.93	159.44
30	3	CT	**480**	**284**	**33.92**	**138.40**
		DT	444	224	51.80	190.93

MSj30 instances with feasible solution for both models

#Act	#Sites	Model	#Feas	#Opt	Gap^{CP} (%)	CPU (s)
30	2	CT	455	**323**	**22.04**	**102.92**
		DT	455	272	34.93	151.71
30	3	CT	444	**279**	**28.51**	**127.50**
		DT	444	224	51.80	182.08

Computational results: heuristic approaches

All MSj30 instances

#Act	#Sites	Approach	Gap^{CP} (%)	$\#MH^{+}$	$\#MH^{-}$	CPU (s)
		MH	**25.02**	0	0	61.86
		LS	29.72	258	61	**55.50**
30	2	SA	26.50	178	93	55.51
		ILS LS	25.86	153	103	70.35
		ILS SA	26.04	152	110	70.80
		MH	**32.35**	0	0	61.32
		LS	37.65	276	89	**60.17**
30	3	SA	34.11	219	119	60.44
		ILS LS	33.42	180	142	76.53
		ILS SA	33.37	192	152	76.42

Computational results: heuristic approaches

All MSj60 instances

#Act	#Sites	Approach	Gap^{CP} (%)	$\#MH^{+}$	$\#MH^{-}$	CPU (s)
		MH	**24.99**	0	0	**123.18**
		LS	27.95	231	133	128.97
60	2	SA	26.19	211	168	130.26
		ILS LS	26.41	198	163	168.71
		ILS SA	26.57	197	161	168.68
		MH	**35.42**	0	0	**137.08**
		LS	38.29	289	123	142.76
60	3	SA	35.51	235	172	143.66
		ILS LS	36.03	242	172	185.79
		ILS SA	36.43	249	149	185.98

Outline

Part I: MILP formulations of the RCPSP

Part II: Multi-site resource-constrained project scheduling

Part III: Workload balancing in resource-constrained project scheduling

Part IV: Conclusions

Proceedings of IEEM 2024

Workload-Balancing Constraints in a Continuous-Time Integer Programming Formulation for the Resource-Constrained Project Scheduling Problem

N. Ackermann, T. Bigler, N. Trautmann
Department of Business Administration, University of Bern, 3012 Bern, Switzerland
email: nina.ackermann@unibe.ch, tamara.bigler@unibe.ch, norbert.trautmann@unibe.ch

***Abstract*—In project management, the resource allocation problem consists of determining a schedule for the set of project activities that are related to each other by prescribed precedence relations and that require some time and some scarce resources to complete. In general, the goal is to minimize the project duration or time-to-market. In many cases, each resource represents a team of people with specific skills, such as engineering or marketing specialists. To promote team productivity and cohesion, it is often desirable to balance the workload within the team, i.e. among the resource units. We analyze two alternative approaches to formulate appropriate workload-balancing constraints in a mixed-binary linear optimization problem. In the first approach, the maximum deviation of each unit's workload from the average workload is bounded, and in the second approach, the maximum workload difference between any pair of units is bounded. Our computational results for a standard test set from the literature show that balanced workloads can generally be achieved without increasing project duration; moreover, the second approach provides more flexibility, resulting in fewer instances for which no feasible solution exists.**

***Keywords*—Project Management, Workload Balancing, Operations Research, Mathematical Programming**

I. INTRODUCTION

Scheduling refers to the allocation of available resources, i.e. machines or people, over time to perform a given set of activities or tasks. In applications such as staff scheduling or parallel machine scheduling, the workload should often be distributed as evenly as possible among the resources (see, for example, [1]). Depending on the application, workload balancing may be the objective criterion to be optimized or, more often, an additional constraint to be considered.

We address the problem of workload balancing in the context of project scheduling (cf. [2]). We consider a single project consisting of a set of activities. Given pairs of these activities are related by a completion-start precedence. Each activity takes some time to perform; during this time, given amounts of different resource types are required. A given number of units of each resource type is available for the execution of the project. What is sought is a project schedule, i.e. a start time for each activity, such that the project duration (or the time-to-market) is minimized, the sequence of the activities is in accordance with the prescribed precedence relations, at no time more units of any resource type are required than are available, and the workload imbalance among the individual units of each resource type does not exceed a prescribed value.

To the best of our knowledge, workload balancing has not been discussed in the extensive literature on project scheduling (see [3]). For an overview of the state of the art in workload balancing in parallel machine scheduling and workforce scheduling, see [1] and [4]. [5] show that in a parallel machine environment, a makespan-minimal schedule does not necessarily have a balanced workload, and vice versa.

In this paper, we analyze two alternative approaches to considering workload balancing as an additional constraint in the planning situation described above. The first approach is to bound, for each resource type, the maximum deviation of a unit's workload from the average workload of all units of that resource type. In contrast, the second approach is to bound, for each resource type, the maximum difference in unit workload among all pairs of units. To compare these two approaches, we present how to extend the formulation of the described planning situation without workload balancing as a mixed binary linear optimization problem presented in [6] by respective additional constraints. In an experimental performance analysis, we applied the resulting formulations to the standard test-set J30 (cf. [7]) widely used in the project scheduling literature. The results of our analysis show that, in general, balanced workloads can be achieved without increasing project duration; moreover, under the second approach, which provides more flexibility, a feasible solution exists for more problem instances of the test set.

This paper is organized as follows. In Section II, we provide an example to illustrate the planning situation and the two alternative workload balancing approaches considered in the following. In Section III, we present the basic MIP formulation and its alternative extensions by workload balancing constraints. In Section IV, we summarize the results of our experimental performance analysis. In Section V, we draw some conclusions and provide some possible future research directions.

II. ILLUSTRATIVE EXAMPLE

In this section, we illustrate, by means of an example project, the two alternative workload-balancing approaches outlined in Section I.

The project consists of $n = 6$ activities $i = 1, \dots, 6$, and two resource types $k = 1$ and $k = 2$ are available to execute them, where resource $k = 1$ consists of $R_1 = 4$ units and resource $k = 2$ consists of $R_2 = 2$ units. As usual, we represent the start and completion of the project by two fictitious activities $i = 0$ and $i = 7$, both of which have a duration of 0 and require no resources. Set $V := \{0, \dots, 7\}$

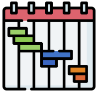
Subject and Contribution

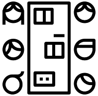
Subject

- Single-site RCPSP
- Frequently, resource types represent teams of people with specific skills
- Workload of a unit: total duration of assigned activities
- Foster team productivity and cohesion (cf., e.g., Hoegl & Gmuenden 2001) by balancing workload across team

Contribution

- Analysis of two alternative approaches to considering workload balancing
- Formulation of additional constraints in CTAB
- Computational results: balanced workloads can generally be achieved without significantly increasing project duration





All images: Flaticon.com

Outline III: Workload balancing in project scheduling

Planning situation

CTAB-based MILP formulation

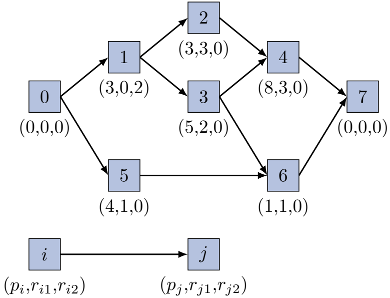
Computational results

Planning situation: project information

Given:

- Set of activities $V = \{0, 1, \ldots, n, n+1\}$
- Activity $i \in V$: duration $p_i \geq 0$
- Set of precedence relations $E \subset V \times V$ among activities
- Set of resource types R; for each resource type $k \in R$
  - Resource capacity R_k
  - Required number of units r_{ik} for executing activity $i \in V$

Illustrative Example



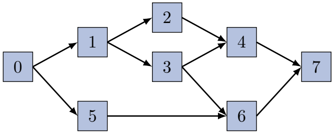
Resource capacities: $R_1 = 4$, $R_2 = 2$

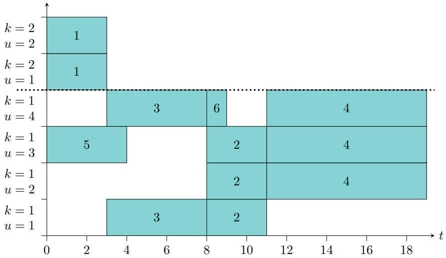
Planning situation: RCPSP

Sought: activity start times so that

- project duration minimized
- all precedence relations considered
- resource capacity never exceeded

Illustrative Example





Planning situation: RCPSP

Sought: activity start times so that

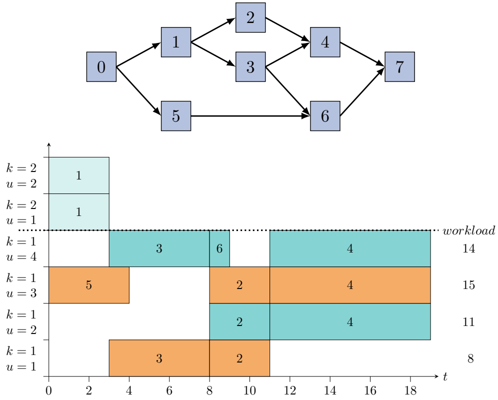
- project duration minimized
- all precedence relations considered
- resource capacity never exceeded

Consideration of workload balancing

Without additional constraints

- Minimum workload $5 + 3 = 8$
- Maximum workload $4 + 3 + 8 = 15$

Illustrative Example



Planning situation: MaxDev approach

Sought: activity start times so that

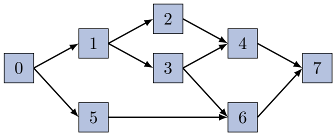
- project duration minimized
- all precedence relations considered
- resource capacity never exceeded

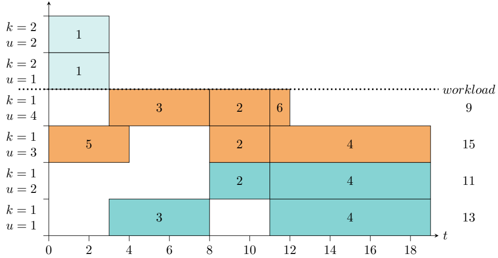
Consideration of workload balancing

MaxDev approach: limit deviation of each unit's workload from average unit workload

- Average workload $\frac{48}{4} = 12$
- E.g., maximum deviation of 3
- Minimum workload $5 + 3 + 1 = 9$
- Maximum workload $4 + 3 + 8 = 15$

Illustrative Example





Planning situation: MaxDiff approach

Sought: activity start times so that

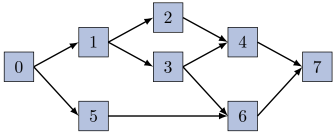
- project duration minimized
- all precedence relations considered
- resource capacity never exceeded

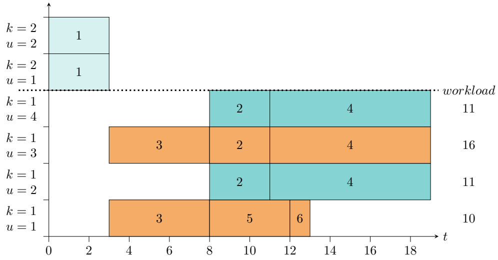
Consideration of workload balancing

MaxDiff approach: limit difference between workload of any two units

- E.g., maximum difference of 6
- Minimum workload $5 + 4 + 1 = 10$
- Maximum workload $5 + 3 + 8 = 16$

Illustrative Example





Outline III: Workload balancing in project scheduling

Planning situation

CTAB-based MILP formulation

Computational results

Workload balancing: MaxDev approach

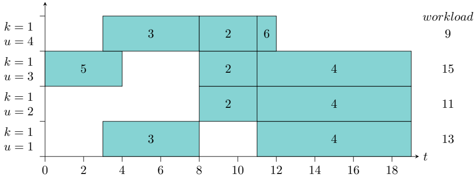
- δ: allowed deviation from the average
- Minimum/maximum unrounded workload

$$\underline{\delta}_k := (1-\delta)\frac{1}{R_k}\sum_{i=1}^{n} p_i r_{ik} \quad (k \in R) \qquad (14)$$

$$\overline{\delta}_k := (1+\delta)\frac{1}{R_k}\sum_{i=1}^{n} p_i r_{ik} \quad (k \in R) \qquad (15)$$

- Workload-balancing constraint

$$\lfloor \underline{\delta}_k \rfloor \leq \sum_{i \in V} p_i r_{ik}^u \leq \lceil \overline{\delta}_k \rceil$$
$$(k \in R;\ u = 1, \dots, R_k) \qquad (16)$$



Example for $k = 1$ and $\delta = 20\%$:

$$\underline{\delta}_1 := (1-0.2)\frac{48}{4} = \frac{48}{5} \qquad (6)$$

$$\overline{\delta}_1 := (1+0.2)\frac{48}{4} = \frac{72}{5} \qquad (7)$$

$$9 \leq \sum_{i \in V} p_i r_{i1}^u \leq 15 \quad (u = 1, \dots, 4) \qquad (8)$$

Workload balancing: MaxDiff approach

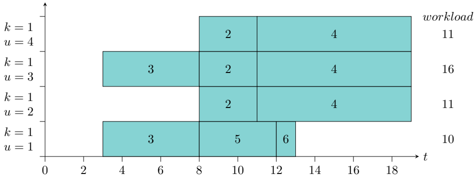
- Auxiliary variables $\underline{r}_k$ and $\overline{r}_k$
- Additional constraints

$$\underline{r}_k \leq \sum_{i \in V} p_i r_{ik}^u \leq \overline{r}_k$$

$$(k \in R;\ u = 1, \ldots, R_k) \qquad (9)$$

- workload-balancing constraint

$$\overline{r}_k - \underline{r}_k \leq \lceil \overline{\delta}_k \rceil - \lfloor \underline{\delta}_k \rfloor \quad (k \in R) \qquad (10)$$



Example for $k = 1$ and $\delta = 20\%$:

$$\underline{r}_1 \leq \sum_{i \in V} p_i r_{i1}^u \leq \overline{r}_1 \quad (u = 1, \ldots, 4) \qquad (9)$$

$$\overline{r}_1 - \underline{r}_1 \leq 15 - 9 \qquad (10)$$

Outline III: Workload balancing in project scheduling

Planning situation

CTAB-based MILP formulation

Computational results

Computational results: experimental design

Analyzed MILP models

- CTAB model without workload-balancing constraints
- maxDev approach
- maxDiff approach

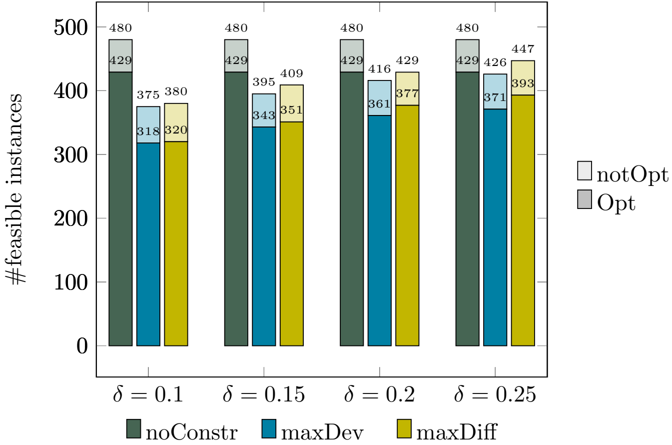
Test set: J30 (Kolisch & Sprecher, 1996)

- 480 RCPSP instances
- $n = 30$ activities

Test environment

- Implementation in Python 3.10.6
- Apple M1 Ultra 3.2 GHz CPU,
  128 GB RAM
- Gurobi 12.0 as solver (maximum 2 threads)
- CPU time limit: 300 seconds per instance

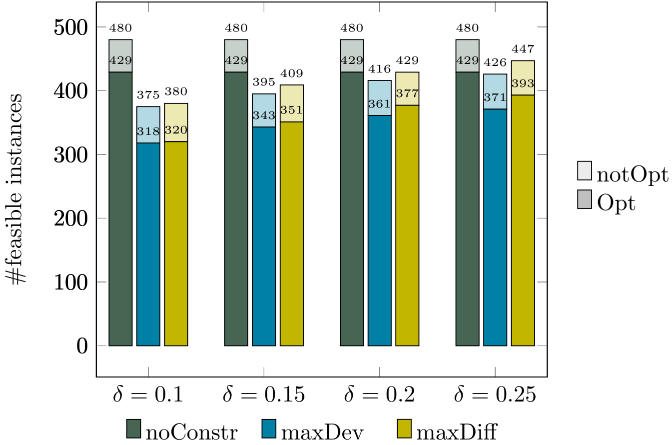
Computational results (preliminary)



Feasibility

- Imposing workload balancing can lead to infeasibility
- More feasible instances under maxDiff approach
- More feasible instances for larger values of δ

Computational results (preliminary)



Model performance

- Feasible solution to each instance (if any)
- Stable percentage of instances solved to optimality

Impact on project duration

- With workload balancing, increase of minimal project duration rather small and in few instances only
- Ex-post balancing often not possible

Outline

Part I: MILP formulations of the RCPSP

Part II: Multi-site resource-constrained project scheduling

Part III: Workload balancing in resource-constrained project scheduling

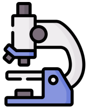
Part IV: Conclusions

Conclusions: multi-site project scheduling

Multi-site resource-constrained project scheduling

- Alternative sites for the execution of the activities
- Some resource units mobile, others non-mobile
- Transportation times between sites

- Continuous-time assignment-based MILP model
- Iterative relax-optimize-and-fix matheuristic
- Outperformance of state-of-the-art approaches



Future research

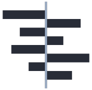
- Eliminate symmetries in feasible region of MILP
- Further analysis of benefits of resource pooling in project management

All images: Flaticon.com

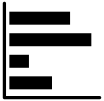
Conclusions: workload balancing

Consideration of workload balancing in project scheduling

- maxDev approach: limit deviation from average unit workload
- maxDiff approach: limit difference in workload of any two units

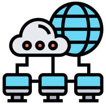


- Continuous-time assignment-based MILP model
- Workload-balanced schedules often have minimal project duration
- Workload balancing often leads to infeasibility; maxDiff more flexible



Future research

- Formulation of workload-balancing constraints as soft constraints
- Consideration of application-specific constraints on workload-balancing (e.g., green cloud computing)



All images: Flaticon.com

Bibliography I

- Artigues C., Michelon P. & Reusser S. (2003). Insertion techniques for static and dynamic resource-constrained project scheduling. European Journal of Operational Research, 149(2), 249–267
- Brucker P., Drexl A., Möhring R., Neumann K. & Pesch E. (1999). Resource-constrained project scheduling: notation, classification, models, and methods. European Journal of Operational Research, 112, 3–41.
- Christofides N., Alvarez-Valdes R. & Tamarit J.M. (1987). Project scheduling with resource constraints: A branch and bound approach. European Journal of Operational Research, 29(3), 262–273.
- Gnägi M., Zimmermann A. & Trautmann N. (2018). A continuous-time unit-based MILP formulation for the resource-constrained project scheduling problem. 2018 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), 432–436.
- Hartmann S. & Briskorn D. (2022). An updated survey of variants and extensions of the resource-constrained project scheduling problem. European Journal of Operational Research, 297(1), 1–14.
- Klein N., Gnägi M. & Trautmann N. (2024). Mixed-integer linear programming for project scheduling under various resource constraints. European Journal of Operational Research, 319(1), 79–88.
- Kolisch R. & Sprecher A. (1996). PSPLIB—a project scheduling problem library. European Journal of Operational Research, 96(1), 205–216.

Bibliography II

- Koné O., Artigues C., Lopez P. & Mongeau M. (2011). Event-based MILP models for resource-constrained project scheduling problems. Computers & Operations Research, 38(1), 3–13.
- Laurent A., Deroussi L., Grangeon N. & Norre S. (2017). A new extension of the RCPSP in a multi-site context: Mathematical model and metaheuristics. Computers & Industrial Engineering, 112, 634–644.
- Ouazene Y., Nguyen N.-Q. & Yalaoui F. (2021). Workload balancing on identical parallel machines: theoretical and computational analysis. Applied Sciences, 11(8), 3677.
- Pritsker, A., Waiters A. B., Lawrence J. & Wolfe P. M. (1969). Multiproject Scheduling with Limited Resources: A Zero-One Programming Approach. Management Science, 16(1), 93–108
- Rinaldi M., Fera M., Bottani E. & Grosse E. H. (2022). Workforce scheduling incorporating worker skills and ergonomic constraints. Computers & Industrial Engineering, 168, 108107.
- Stiti C. & Driss O. B. (2019). A new approach for the multi-site resource-constrained project scheduling problem. Procedia Computer Science, 164, 478–484.
- Walter R. & Lawrinenko A. (2014). A note on minimizing the normalized sum of squared workload deviations on m parallel processors. Computers & Industrial Engineering, 75, 257–259.