Constraint Programming applied to the Multi-Skill Project Scheduling Problem

Kenneth D. Young, Thibaut Feydy and Andreas Schutt

CP2017, 31st of August 2017

Intro: The Problem

What is the Multi-Skill Project Scheduling Problem (MSPSP)?

Intro: The Problem

What is the Multi-Skill Project Scheduling Problem (MSPSP)?

- Activities
- Workers
- Skills

Intro: The Problem

What is the Multi-Skill Project Scheduling Problem (MSPSP)?

- Activities
- Workers
- Skills

Aim: Find the fastest way to complete all the activities

Intro: The Problem

What is the Multi-Skill Project Scheduling Problem (MSPSP)?

- Activities
- Workers
- Skills

Aim: Find the fastest way to complete all the activities

Constraints

- Activity constraint: Precedence relations between activities
- Skill constraint: Activities require skills
- Worker constraint: Workers each have a variety of skills

Intro: Example

Table: Workers' Skills

	Alice	Bob	Carl	Dora
Programmer	–	✓	✓	✓
DB Designer	✓	–	–	–
Webmaster	✓	✓	–	✓

Intro: Example

Table: Workers' Skills

	Alice	Bob	Carl	Dora
Programmer	-	✓	✓	✓
DB Designer	✓	-	-	-
Webmaster	✓	✓	-	✓

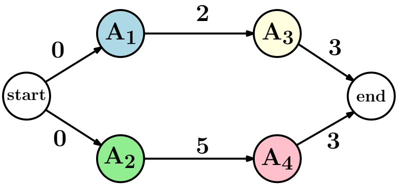

Figure: Precedence Graph

Intro: Example

Table: Workers' Skills

	Alice	Bob	Carl	Dora
Programmer	-	✓	✓	✓
DB Designer	✓	-	-	-
Webmaster	✓	✓	-	✓

Table: Skill Requirement

	A_1	A_2	A_3	A_4
Programmer	-	1	2	1
DB Designer	1	-	-	1
Webmaster	1	1	-	-

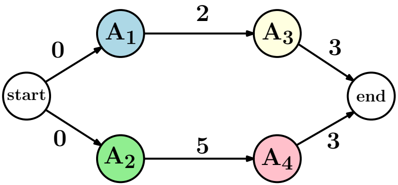

Figure: Precedence Graph

Intro: Example

Table: Workers' Skills

	Alice	Bob	Carl	Dora
Programmer	-	✓	✓	✓
DB Designer	✓	-	-	-
Webmaster	✓	✓	-	✓

Table: Skill Requirement

	A_1	A_2	A_3	A_4
Programmer	-	1	2	1
DB Designer	1	-	-	1
Webmaster	1	1	-	-

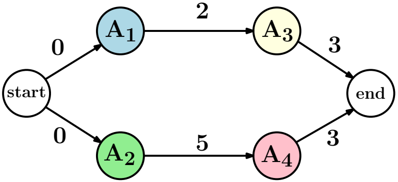

Figure: Precedence Graph

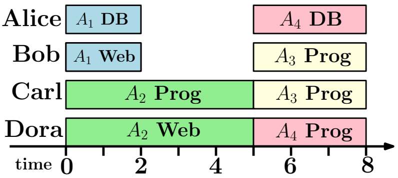

Figure: Schedule

Intro: The Literature

- French research group
  - Principal researchers: Odile Belleguez-Morineau, Emmanuel Néron, Carlos Montoya
  - Exact branch and bound methods
  - Lower bounds
  - Adapted data from PSPLib

Intro: The Literature

- French research group
  - Principal researchers: Odile Belleguez-Morineau, Emmanuel Néron, Carlos Montoya
  - Exact branch and bound methods
  - Lower bounds
  - Adapted data from PSPLib
- Portuguese research group
  - Principal researchers: Bernardo Almeida, Isabel Correia, Francisco Saldanha-da-Gama
  - Constructive heuristics
  - Randomised search heuristics
  - Generated their own data

Intro: The Literature

- French research group
  - Principal researchers: Odile Belleguez-Morineau, Emmanuel Néron, Carlos Montoya
  - Exact branch and bound methods
  - Lower bounds
  - Adapted data from PSPLib
- Portuguese research group
  - Principal researchers: Bernardo Almeida, Isabel Correia, Francisco Saldanha-da-Gama
  - Constructive heuristics
  - Randomised search heuristics
  - Generated their own data
- Polish research group
  - Principal researchers: Myszkowski, Skowronski
  - Randomised search heuristics
  - Generated their own data

Model

Model

- Objective
  - Minimise the total project duration

Model

- Objective
  - Minimise the total project duration

- Two main decisions

Model

- Objective
  - Minimise the total project duration

- Two main decisions
  1. Scheduling decisions
     - Activity start times

Model

- Objective
  - Minimise the total project duration

- Two main decisions
  1. Scheduling decisions
     - Activity start times
  2. Assignment decisions
     - Workers to activities
     - Skill contribution of workers

Model: Constraints Outline

Model: Constraints Outline

- Precedence relations are respected

Model: Constraints Outline

- Precedence relations are respected
- Workers perform only one activity at a time

Model: Constraints Outline

- Precedence relations are respected
- Workers perform only one activity at a time
- Workers cannot multi-task

Model: Constraints Outline

- Precedence relations are respected
- Workers perform only one activity at a time
- Workers cannot multi-task
- Skill requirement is satisfied
  - A worker for each skill must be present to perform the activity

Model: Constraints Outline

- Precedence relations are respected
- Workers perform only one activity at a time
- Workers cannot multi-task
- Skill requirement is satisfied
  - A worker for each skill must be present to perform the activity
- Redundant constraints

Model: Decision Variables

Decision Variables		
Primary	s_i	Start time of activity $i \in V$
	y_{ir}^s	1 iff resource $r \in R$ contributes with skill $s \in S$ to activity $i \in V$
Auxiliary	o_{ij}	1 iff activities i and j overlap for $(i, j) \in U$
	x_{ir}	1 iff resource $r \in R$ is assigned activity $i \in V$

Model: Basic Constraints

$$s_i + p_i \leq s_j \qquad \forall (i,j) \in E$$

$$\sum_{r \in R} y_{ir}^s = sr_i^s \qquad \forall i \in V, \forall s \in S$$

$$\sum_{s \in S} \sum_{i \in V : s_i \leq t < s_i + p_i} y_{ir}^s \leq 1 \quad \forall r \in R, \forall t \in \left\{0, 1, \ldots, \sum_{i \in V} p_i\right\}$$

$$y_{ir}^s \leq mast_{rs} \qquad \forall i \in V, \forall r \in R, \forall s \in S$$

Model: Redundant Constraints

$$\texttt{cumulative}(s,\ p,\ [\ sr_{is} : i \in V],\ |R_s|) \qquad \forall s \in S$$

$$\texttt{cumulative}\left(s,\ p,\ \left[\ \sum_{s \in S} sr_{is} : i \in V\ \right],\ m\right)$$

Model: Choice of Constraints

Unary Resource Constraint

- Each worker only performs one activity at a time

Model: Choice of Constraints

Unary Resource Constraint

- Each worker only performs one activity at a time

Possible ways of modelling

1. Time-indexed decomposition
2. Global constraints (either disjunctive or cumulative)
3. Order constraints

Model: Unary Resource Constr.

- Time-indexed decomposition

$$\sum_{s \in S} \sum_{i \in V : s_i \leq t < s_i + p_i} y_{ir}^{s} \leq 1 \;\; \forall r \in R, \forall t \in \left\{0, 1, \ldots, \sum_{i \in V} p_i\right\}$$

Model: Unary Resource Constr.

- Time-indexed decomposition

$$\sum_{s \in S} \sum_{i \in V : s_i \leq t < s_i + p_i} y_{ir}^{s} \leq 1 \;\; \forall r \in R, \forall t \in \left\{0, 1, \ldots, \sum_{i \in V} p_i\right\}$$

- Cumulative

$$\texttt{cumulative}((s_i)_{i \in V}, (p_i)_{i \in V}, (x_{ir})_{i \in V}, 1) \qquad \forall r \in R$$

Model: Unary Resource Constr.

- First order constraint formulation

$$\neg o_{ij} \Leftrightarrow (s_i + p_i \leq s_j) \vee (s_j + p_j \leq s_i) \quad \forall (i,j) \in U$$

$$(x_{ir} \wedge x_{jr}) \Rightarrow \neg o_{ij} \quad \forall (i,j) \in U, r \in R$$

Model: Unary Resource Constr.

- First order constraint formulation

$$\neg o_{ij} \Leftrightarrow (s_i + p_i \leq s_j) \vee (s_j + p_j \leq s_i) \quad \forall (i,j) \in U$$
$$(x_{ir} \wedge x_{jr}) \Rightarrow \neg o_{ij} \quad \forall (i,j) \in U, r \in R$$

- Second order constraint formulation

$$(o_{ij} \Rightarrow s_i + p_i \leq s_j) \wedge (\neg o_{ij} \Rightarrow s_j + p_j \leq s_i)$$
$$\forall (i,j) \in U, \exists s \in S : sr_{is} + sr_{js} > \sum\nolimits_{r \in R} mast_{rs}$$

Data: Overview

- Tested on data from the literature and generated our own data

Data: Overview

- Tested on data from the literature and generated our own data

					Best known results		
set	#instances	*n*	*l*	*m*	source	%optimal	#unsolved
1a	216	22	4	10-30	Correia et al. 2012	93.98	13
1b	216	42	4	20-60	Almeida et al. 2016	2.31	211
2a	110	20-51	2-8	5-14	Montoya et al. 2014	43.64	62
2b	77	32-62	9-15	5-19	Montoya et al. 2014	66.20	24
2c	91	22-32	3-12	4-15	Montoya et al. 2014	51.11	44

Data: Complexity Measures

1. Skill Factor
   - $SF \in \{1,\ 0.75,\ 0.5,\ \textit{variable}\}$

Data: Complexity Measures

1. Skill Factor
   - $SF \in \{1,\ 0.75,\ 0.5,\ \mathit{variable}\}$
2. Network Complexity
   - $NC \in \{1.5,\ 1.8,\ 2.1\}$

Data: Complexity Measures

1. Skill Factor
   - $SF \in \{1,\ 0.75,\ 0.5,\ \mathit{variable}\}$
2. Network Complexity
   - $NC \in \{1.5,\ 1.8,\ 2.1\}$
3. Modified Resource Strength
   - varied over 3 values
   - $$MRS = \frac{m}{\sum_{i \in V} \sum_{s \in S} sr_{is}}$$

Experiments: Search Strategies

- Basic Search
  - Start times (s_i)

Experiments: Search Strategies

- Basic Search
  - Start times (s_i)
- Sequential Searches
  - Start times (s_i), then worker assignment (x_{ir})
  - Start times (s_i), then contribution of each worker (y_{ir}^s)

Experiments: Search Strategies

- Basic Search
  - Start times (s_i)

- Sequential Searches
  - Start times (s_i), then worker assignment (x_{ir})
  - Start times (s_i), then contribution of each worker (y^s_{ir})

- Priority searches $\Rightarrow$ Group scheduling and assignment decisions of each activity together
  - `priority-ff`: choose activity group by smallest start time domain
  - `priority-sml`: choose activity group by smallest largest possible start time
  - `priority-sm`:choose activity group by smallest possible start time

Experiments: Set 1'a

- Tested on 216 generated instances with 22 activities
- Time limit of 600 seconds

Experiments: Set 1'a

- Tested on 216 generated instances with 22 activities
- Time limit of 600 seconds

unary cons.	redundant cons.	search	#nodes	%optimal	runtime
cumulative	both	default	370,174	100.00	10.23s
order-1	both	default	97,085	100.00	2.73s
order-2	both	default	54,282	100.00	1.30s

Experiments: Set 1'a

- Tested on 216 generated instances with 22 activities
- Time limit of 600 seconds

unary cons.	redundant cons.	search	#nodes	%optimal	runtime
cumulative	both	default	370,174	100.00	10.23s
order-1	both	default	97,085	100.00	2.73s
order-2	both	default	54,282	100.00	1.30s
order-2	both	`priority-ff`	41,762	100.00	1.25s
order-2	both	`priority-sml`	20,786	100.00	0.68s
order-2	both	`priority-sm`	13,241	100.00	**0.51s**

Experiments: Set 1'a

- Tested on 216 generated instances with 22 activities
- Time limit of 600 seconds

unary cons.	redundant cons.	search	#nodes	%optimal	runtime
cumulative	both	default	370,174	100.00	10.23s
order-1	both	default	97,085	100.00	2.73s
order-2	both	default	54,282	100.00	1.30s
order-2	both	`priority-ff`	41,762	100.00	1.25s
order-2	both	`priority-sml`	20,786	100.00	0.68s
order-2	both	`priority-sm`	13,241	100.00	**0.51s**
order-2	skill-res	`priority-sm`	847,879	85.19	94.81s
order-2	all-res	`priority-sm`	13,953	100.00	0.67s

Experiments: Set 1'b and Set 2

- Tested on remaining benchmark instances
- Time limit of 600 seconds

Experiments: Set 1'b and Set 2

- Tested on remaining benchmark instances
- Time limit of 600 seconds

set	#nodes	%gap	#opt	%opt	mean runtime	#closed
$1'b$	7,584k	49.3	27/216	12.5	534.6s	–
$2a$	2,223k	185.2	81/110	73.6	195.2s	$\geq$33
$2b$	816k	22.4	63/77	81.8	122.9s	$\geq$10
$2c$	14k	0.0	91/91	100.0	1.2s	44

Experiments: Complexity Measures

- Results of set $1'b$
- Time limit of 600 seconds

Experiments: Complexity Measures

- Results of set $1'b$
- Time limit of 600 seconds

measure	value	#nodes	#props	%gap	#opt	%opt	runtime(s)
SF	1	5.9m	706k	48.1	6/54	11.1	535.5
	0.75	8.9m	765k	51.8	4/54	7.4	559.9
	0.5	6.7m	914k	44.1	12/54	22.2	489.7
	variable	8.6m	754k	52.4	5/54	9.26	553.5
NC	1.5	7.7m	872k	58.3	6/72	8.3	559.9
	1.8	7.9m	812k	47.7	8/72	11.1	541.3
	2.1	7.0m	671k	41.03	13/72	18.1	502.8
MRS	#1	7.4m	1,106k	79.2	11/72	15.3	525.4
	#2	7.9m	734k	45.6	6/72	8.3	558.4
	#3	7.3m	515k	24.0	10/72	13.9	520.1
Overall		7.5m	785k	49.3	27/216	12.5	534.6

Summary

- Applied the constraint programming solver chuffed to the MSPSP

Summary

- Applied the constraint programming solver chuffed to the MSPSP
- Generated a set of benchmark instances

Summary

- Applied the constraint programming solver chuffed to the MSPSP
- Generated a set of benchmark instances
- Created an effective constraint programming model
  - Together with an application tailored search strategy

Acknowledgements

- Dr. Andreas Schutt
- Dr. Thibaut Feydy
- Adrian Goldwaser

Thanks for listening!

Questions?