

### Tractable Scheduling Problems

Geoffrey Chu, Serge Gaspers, Nina Narodytska, Andreas Schutt, **Toby Walsh To appear in Proc. of IJCAI 2013, Beijing, China** 

## + Motivation

- Scheduling an important but computationally challenging problem
  - Given: number of machines, set of jobs, each with a release time and due date, duration, resources required
  - Question: can we find start times and machines for each job to satisfy release time, due date, and resource usage constraints?

Re SIMULE Planner												×
Rufes 🗌 R	eseq			a sotaro								
-	10/07/2008	10.09/2005	10/11/200	6 10/13/2	10 0006	15/2006	10/17/2006	10/19/2008	10/21/2006	10/23/200	6 10/25/2006	10/2
	723.0713.2703.4	018.0008.2022	47 13:07 03:27 17	4708.0722.271	2.4803.0817.28	07.4822.0811	2802 48 17:090	7 2021 40 12 0002 3	16:4907:1021:3	011.5002.1016	3008.5021.1111.310	01.5116.11
Tank 3	70001C15 7901	7921 12 129	2621 02 0292628	C 876 879	92, 91	C 1129	0 020 020 026	8979 875 875	23	C 0797:00 0	029-0297 0297 0297 000	0.0505 0.25
Tank 4	7000 C S \$343	620001 C 83 838	2622	839015 4 839	203	82065	1 2 200 200 200 1	8390160 (839) 836	1240 1245 124	83973800071 68	8393 83	1908 1906 C
Tank 5								4 (0.25 10	II: 000000000	727		
Tank 6	21 C 83 836012 836	1	BODED CORCE	C III COVIC COM	abasee at a	80 <b>17 1991</b>	CMIN	838633 C				
Tank 7	0397-0	8 8 <b>6.6</b> C 4.35	0.2 0.24 0.24 0.25	602	0.56 0	0.0397 0360	<b>400 0 6397 636 0</b>	633 B	6.334	83973		
Tank 8	8 83975600		666 C 8 8 18		83974800 0			36670000 C 83867 8	839737 4	THE OWNER		
Tank 10												
Tank 11	TOP TOP		P9015406 6 71 P9	01 71 7401 74015	ext znarodal a h	ר לאת לאר	C 27 PH 79 PHC 17	128 <b>(101)</b> (1	THE PROPERTY OF	107000 7903 79	7907 70 70	
Tank 12			210 1 202 21		7160 750	2 22 2 22 2	760 7 766 7 766		1	74700 760	790 790	
Tank 13	21			ř.	C PR	C P P R P P R	PLANK PR			× 7× 1×	716 176	
Tank 14			700 71 70	C 21	· ***	7. 7. 7. 7. 7. 7.	2 7 7 2 7 7 10 F			796 790 79	796 790	
Tank 15	7901		79015400 0	7901	79075506	ת אק ה אל ל	78 7 78 7 78	TRO ISCOR	790	767000 2 79 79		
Tank 16	01 (77001) <b>10010</b>		790154 6 71 7901	71 7901	760756 27175	7 78 7 78	<b>27 1</b> 29 7 29 7 29 7 10 10 10 10 10 10 10 10 10 10 10 10 10	7 78 DECIMAN	294 P44	1767 C 79076 790	79 79 79	
Mill 1	<											2
Start Date / Time	End Date / Time	1										
10/06/2006	10/27/2006	*										

## + Motivation

- Scheduling an important but computationally challenging problem
  - NP-hard in general
  - Can we identify structural restrictions under which it becomes fixed parameter tractable?



## + Motivation

- Scheduling an important but computationally challenging problem
  - NP-hard in general
  - Can we identify structural restrictions under which it becomes fixed parameter tractable?
    - Number of processors
    - Number of start times
    - Structure of release and due dates



**.**...

[Fellows, Gaspers & Rosamond, Parameterizing by the number of numbers]

## Parameterizing by the Number of Numbers [Fellows, Gaspers, Rosamond 2010]

- Number problems
  - Subset sum
  - Partition
  - 3-Partition
  - 3-D Matching
  - **...**
- Input
  - Bag of numbers
  - Interesting parameter: number of numbers

Parameterizing by the Number of Numbers [Fellows, Gaspers, Rosamond 2010]

- Number problems
  - Subset sum
  - Partition
  - 3-Partition
  - 3-D Matching
  - **...**
- Input
  - Bag of numbers
  - Interesting parameter: number of numbers

- Often FPT
  - Often use an ILP encoding with polynomial number of vars
  - E.g. Subset sum is FPT in number of numbers being partitioned

Parameterizing by the Number of Numbers [Fellows, Gaspers, Rosamond 2010]

- Number problems
  - Subset sum
  - Partition
  - 3-Partition
  - 3-D Matching
  - **...**
- Input
  - Bag of numbers
  - Interesting parameter: number of numbers

- Often FPT
  - Often use an ILP encoding with polynomial number of vars
  - E.g. Subset sum is FPT in number of numbers being partitioned
- Inspired perhaps a little by [Bessiere et al AAAI 2008]?
  - Propagating global constraint like NVALUE is NP-hard but FPT in number of numbers in domains

# + Outline of this talk

- 4 case studies
  - 3 positive (FPT algorithms)
  - I negative (NP-hard even under strong structural restrictions)
- Previous work limited
  - [Marx 2011] observed limited work on FPT algorithms for scheduling
  - One exception where parameter is tree width of precedence graph and (#late tasks or #tasks on time)

# + Global constraints in scheduling

CUMMULATIVE

INTER DISTANCE

## + Global constraints in scheduling

#### CUMMULATIVE

- Introduced in CHIP
- Each task has release time, length, due date and resource usage
- Can we schedule tasks so tasks execute or or after release time, finish before due dates without exceeding resource capacity?
  - NP-hard to enforce domain consistency

#### INTER DISTANCE

# Global constraints in scheduling

#### CUMMULATIVE

- Introduced in CHIP [1993]
- Each task has release time, length, due date and resource usage
- Can we schedule tasks so tasks execute or or after release time, finish before due dates without exceeding resource capacity?
  - NP-hard to enforce domain consistency

- INTER DISTANCE
  - Introduced in [Regin 1997]
  - Scheduling equal length tasks on a single machine
  - Each task has a set of possible start times
  - Generalizes AllDifferent
    - $|\mathbf{S}_i \mathbf{S}_j| \ge 1$
    - $\bullet |S_i S_j| \ge m$

# Bounded task types, single machine

- Suppose tasks divide into a small number of types
  - Within each type, same release times, due dates, length, and precedences
- THM: Checking consistency of CUMMULATIVE is FPT in number of task types
  - Proof: Divide problem into blocks (periods with no release times or due dates). Within each block, we can put all tasks of the same type together into a "run". Then construct ILP with polynomial number of vars

Sij = start time in the ith block of the run of task type j

Xij = number of repititions of task type j in ith block

# Bounded task types, multiple machines

- Suppose tasks divide into a small number of types
  - Within each type, same release times, due dates, length, and precedences
  - Each task requires a single machine
- THM: Checking consistency of CUMMULATIVE is FPT in number of task types + number of processors
  - Proof: Similar ILP constructed but now with another index for machine

## + Nested task types, single machine

- Suppose tasks are nested
  - Think Russian doll
  - $r_1 < r_2 < ... < r_m < d_m < ... < d_2 < d_1$
  - E.g. take item apart then put it back together
- THM: Checking consistency of CUMMULATIVE is FPT in number of task types
  - Proof: Can push all tasks to left, build ILP with polynomial number of vars encoding such solutions
  - Corrects [Braind et al 2006] who claim this is NP-hard

# Pairwise overlapping start intervals

- Start interval of task =  $[s_i, d_i-l_i]$
- For any time point, t let S[t] be number of tasks whose start interval contains t
- Let k = max value of S[t] across all time points
  - Related to well known disjunctive ratio [Baptiste & Pappe 2000]

# Pairwise overlapping start intervals

- Start interval of task =  $[s_i, d_i-l_i]$
- For any time point, t let S[t] be number of tasks whose start interval contains t
- Let k = max value of S[t] across all time points
- **THM:** Checking consistency of CUMMULATIVE is FPT in k
  - Proof: Dynamic program over all possible subsets of tasks. Only polynomial number need be examined due to problem constraints.

## + InterDistance constraint

- Interesting special case of CUMMULATIVE
  - Single machine
  - All tasks of same length
- Polynomial cases
  - Start times are intervals [Artiouchine & Baptiste 2007]
  - Task length = 1 [Regin 1994]

# InterDistance constraint

- Interesting special case of CUMMULATIVE
  - Single machine
  - All tasks of same length
- Polynomial cases
  - Start times are intervals [Artiouchine & Baptiste 2007]
  - Task length = 1 [Regin 1994]
- THM: Checking consistency of InterDistance is NP-hard even if task lengths = 2 and start times contain at most two intervals

## + InterDistance constraint

- Interesting special case of InterDistance
  - Start times are all of form  $\{s_i, s_i+h, s_i+2h, ..., s_i+k_ih\}$
  - E.g. aircraft landing times!
- THM: Checking consistency of InterDistance is NP-hard even if k<sub>i</sub> in {1,k}

## + InterDistance constraint

- Interesting special case of InterDistance
  - Start times are all of form {s<sub>i</sub>, s<sub>i</sub>+h, s<sub>i</sub>+2h, ..., s<sub>i</sub>+k<sub>i</sub>h}
  - E.g. aircraft landing times!
- THM: Checking consistency of InterDistance is NP-hard even if k<sub>i</sub> in {1,k}
  - But linear when k<sub>i</sub> = k

# + Conclusions

- We can exploit structural properties of scheduling to identify tractable cases
  - E.g. CUMMULATIVE with bounded task types
  - E.g. CUMMULATIVE with bounded number of nested task types
  - E.g. CUMMULATIVE with bounded number of pairwise overlapping start intervals
- But some scheduling problems resist such analysis
  - InterDistance constraint with start times of form s<sub>i</sub>+j.h
- Open question
  - What other common structural parameters give tractability?



PS I'm running a summer school on optimsation/constraint



http://www.AiAccess.org/





