Technische Universität Braunschweig

A Student Scheduling System for Federal Law Enforcement Training Centers (FLETC)

Frederik Fiand FrederikFiand@googlemail.com

Diploma Thesis at the Institute for Mathematical Optimization, TU Braunschweig

Advised by
Prof. Dr. Uwe Zimmermann
Dr. Ronny Hansmann

GOR

GOR-Meeting - Hybrid Methods
Bad Honnef, November 2012

Outline

- What is FLETC
- The Scheduling Problem – A Minimum Example
- The Real World Problem
- A Monolithic Model
- A Heuristic Solution Approach
- Computational Results
- Outlook

FLETC: Federal Law Enforcement Training Center

Source: http://www.fletc.gov/

The Scheduling Problem

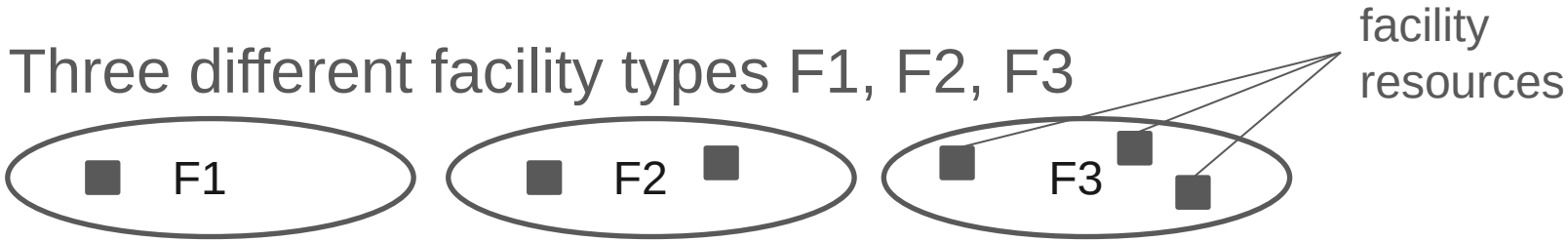
- Different training programs
- One class gets through one program
- Classes start non-periodic
- Precedence relations between courses
- Facility requirements of courses (Bottleneck!)
- Predefined schedules for all programs

→ Task: Identify conflicts caused by predefined schedules and resolve as much as possible.

→ A minimum example

A Minimum Example

- Three different facility types F1, F2, F3



- Two different programs available

p1

1	2	3	4
C1 F3		C7 F2	C11 F2
	C4 F2	C8 F2	C12 F2
C2 F1	C5 F1	C9 F1	C13 F2
C3 F3	C6 F2	C10 F3	

p2

1	2	3	4
Ca F1		Cg F3	Cj F2
	Cd F1	Ch F2	Ck F1
Cb F1	Ce F2	Ci F2	
Cc F2	Cf F3		Cl F3

→ Generate Master Schedule with start dates

- Three classes: class1, class2, class3
- Identify conflicts

p1

1	2	3	4
C1 F3		C7 F2	C11 F2
	C4 F2	C8 F2	C12 F2
C2 F1	C5 F1	C9 F1	C13 F2
C3 F3	C6 F2	C10 F3	

p2

1	2	3	4
Ca F1		Cg F3	Cj F2
	Cd F1	Ch F2	Ck F1
Cb F1	Ce F2	Ci F2	
Cc F2	Cf F3		Cl F3

Class1 starts on monday

Class3 starts on wednesday

Class2 starts on tuesday

Master Schedule

Mon	Tue	Wed	Thu	Fri	…
C1 F3		C7 F2	C11 F2		
	Ca F1		Cg F3	Cj F2	…
		C1 F3		C7 F2	
	C4 F2	C8 F2	C12 F2		
		Cd F1	Ch F2	Ck F1	…
			C4 F2	C8 F2	
C2 F1	C5 F1	C9 F1	C13 F2		
	Cb F1	Ce F2	Ci F2		…
		C2 F1	C5 F1	C9 F1	
C3 F3	C6 F2	C10 F3			
	Cc F2	Cf F3		Cl F3	…
		C3 F3	C6 F2	C10 F3	

- Resolve conflicts

Master Schedule

Mon	Tue	Wed	Thu	Fri
C1 F3		C7 F2	C11 F2	
	Ca F1		Cg F3	Cj F2
		C1 F3		C7 F2
	C4 F2	C8 F2	C12 F2	
		Cd F1	Ch F2	Ck F1
			C4 F2	C8 F2
C2 F1	C5 F1	C9 F1	C13 F2	
	Cb F1	Ce F2	Ci F2	
		C2 F1	C5 F1	C9 F1
C3 F3	C6 F2	C10 F3		
	Cc F2	Cf F3		Cl F3
		C3 F3	C6 F2	C10 F3

First conflict:
Precedence Rules for course Cb
→ Easy to solve

Second conflict:
→ More than one course needs to be moved

Third conflict:
→ Switch of slots possible

- Resolve conflicts

Master Schedule

Mon	Tue	Wed	Thu	Fri
C1 F3		C7 F2	C11 F2	
	Ca F1	Cd F1	Ch F2	Cj F2
		C1 F3		C7 F2
	C4 F2	C8 F2	C12 F2	
	Cb F1		Cg F3	Ck F1
		C2 F1	C4 F2	C8 F2
C2 F1	C5 F1	C9 F1	C13 F2	
		Ce F2	Ci F2	
			C5 F1	C9 F1
C3 F3	C6 F2	C10 F3		
	Cc F2	Cf F3		Cl F3
		C3 F3	C6 F2	C10 F3

All conflicts resolved!

A Selection of Business Rules

- Precedence Rules
  - determining order
  - determining minimum and maximum distance
- Facility Requirements
  - courses require one or multiple facilities
- Course Properties
  - different course durations
  - classes might be subdivided into groups for some courses
  - fixed slots for some courses
- Exclusive Courses and Facilities
  - some courses are not allowed to run simultaneously across classes
  - some facilities cannot be used simultaneously
- Fixed Start and End Date for Each Class

Real World Problem

- Minimize number of conflicts
  - ⚠ conflicts are allowed but penalized!
- Fulfill all business rules
- Field of application
  - **long term planning**
  - add/remove classes to/from existing schedule on short notice
  - analyze the facility situation (too less/much of a certain kind)

IP Formulation

$e \in E$ Events

$t \in T$ Time Slots

$f \in F$ Facility Types

- Schedule all events at an appropriate time slot and facility type with a minimum number of conflicts!
- Penalize movement of courses
  - all possible movements are cheaper than a single conflict
  - the further the movement the higher the penalty
  - **decreases solution time**

$$
\begin{aligned}
min \quad & \sum_{t\in T}\sum_{f\in F} O_{t,f} + \sum_{t\in T}\sum_{c^{ex}\in C^{ex}} O^{ex}_{t,c^{ex}} + \sum_{t\in T}\sum_{f^{ex}\in F^{ex}} O^{ex}_{t,f^{ex}} \\
& + \sum_{e\in E}\sum_{\{t,t'\}\in T_e\times T_e} X_{e,t'} \cdot movepen_{t,t'} \cdot \mathbb{1}_{\{e,t\}\in X^{start}} \\
s.t. \quad & \sum_{\{g'\in G|g'\succ g\}}\sum_{e\in E_{g'}}\sum_{t'\in T^{occ}_{e,t}} X_{e,t'} \le 1 && \forall g\in G^{basic}, t\in T \\
& \sum_{t\in T_e} X_{e,t} = 1 && \forall e\in E \\
& \sum_{t\in T}\sum_{f\in F_{fg}} Y_{e,t,f} = req_{e,fg} && \forall e\in E, fg\in FG_e \\
& \sum_{e\in E_f}\sum_{t'\in T^{occ}_{e,t}} Y_{e,t',f} \le n_f + O_{t,f} - down_{t,f} && \forall t\in T, f\in F_t \\
& X_{e,t} = 1 && \forall \{e,t\}\in X^{fix} \\
& X_{e,t}\cdot \sum_{fg\in FG_e} req_{e,fg} = \sum_{f\in F_e} Y_{e,t,f} && \forall e\in E, t\in T_e \\
& \sum_{\{e\in E_e|e\in c^{ex}\}}\sum_{t'\in T^{occ}_{e,t}} X_{e,t'} \le 1 + O^{ex}_{t,c^{ex}} && \forall c^{ex}\in C^{ex}, t\in T \\
& \sum_{f\in f^{ex}}\sum_{e\in E_f}\sum_{t'\in T^{occ}_{e,t}} Y_{e,t',f} \le 1 + O^{ex}_{t,f^{ex}} && \forall f^{ex}\in F^{ex}, t\in T \\
& X_{e_1,t} = X_{e_2,t} && \forall e^{sbs}\in E^{sbs}, t\in T_{e^{sbs}} \\
& X_{e_1,t} = X_{e_2,t+dur_{e_1}} && \forall e^{btb}\in E^{btb}, t\in T_{e_1} \\
& \sum_{\substack{\{t'\in T|pd_t-max\le pd_{t'}\le pd_t-min,\\ t'<t,\ t'\notin T^{occ}_{e_1,t}\}}} X_{e_1,t'} \ge X_{e_2,t} && \forall p\in P, t\in T_{e_2} \\
& \sum_{e\in E^{dmd}_{cid}}\sum_{\{t\in T_e|block_t<5\}} X_{e,t} \le M\cdot(1-DMD_{cid}) && \forall cid\in CID^{dmd} \\
& \sum_{e\in E^{dmd}_{cid}}\sum_{\{t\in T_e|5<block_t<10\}} X_{e,t} \le M\cdot DMD_{cid} && \forall cid\in CID^{dmd} \\
& X_{e,t} \in \{0,1\} \\
& Y_{e,t,f},\ O_{t,f},\ O^{ex}_{t,c^{ex}},\ O^{ex}_{t,f^{ex}} \in \mathbb{N}_0
\end{aligned}
$$

Question: How do we define appropriate time slots for the events?

Answer: Control the maximum deviation from the standard schedule (in days).

- **Too much:** Model blows up → No solution in acceptable time
- **Too little:** Bad solution

Results: Significant reduction of conflicts but solution is still improvable.

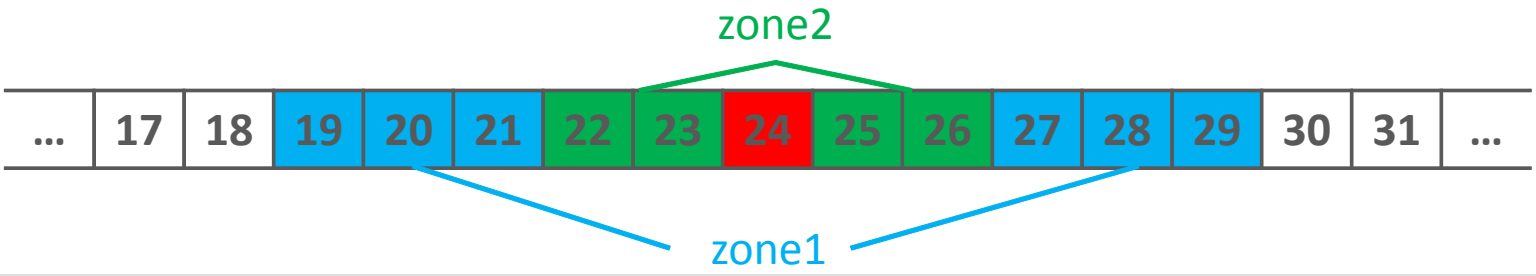
- **Idea:** Resolve conflicts only for parts of the schedule.

Resolve Algorithm

Do

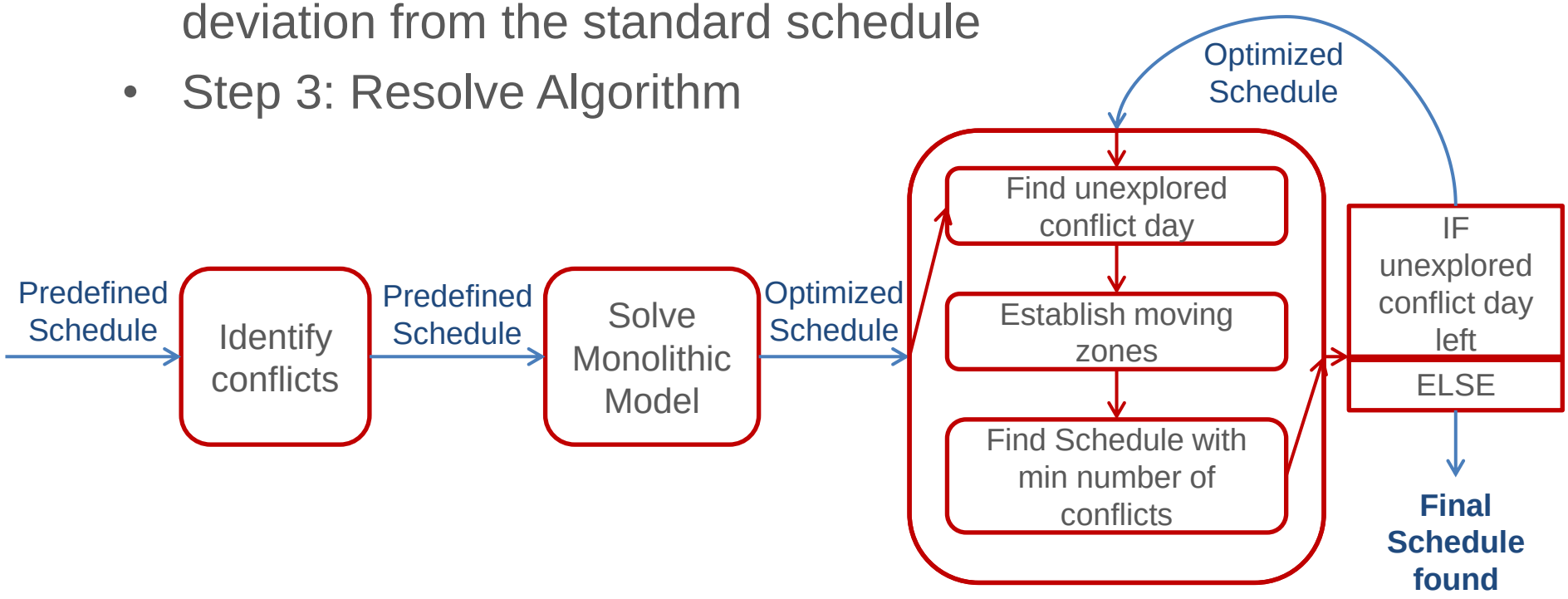
1. Search for 1st unexplored day with conflict
2. Define moving zones for classes involved in conflict
3. Find schedule with minimum number of conflicts
4. Go to step 1

Until #conflicts = 0 **or** no unexplored conflict day left
or time limit reached



A 3 Step Method

- Step 1: Identify conflicts
- Step 2: Solve Monolithic Model with a small maximum deviation from the standard schedule
- Step 3: Resolve Algorithm



Computational Results

Instance		1 (benchmark)		2		3	
Classes		9		11		13	
Courses		2298		2944		3590	
Facilty Requests		2427		3111		3795	
Precedence Req.		2385		3077		3769	
		conflicts	**time**	**conflicts**	**time**	**conflicts**	**time**
Start		530		627		858	
MM	Max. dev. = 0	253	15	389	20	556	47
	Max. dev. = 1	13	347	61	3600	332	3600
	Max. dev. = 2	4	3600	44	3600	411	3600
	Max. dev. = 3	4	3600	4	3600	556	3600
3 Step Method		0	349	0	574	8	1419

3 Step Method includes MM with MaxMove = 1 and a time limit of 180

red: time limit reached

Outlook

- Try to solve remaining conflicts, e.g. by a slightly modified Resolve algorithm:

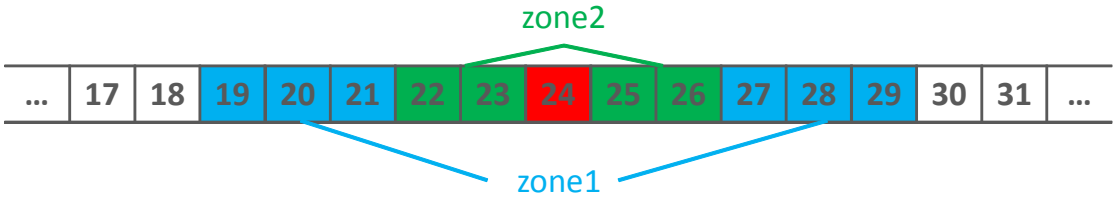
Resolve Algorithm

Do

1. Search for 1st unexplored day with conflict
2. Define moving zones for ~~classes involved in conflict~~
3. Find schedule with minimum number of conflicts
4. Go to step 1

Until #conflicts = 0 **or** no unexplored conflict day left
or time limit reached

Replace by **ALL** classes



- Use a completely different approach, e.g. a local search Algorithm (in progress)

Technische Universität Braunschweig

Thank You!

Frederik Fiand
FrederikFiand@googlemail.com

GOR

GOR-Meeting - Hybrid Methods
Bad Honnef, November 2012