Planning in Education

Some Challenging Scheduling Problems (and Some Easy Ones)

Marco Chiarandini

Department of Mathematics & Computer Science
University of Southern Denmark

EU/ME Meeting, DTU Copenhagen
Friday, May 11th, 2012

Problems Encountered

1. Balanced academic curriculum
   [with Di Gaspero, Gualandi, Schaerf, JoH, 2011]

2. Teacher enrollment in a school
   [with Bjerg, 2012]

3. Enrollment based course timetabling
   [with Weinkauff Jakobsen, 2011]

4. Enrollment based course timetabling (Elective courses) ✔
   [since 2007]

5. Project assignment ✔
   [with Gualandi and Fagerberg, CP2012 (submitted)]

6. Student Sectioning ✔

Outline

Outline

Study Curriculum

3.4	DM515 Introduktion til lineær og heltalsprogrammering	Bachelorprojekt	Tilvalg
3.3	DM516 Compilerteori		
3.2	NAT506 Videnskabsteori for datalogi	MM518 Numerisk analyse A	
3.1	MM505 Lineær algebra	DM517 Beregnelighed	

2.4	DM510 Operativsystemer	Tilvalg	Tilvalg
2.3		DM508 Algoritmer og kompleksitet	
2.2	DM509 Programmeringssprog	DM528 Kombinatorik, sandsynlighed og randomiserede algoritmer	
2.1	DM506 Maskinarkitektur	DM529 Iterativ systemudvikling	

1.4	DM507 Algoritmer og datastrukturer	NAT501 Naturvidenskabeligt projekt	
1.3		DM505 Databasedesign- og programmering	MM502 Calculus II
1.2	DM526 Introduktion til datalogi	DM503 Programmering B	MM501 Calculus I
1.1		DM502 Programmering A	DM527 Matematiske redskaber i datalogi

Academic Curriculum

[joint work with Di Gaspero, Gualandi, Schaerf, 2010]

Input

- Periods

$$P := \{1 \quad 2 \quad 3 \quad 4\}$$

- Courses each with a working load=credit (eg, ECTS)

$$\begin{array}{ccc} C & := & \{A \quad B \quad C \quad D \quad E \quad F\} \\ r & := & \{5 \quad 10 \quad 5 \quad 4 \quad 20 \quad 7\} \end{array}$$

- Curricula

$$\begin{aligned} \mathcal{Q} := \{&[A, B, C, D], \\ &[B, C, D, F], \\ &[A, B, E, F]\} \end{aligned}$$

Academic Curriculum

[joint work with Di Gaspero, Gualandi, Schaerf, 2010]

Input

- Periods

$$P := \{1 \quad 2 \quad 3 \quad 4\}$$

- Courses each with a working load=credit (eg, ECTS)

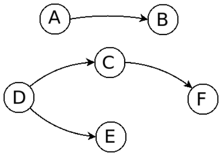
$$\begin{array}{ccllllll} C & := & \{A & B & C & D & E & F\} \\ r & := & \{5 & 10 & 5 & 4 & 20 & 7\} \end{array}$$

- Curricula

$$\mathcal{Q} := \{[A, B, C, D], [B, C, D, F], [A, B, E, F]\}$$

Constraints

- Limits to courses per periods $\{m, \dots, M\}$
- Prerequisites



precedence digraph $D = (V, A)$

Objectives

- Balance load distribution
- Avoid undesired assignments

$$U := \{(B, 3), (A, 2)\}$$

Example

Entities

- Periods

$$P := \{1 \quad 2 \quad 3 \quad 4\}$$

- Courses each with a working load=credit (eg, ECTS)

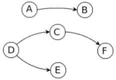
$$\begin{array}{ccccccc} C & := & \{A & B & C & D & E & F\} \\ r & := & \{5 & 10 & 5 & 4 & 20 & 7\} \end{array}$$

- Curricula

$$\begin{aligned} \mathcal{Q} := \{&[A, B, C, D], \\ &[B, C, D, F], \\ &[A, B, E, F]\} \end{aligned}$$

Constraints

- Limits to courses per periods $\{m, \ldots, M\}$
- Prerequisites



precedence digraph $D = (V, A)$

Objectives

- Balance load distribution
- Undesired assignments

$$U := \{(B, 3), (A, 2)\}$$

Example

Entities

- Periods

$$P := \{1 \quad 2 \quad 3 \quad 4\}$$

- Courses each with a working load=credit (eg, ECTS)

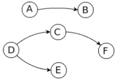
$$\begin{array}{ccllllll} C & := & \{A & B & C & D & E & F\} \\ r & := & \{5 & 10 & 5 & 4 & 20 & 7\} \end{array}$$

- Curricula

$$\begin{aligned} Q := \{&[A, B, C, D], \\ &[B, C, D, F], \\ &[A, B, E, F]\} \end{aligned}$$

Constraints

- Limits to courses per periods $\{m, \dots, M\}$
- Prerequisites



precedence digraph $D = (V, A)$

Objectives

- Balance load distribution
- Undesired assignments

$$U := \{(B, 3), (A, 2)\}$$

$$X = \begin{array}{c|cccccc} & A & B & C & D & E & F \\ \hline 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 2 & 0 & 0 & 1 & 0 & 1 & 0 \\ 3 & 1 & 0 & 0 & 0 & 0 & 1 \\ 4 & 0 & 1 & 0 & 0 & 0 & 0 \end{array}$$

$$\sigma = [3, 4, 2, 1, 2, 3]$$

Example

Entities

- Periods

$$P := \{1 \quad 2 \quad 3 \quad 4\}$$

- Courses each with a working load=credit (eg, ECTS)

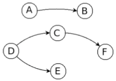
$$\begin{array}{lllllll} C & := & \{A & B & C & D & E & F\} \\ r & := & \{5 & 10 & 5 & 4 & 20 & 7\} \end{array}$$

- Curricula

$$\mathcal{Q} := \{[A, B, C, D], [B, C, D, F], [A, B, E, F]\}$$

Constraints

- Limits to courses per periods $\{m, \dots, M\}$
- Prerequisites



precedence digraph $D = (V, A)$

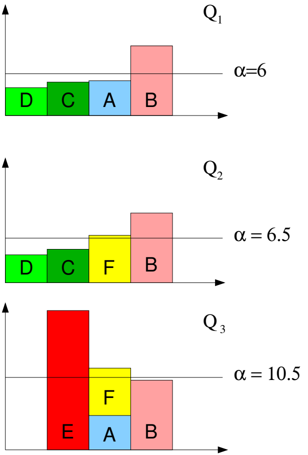
Objectives

- Balance load distribution
- Undesired assignments

$$U := \{(B, 3), (A, 2)\}$$

$$X = \begin{array}{c|cccccc} & A & B & C & D & E & F \\ \hline 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 2 & 0 & 0 & 1 & 0 & 1 & 0 \\ 3 & 1 & 0 & 0 & 0 & 0 & 1 \\ 4 & 0 & 1 & 0 & 0 & 0 & 0 \end{array}$$

$$\sigma = [3, 4, 2, 1, 2, 3]$$



Example

Entities

- Periods

$$P := \{1 \quad 2 \quad 3 \quad 4\}$$

- Courses each with a working load=credit (eg, ECTS)

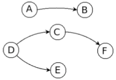
$$\begin{array}{ccllllll} C & := & \{A & B & C & D & E & F\} \\ r & := & \{5 & 10 & 5 & 4 & 20 & 7\} \end{array}$$

- Curricula

$$\mathcal{Q} := \{[A, B, C, D], \\ [B, C, D, F], \\ [A, B, E, F]\}$$

Constraints

- Limits to courses per periods $\{m, \ldots, M\}$
- Prerequisites



precedence digraph $D = (V, A)$

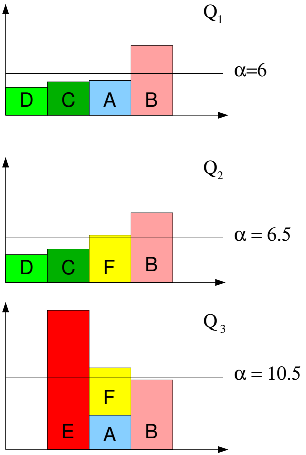
Objectives

- Balance load distribution
- Undesired assignments

$$U := \{(B, 3), (A, 2)\}$$

$$X = \begin{array}{c|cccccc} & A & B & C & D & E & F \\ \hline 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 2 & 0 & 0 & 1 & 0 & 1 & 0 \\ 3 & 1 & 0 & 0 & 0 & 0 & 1 \\ 4 & 0 & 1 & 0 & 0 & 0 & 0 \end{array}$$

$$\sigma = [3, 4, 2, 1, 2, 3]$$



Equivalent to a resource constrained project scheduling problem

Example

Entities

- Periods

$$P := \{1 \quad 2 \quad 3 \quad 4\}$$

- Courses each with a working load=credit (eg, ECTS)

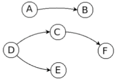
$$\begin{array}{lllllll} C & := & \{A & B & C & D & E & F\} \\ r & := & \{5 & 10 & 5 & 4 & 20 & 7\} \end{array}$$

- Curricula

$$\mathcal{Q} := \{[A, B, C, D], \\ [B, C, D, F], \\ [A, B, E, F]\}$$

Constraints

- Limits to courses per periods $\{m, \ldots, M\}$
- Prerequisites



precedence digraph $D = (V, A)$

Objectives

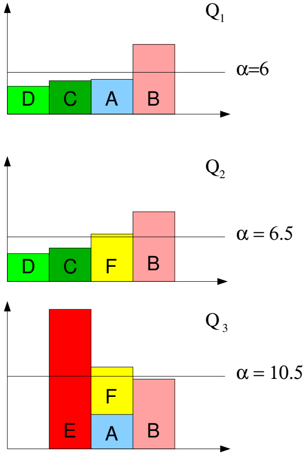
- Balance load distribution
- Undesired assignments

$$U := \{(B, 3), (A, 2)\}$$

$$X = \begin{array}{c|cccccc} & A & B & C & D & E & F \\ \hline 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 2 & 0 & 0 & 1 & 0 & 1 & 0 \\ 3 & 1 & 0 & 0 & 0 & 0 & 1 \\ 4 & 0 & 1 & 0 & 0 & 0 & 0 \end{array}$$

$$\sigma = [3, 4, 2, 1, 2, 3]$$

Equivalent to a resource constrained project scheduling problem



strongly NP-hard by reduction from 3-partition

Literature

- [Castro and Manzano, 2001] formalize the problem and include it in CSPLib with three instances.
- [Hnich, Kiziltan, and Walsh, 2002] apply CP and ILP techniques. Optimal solutions for all three instances in times ranging from 1 to hundreds seconds.
- [Lambert, Castro, Monfroy and Saubion, 2006] hybrid genetic + constraint propagation.
- [Monette, Schaus, Zampelli, Deville, Dupont, 2007] extensive empirical study on hundreds of easy solvable instances.

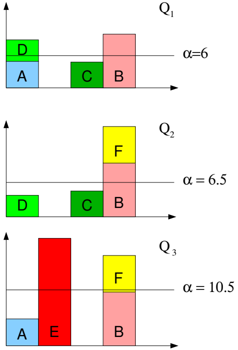
However: only one curriculum is considered and no preferences

Load Balancing Criterion

Modeled by means of norm functions

$$L_0 = \max_{p \in P} z_{Qp}$$

$$L_0 = 20$$

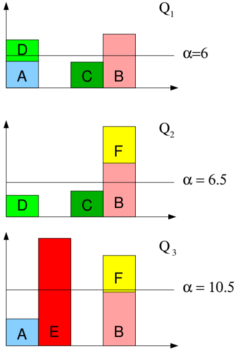


Load Balancing Criterion

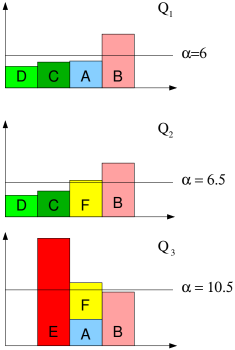
Modeled by means of norm functions

$$L_0 = \max_{p \in P} z_{Qp} \qquad L_1 = \sum_{p \in P} |z_{Qp} - \alpha(Q)|$$

$L_0 = 20$
$L_1 = 67$



$L_0 = 20$
$L_1 = 38$

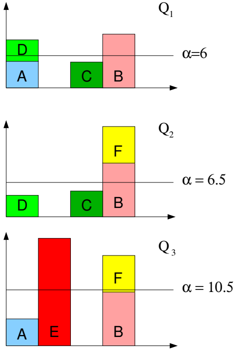


Load Balancing Criterion

Modeled by means of norm functions

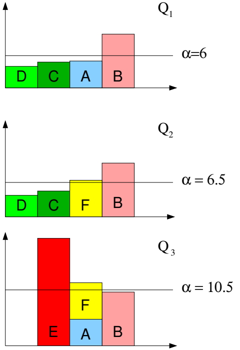
$$L_0 = \max_{p \in P} z_{Qp}$$

$L_0 = 20$
$L_1 = 67$
$L_2 = 496$



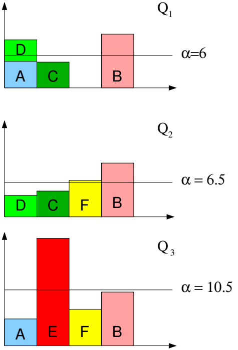
$$L_1 = \sum_{p \in P} |z_{Qp} - \alpha(Q)|$$

$L_0 = 20$
$L_1 = 38$
$L_2 = 246$



$$L_2 = \sum_{p \in P} |z_{Qp} - \alpha(Q)|^2$$

$L_0 = 20$
$L_1 = 41$
$L_2 = 216$



IP models

Objective in norm ℓ

Every course assigned

Course load limits

Prerequisites

Prerequisites

$$x_{cp} \in \{0,1\} \qquad \forall c \in C, p \in P \qquad \text{Variables} \qquad (1)$$

IP models

Objective in norm ℓ

$$\text{s.t.} \sum_{p \in P} x_{cp} = 1 \qquad \forall c \in C$$

Every course assigned

Course load limits

Prerequisites

Prerequisites

$$x_{cp} \in \{0,1\} \qquad \forall c \in C, p \in P$$

Variables

(1)

IP models

Objective in norm ℓ

$$\text{s.t.} \sum_{p \in P} x_{cp} = 1 \qquad \forall c \in C$$

Every course assigned

$$m \leq \sum_{c \in Q} x_{cp} \leq M \qquad \forall Q \in \mathcal{Q}, p \in P$$

Course load limits

Prerequisites

Prerequisites

$$x_{cp} \in \{0,1\} \qquad \forall c \in C, p \in P$$

Variables

(1)

IP models

$$
\begin{aligned}
& & & \text{Objective in norm } \ell\\
\text{s.t. } & \sum_{p \in P} x_{cp} = 1 & \forall c \in C & \quad \text{Every course assigned}\\
& m \leq \sum_{c \in Q} x_{cp} \leq M & \forall Q \in \mathcal{Q}, p \in P & \quad \text{Course load limits}\\
& \sum_{s=1}^{p-1} x_{c_1 s} \leq x_{c_2 p} & \forall [c_1, c_2] \in A, p \in P & \quad \text{Prerequisites}\\
& \sum_{p \in P} p x_{c_2 p} - \sum_{p \in P} p x_{c_1 p} \geq 1 & \forall [c_1, c_2] \in A & \\
& & & \quad \text{Prerequisites}\\
& x_{cp} \in \{0, 1\} & \forall c \in C, p \in P & \quad \text{Variables}
\end{aligned}
\qquad (1)
$$

IP models

$$
\begin{aligned}
&\min\ w_{\ell,1} L_\ell(s) + w_{\ell,2} \sum_{(c,p)\in U} x_{cp} && && \text{Objective in norm } \ell\\
&\text{s.t.}\ \sum_{p\in P} x_{cp} = 1 && \forall c \in C && \text{Every course assigned}\\
&\quad m \le \sum_{c\in Q} x_{cp} \le M && \forall Q \in \mathcal{Q}, p \in P && \text{Course load limits}\\
&\quad \sum_{s=1}^{p-1} x_{c_1 s} \le x_{c_2 p} && \forall [c_1, c_2] \in A, p \in P && \text{Prerequisites}\\
&\quad \sum_{p\in P} p x_{c_2 p} - \sum_{p\in P} p x_{c_1 p} \ge 1 && \forall [c_1, c_2] \in A && \\
& && && \text{Prerequisites}\\
&\quad x_{cp} \in \{0,1\} && \forall c \in C, p \in P && \text{Variables}
\end{aligned}
\qquad (1)
$$

Linearization

$$\min \; w_{\ell,1} L_\ell(s) + \ldots$$

Objective in norm ℓ

$$L_\ell = \sum_{Q \in \mathcal{Q}} L_{\ell,Q} = \sum_{p \in P} |z_{Qp} - \alpha(Q)|^\ell$$

➨ not linear

Linearization

$$\min \; w_{\ell,1} L_\ell(s) + \dots$$

Objective in norm ℓ

$$L_\ell = \sum_{Q \in \mathcal{Q}} L_{\ell,Q} = \sum_{p \in P} |z_{Qp} - \alpha(Q)|^\ell$$

➨ not linear

Auxiliary variables

$$y_{Qp} \geq 0 \qquad \forall Q \in \mathcal{Q}, p \in P$$

$$y_{Qp} \geq \sum_{c \in C} r(c) x_{cp} - \alpha(Q) \qquad \forall Q \in \mathcal{Q}, p \in P$$

$$y_{Qp} \geq \alpha(Q) - \sum_{c \in C} r(c) x_{cp} \qquad \forall Q \in \mathcal{Q}, p \in P$$

$$\min \quad w_{\ell,1} \sum_{p \in P} y^\ell_{Q,p} + \dots$$

Linearization

$$\min \; w_{\ell,1} L_\ell(s) + \dots$$

Objective in norm ℓ

$$L_\ell = \sum_{Q \in \mathcal{Q}} L_{\ell,Q} = \sum_{p \in P} |z_{Qp} - \alpha(Q)|^\ell$$

➨ not linear

Auxiliary variables

$$y_{Qp} \geq 0 \qquad \forall Q \in \mathcal{Q}, p \in P$$

$$y_{Qp} \geq \sum_{c \in C} r(c) x_{cp} - \alpha(Q) \qquad \forall Q \in \mathcal{Q}, p \in P$$

$$y_{Qp} \geq \alpha(Q) - \sum_{c \in C} r(c) x_{cp} \qquad \forall Q \in \mathcal{Q}, p \in P$$

$$\min \quad w_{\ell,1} \sum_{p \in P} y^\ell_{Q,p} + \dots$$

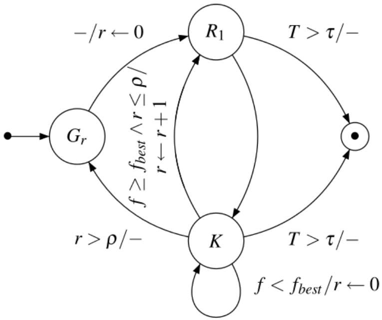
For $\ell = 1$ integer liner programming model
For $\ell = 2$ integer quadratic programming model

Local Search Approach

- Solution representation: $\sigma = [3, 4, 2, 1, 2, 3]$
- Neighborhood: one-exchange $\cup$ swap
- Evaluation function: $\sum$ # violations of load and prereq.
  + balance deviations + preferences viol.

Local Search Approach

- Solution representation: $\sigma = [3, 4, 2, 1, 2, 3]$
- Neighborhood: one-exchange $\cup$ swap
- Evaluation function: $\sum$ # violations of load and prereq.
  + balance deviations + preferences viol.
- Search strategy: R_1 runner (tabu search, simulated annealing)
  + K kicker (large neighborhood search)



Computational Results

Instances from Faculty of Engineering of University of Udine
Periods 6 to 9, Courses 140 to 300, curricula 15 to 40

Computational Results

Instances from Faculty of Engineering of University of Udine
Periods 6 to 9, Courses 140 to 300, curricula 15 to 40

Optimizing L_1 the correlation with L_2 is 0.861
Optimizing L_2 the correlation with L_1 is 0.967

Computational Results

Instances from Faculty of Engineering of University of Udine
Periods 6 to 9, Courses 140 to 300, curricula 15 to 40

Optimizing L_1 the correlation with L_2 is 0.861
Optimizing L_2 the correlation with L_1 is 0.967

Results on the quadratic model

Inst.	LB	IP UB @ 3600	IP UB @ 320	Heur. UB @ 320 (best)
UD0	31.74	320	362	55 (49)
UD1	5.51	718	2049	282 (265)
UD2	25.20	190	222	155 (148)
UD3	0.00	373	587	172 (166)
UD4	30.83	396	396	396 (396)
UD5	47.55	313	534	222 (215)
UD6	24.00	57	116	61 (57)
UD7	0.00	608	1560	237 (214)
UD8	0.00	107	107	52 (45)
UD9	5.00	624	1376	236 (221)

Open Issues

- Finding optimal solutions, faster
- Allowing heterogeneous classes: students can attend a course in different years of their curricula while still having the course taught only once per year (with large discrepancies in the academic age of students penalized) [preliminary work with Schaerf, Di Gaspero, 2010]

At SDU Nat

3. år			
4. kvartal	DM5XX Lineær og Heltalsprogrammering	Bachelorprojekt	Tilvalg
3. kvartal	DM5XX Oversætterkonstruktion		
2. kvartal	NAT506 Videnskabsteori for datalogi	Tilvalg	
1. kvartal	MM505 Lineær algebra	DM517 Beregnelighed	

2. år				
4. kvartal	DM510 Operativsystemer	DM508 Algoritmer og kompleksitet		Tilvalg
3. kvartal		DM505 Databasedesign- og programmering		
2. kvartal	DM5XX Computerarkitektur	DM5XX Introduktion til Software Engineering	DM5XX Algorithmer og sandsynlighed	
1. kvartal	DM509 Programmeringssprog			

1. år					
4. kvartal	DM507 Algoritmer og datastrukturer		NAT501 Naturvidenskabeligt projekt		
3. kvartal			DM519 Concurrent Programming		MM5XX Matematiske metoder
2. kvartal	DM526 Introduktion til datalogi	DM5XX Diskrete metoder til datalogi	DM5XX Objekt-orienteret Programmering		
1. kvartal			DM5XX Introduktion til Programmering		

At SDU Nat

3. år			
4. kvartal	DM5XX Lineær og Heltalsprogrammering	Bachelorprojekt	Tilvalg
3. kvartal	DM5XX Oversætterkonstruktion		
2. kvartal	NAT506 Videnskabsteori for datalogi	Tilvalg	
1. kvartal	MM505 Lineær algebra	DM517 Beregnelighed	

2. år				
4. kvartal	DM510 Operativsystemer	DM508 Algoritmer og kompleksitet		Tilvalg
3. kvartal		DM505 Databasedesign- og programmering		
2. kvartal	DM5XX Computerarkitektur	DM5XX Introduktion til Software Engineering	DM5XX Algorithmer og sandsynlighed	
1. kvartal	DM509 Programmeringssprog			

1. år					
4. kvartal	DM507 Algoritmer og datastrukturer		NAT501 Naturvidenskabeligt projekt		
3. kvartal			DM519 Concurrent Programming		MM5XX Matematiske metoder
2. kvartal	DM526 Introduktion til datalogi	DM5XX Diskrete metoder til datalogi	DM5XX Objekt-orienteret Programmering		
1. kvartal			DM5XX Introduktion til Programmering		

- 2020 plan: students can enter university twice a year...

Outline

School Teacher Enrollment

- A Danish school with 11 grades and about 3 sections per grade
  1–3: primary school / 4–6: middle school / 7–9 high-school
  Grades 0 and 10 are scheduled apart

How many teachers to enroll for the next year such that all teaching duties are covered?

School Teacher Enrollment

- A Danish school with 11 grades and about 3 sections per grade
  1–3: primary school / 4–6: middle school / 7–9 high-school
  Grades 0 and 10 are scheduled apart

How many teachers to enroll for the next year such that all teaching duties are covered?

- About 15 subjects, and each grade has a required number of hours per week for each subject.
- About 67 teachers who work 24 hours per week
- A teacher in DK usually teaches 2 to 4 main subjects
- Working week is made of 5 days and each day is divided into 8 time periods of one hour each. (40 periods)

School Teacher Enrollment

- A Danish school with 11 grades and about 3 sections per grade
  1–3: primary school / 4–6: middle school / 7–9 high-school
  Grades 0 and 10 are scheduled apart

How many teachers to enroll for the next year such that all teaching duties are covered?

- About 15 subjects, and each grade has a required number of hours per week for each subject.
- About 67 teachers who work 24 hours per week
- A teacher in DK usually teaches 2 to 4 main subjects
- Working week is made of 5 days and each day is divided into 8 time periods of one hour each. (40 periods)

⇝ Class, a body of students that belong to the same grade and section

Constraints

Assign teachers to subjects and classes such that competence requirements and some continuity between grades are satisfied and

Constraints

Assign teachers to subjects and classes such that competence requirements and some continuity between grades are satisfied and

from the subject perspective:

1. all required subject meetings for a class must be scheduled
2. same subject in a day has consecutive time periods

Constraints

Assign teachers to subjects and classes such that competence requirements and some continuity between grades are satisfied and

from the subject perspective:

1. all required subject meetings for a class must be scheduled
2. same subject in a day has consecutive time periods

from the teacher perspective:

3. no overlaps
4. meetings in a day must occur in consecutive time periods
5. at least 3 meetings on a day or none

Constraints

Assign teachers to subjects and classes such that competence requirements and some continuity between grades are satisfied and

from the subject perspective:

1. all required subject meetings for a class must be scheduled
2. same subject in a day has consecutive time periods

from the teacher perspective:

3. no overlaps
4. meetings in a day must occur in consecutive time periods
5. at least 3 meetings on a day or none

from the class (ie, student) perspective:

6. no overlaps
7. meetings in a day must occur in consecutive time periods
8. at least 4 meetings scheduled each day
9. other grade dependent restrictions

Example

A solution to a small case with 2 grades and 2 sections

Teacher perspective with the working periods

	M	T	W	H	F
7:	[2 3 4]	[2 3 4]	[]	[]	[]
8:	[2 3 4]	[1 2 3]	[]	[]	[]
16:	[0 1 2 3 4]	[0 1 2 3 4]	[0 1 2 3 4]	[]	[]
19:	[]	[]	[]	[]	[]
21:	[]	[]	[0 1 2 3 4]	[0 1 2 3 4]	[0 1 2 3 4]
23:	[]	[]	[]	[]	[]
27:	[2 3 4]	[1 2 3 4]	[0 1 2 3 4]	[0 1 2 3 4]	[0 1 2 3 4]
28:	[0 1 2]	[2 3 4]	[]	[0 1 2 3 4]	[0 1 2 3 4]
30:	[]	[]	[2 3 4]	[0 1 2]	[2 3 4]
51:	[]	[]	[2 3 4]	[1 2 3]	[2 3 4]

Example

The timetable for the four classes identified by (grade, section)

```
(1, 0): M {0: 'TyF', 1: 'TyF', 2: 'Dan', 3: 'Dan', 4: 'Dan'}
        T {0: 'TyF', 1: 'Dan', 2: 'Dan', 3: 'Dan', 4: 'TyF'}
        W {0: 'Eng', 1: 'TyF', 2: 'Dan', 3: 'Dan', 4: 'Dan'}
        H {0: 'Dan', 1: 'Dan', 2: 'Dan', 3: 'Eng', 4: 'Eng'}
        F {0: 'Eng', 1: 'Eng', 2: 'Dan', 3: 'Dan', 4: 'Dan'}

(1, 1): M {0: 'TyF', 1: 'TyF', 2: 'Dan', 3: 'Dan', 4: 'Dan'}
        T {0: 'TyF', 1: 'Dan', 2: 'Dan', 3: 'Dan', 4: 'Kri'}
        W {0: 'Kri', 1: 'Kri', 2: 'Mat', 3: 'Mat', 4: 'Mat'}
        H {0: 'Mat', 1: 'Mat', 2: 'Mat', 3: 'Kri', 4: 'Kri'}
        F {0: 'Kri', 1: 'Kri', 2: 'Mat', 3: 'Mat', 4: 'Mat'}

(1, 2): M {0: 'Dan', 1: 'Dan', 2: 'Dan', 3: 'Kri', 4: 'Kri'}
        T {0: 'Kri', 1: 'Kri', 2: 'Dan', 3: 'Dan', 4: 'Dan'}
        W {0: 'Kri', 1: 'Kri', 2: 'Mat', 3: 'Mat', 4: 'Mat'}
        H {0: 'Dan', 1: 'Mat', 2: 'Mat', 3: 'Mat', 4: 'Dan'}
        F {0: 'Dan', 1: 'Dan', 2: 'Mat', 3: 'Mat', 4: 'Mat'}

(2, 0): M {0: 'TyF', 1: 'TyF', 2: 'Dan', 3: 'Dan', 4: 'Dan'}
        T {0: 'TyF', 1: 'TyF', 2: 'Dan', 3: 'Dan', 4: 'Dan'}
        W {0: 'Kri', 1: 'Kri', 2: 'Mat', 3: 'Mat', 4: 'Mat'}
        H {0: 'Mat', 1: 'Mat', 2: 'Mat', 3: 'Kri', 4: 'Kri'}
        F {0: 'Kri', 1: 'Kri', 2: 'Mat', 3: 'Mat', 4: 'Mat'}

(2, 1): M {0: 'Dan', 1: 'Dan', 2: 'Dan', 3: 'TyF', 4: 'TyF'}
        T {0: 'TyF', 1: 'TyF', 2: 'Dan', 3: 'Dan', 4: 'Dan'}
        W {0: 'Eng', 1: 'TyF', 2: 'Dan', 3: 'Dan', 4: 'Dan'}
        H {0: 'Eng', 1: 'Kri', 2: 'Kri', 3: 'Kri', 4: 'Eng'}
        F {0: 'Eng', 1: 'Eng', 2: 'Kri', 3: 'Kri', 4: 'Kri'}
```

Solution approaches

- Full ILP model does not work:
  Largest instance solvable: 4 teachers, 3 classes, 4 subjects and 15 periods

Solution approaches

- Full ILP model does not work:
  Largest instance solvable: 4 teachers, 3 classes, 4 subjects and 15 periods
- Very large scale neighborhood with tabu search
  (or Logic Based Benders decomposition)

  1. enumerate teacher working patterns (dedicated algorithm in java)
  2. solve generalized set partitioning problem (gurobi)
  3. given the assignment of teachers to subjects solve the timetabling problem (IP gurobi or CP gecode)
  4. if no feasible solution goto 2 introducing constraint that avoids the same selection of patterns

Computational Tests

- Timetabling problem (step 3) still hard to solve
- Constraint relaxation (in decreasing order of difficulty to satisfy):

  1. compactness for teachers
  2. teachers have at least three periods in a row or none
  3. compactness for classes
  4. same subject in a day have consecutive periods

  Several things tried: removal, bringing into objective, iterative insertion

Solutions of reasonable quality were finally found.
Feedback from the school is still pending.

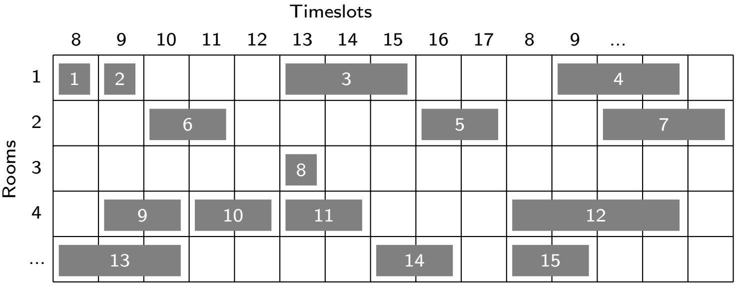
Outline

Post Enrollment Course Timetabling @ SDU Nat

Schedule classes of courses such that:

1. All classes of the quarter must be scheduled in a valid timeslot
2. Students, teachers and rooms do not have overlaps
3. Rooms meet class requirements
4. For exercise classes, students are distributed in teams of limited sizes (Student sectioning [Müller, 2010])

- classes have different durations
- different schedule between weeks
- rooms



Local Search

Design choices

1. Solution representation

   - Complete vs Partial
   - Week kern vs quarter

2. Evaluation function

3. Initial solution

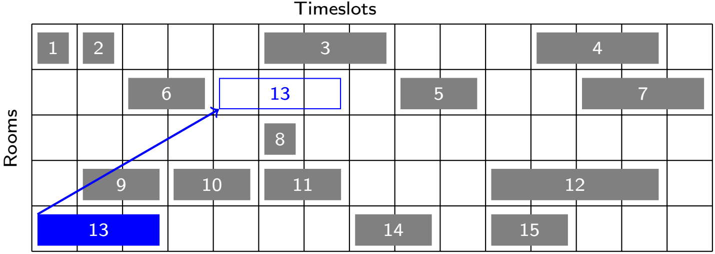
4. Neighborhood: one-exchange, swap

5. Search strategy:
   ((MinConflict heuristic with tabu) + Random Walk) + VNS

6. Termination criterion: Idle iterations

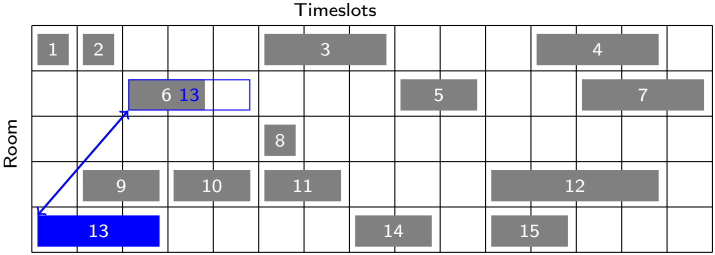
Complete State Representation
One-exchange neighborhood

Representation:
All lectures scheduled, Ev. funct: $|S_c| + 1000|P_c| + 10000|T_c|$
Neighborhood:
Move a class to a new valid, empty location in the matrix



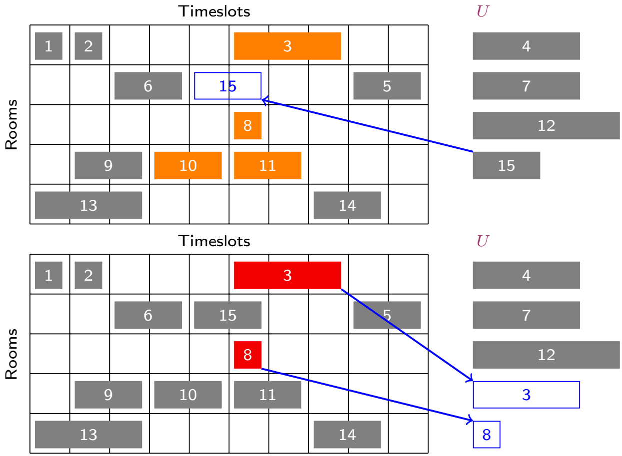
Complete State Representation
Swap neighborhood

Representation:
All lectures scheduled, Ev. funct: $|S_c| + 1000|P_c| + 10000|T_c|$
Neighborhood:
Swap with the class occupying the periods



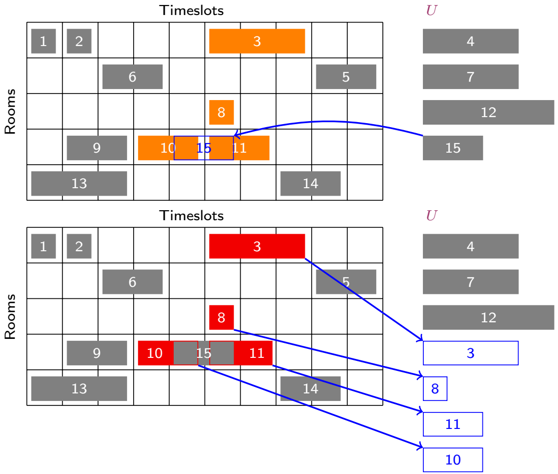
Partial State Representation
One-exchange neighborhood

Representation:
Only valid lectures scheduled, Ev. funct: $|U|$
Neighborhood:
Insert a class in an empty, valid room and time



Partial state representation
Swap neighborhood

Representation:
Only valid lectures scheduled, Ev. funct: $|U|$
Neighborhood:
Swap with the class that occupies the period



Configuration

Combined solvers changing representation:

- complete kernel
- complete quarter
- partial kernel
- partial quarter
- complete kernel + complete quarter
- complete kernel + partial kernel
- complete kernel + partial quarter
- partial kernel + partial quarter

Configuration

Combined solvers changing representation:

- complete kernel
- complete quarter
- partial kernel
- partial quarter

- complete kernel + complete quarter
- complete kernel + partial kernel
- complete kernel + partial quarter
- partial kernel + partial quarter

- Solvers use the same configurations as when run alone
- Different number of idle iterations are included in the tests

Configuration

Combined solvers changing representation:

- complete kernel
- complete quarter
- partial kernel
- partial quarter

- complete kernel + complete quarter
- complete kernel + partial kernel
- complete kernel + partial quarter
- partial kernel + partial quarter

- Solvers use the same configurations as when run alone
- Different number of idle iterations are included in the tests

- F-Race in R [Birattari et al. 2002]
- All pairwise comparisons with time limit of 2,5 minutes
- partial kernel + partial quarter was the winner

Computational Tests

Quarter	1	2	3	4
Students	1063	1013	788	717
Teachers	63	61	69	60
Rooms	36	35	28	30
Courses	90	95	102	67
Classes	294	266	274	174
# slots in total	3408	2977	3707	2346

# of classes not scheduled (% over the total number of classes)

	Partial kernel + quarter	Complete kernel	Current
1. quarter	5 (1,8%)	12 (4,4%)	96 (24,2%)
2. quarter	11 (6,3%)	18 (10,3%)	106 (34,6%)
3. quarter	7 (2,4%)	13(4,4%)	95 (25,7%)
4. quarter	1 (0,4%)	6 (2,3%)	84 (22,2%)

Outline

Course Timetabling

Given the schedule of mandatory courses, schedule classes of elective courses avoiding students, teachers and room conflicts.

Elective Courses at IMADA – Fourth Quarter – Seminarrum					
	Monday	Tuesday	Wednesday	Thursday	Friday
8:00-10:00					
10:00-12:00	DM203 *(Joan Boyar)*	DM805 *(Tao Gu)*	DM204 *(Marco Chiarandini)*	DM203 *(Joan Boyar)*	DM805 *(Tao Gu)*
12:00-14:00		DM817 *(Jørgen Bang-Jensen)*	DM825 *(Marco Chiarandini)*	DM204 *(Marco Chiarandini)*	DM817 *(Jørgen Bang-Jensen)*
14:00-16:00	DM805 *(Tao Gu)*	COLLOQUIUM	DM203 *(Joan Boyar)*	COLLOQUIUM	DM825 *(Marco Chiarandini)*
16:00-18:00	DM825 *(Marco Chiarandini)*	DM204 *(Marco Chiarandini)*	DM817 *(Jørgen Bang-Jensen)*	MM811 *(Niels Jørgen Nielsen)*	

IMADA Timetabling

Input:

- set of students S, a set of teachers T
- set of courses $C = \{1, \ldots, n\}$ each with $l(i)$ classes, $M \subset C$ mandatories, $E = C \setminus M$ electives
- collection of enrollments $\mathcal{Q} = \{Q_s \subset C | s \in S\}$ that are courses a student has subscribed (post enrollment model)
- collection of teaching duties $\mathcal{D} = \{D_t \subset C | t \in T\}$
- set of periods $P = Days \times H$ (5 days $\times$ 5 slots of two hours).
- set of rooms R (seminarrum + dummy)
- Schedule of mandatory courses M
- Teachers unavailabilites $\mathcal{U} = \{U_t \subset P \mid t \in T\}$

Integer Programming Model
Hard Constraints

Variables

$$x_{ijrdh} \in \{0,1\} \qquad \forall i \in E,\ j \in L(i)\ r \in R\ (d,h) \in P$$

Variables

$$x_{ijrdh} \in \{0,1\} \qquad \forall i \in E,\ j \in L(i)\ r \in R\ (d,h) \in P$$

H1. all classes are scheduled

$$\sum_{(d,h)\in P} \sum_{r\in R} x_{ijrdh} = 1 \qquad \forall i \in E,\ j \in L(i)$$

Integer Programming Model
Hard Constraints

Variables

$$x_{ijrdh} \in \{0,1\} \qquad \forall i \in E,\ j \in L(i)\ r \in R\ (d,h) \in P$$

H1. all classes are scheduled

$$\sum_{(d,h)\in P} \sum_{r \in R} x_{ijrdh} = 1 \qquad \forall i \in E,\ j \in L(i)$$

H2. at most one lecture in a room

$$\sum_{i \in E} \sum_{j \in L(i)} x_{ijrdh} \leq a_{rdh} \qquad \forall r \in R,\ (d,h) \in P$$

Integer Programming Model
Hard Constraints

Variables

$$x_{ijrdh} \in \{0,1\} \qquad \forall i \in E,\ j \in L(i)\ r \in R\ (d,h) \in P$$

H1. all classes are scheduled

$$\sum_{(d,h)\in P} \sum_{r\in R} x_{ijrdh} = 1 \qquad \forall i \in E,\ j \in L(i)$$

H2. at most one lecture in a room

$$\sum_{i\in E} \sum_{j\in L(i)} x_{ijrdh} \leq a_{rdh} \qquad \forall r \in R,\ (d,h) \in P$$

H3. teacher are available for the class and have at most one class at a time

$$\sum_{i\in E} \sum_{j\in L(i)} \sum_{r\in R} x_{ijrdh} \leq u_{tdh} \qquad \forall t \in T,\ (d,h) \in P$$

Integer Programming Model
Hard Constraints

H5. students have at most one class at a time

$$\sum_{i \in E} \sum_{j \in L(i)} \sum_{r \in R} x_{ijrdh} + m_{sdh} \leq 1 \qquad \forall s \in S,\ (d,h) \in P$$

Integer Programming Model
Hard Constraints

H5. students have at most one class at a time

$$\sum_{i \in E} \sum_{j \in L(i)} \sum_{r \in R} x_{ijrdh} + m_{sdh} \leq 1 \qquad \forall s \in S,\ (d,h) \in P$$

H6. at most one class of a course in a day

$$\sum_{j \in L(i)} \sum_{r \in R} \sum_{h \in H} x_{ijrdh} \leq 1 \qquad \forall i \in E, d \in Days$$

Integer Programming Model
Hard Constraints

H5. students have at most one class at a time

$$\sum_{i \in E} \sum_{j \in L(i)} \sum_{r \in R} x_{ijrdh} + m_{sdh} \leq 1 \qquad \forall s \in S,\ (d,h) \in P$$

H6. at most one class of a course in a day

$$\sum_{j \in L(i)} \sum_{r \in R} \sum_{h \in H} x_{ijrdh} \leq 1 \qquad \forall i \in E, d \in Days$$

Auxiliary variables

$$v_{td} \geq 0 \qquad \forall t \in T, d \in Days$$

if teacher has more than a class in a day

$$\sum_{i \in E} \sum_{j \in L(i)} \sum_{r \in R} \sum_{h \in H} x_{ijrdh} + \sum_{h \in H} m_{pdh} - 1 \leq v_{td} \qquad \forall t \in T, d \in Days$$

H7. at most two classes in a day for teachers

$$v_{td} \leq 1 \qquad \forall t \in T, d \in Days$$

H8. at most once in unlucky timeslots

$$\sum_{i \in E} \sum_{j \in L(i)} \sum_{r \in R} \sum_{d \in Days} (x_{ijrd8} + x_{ijrd16}) \leq 1 \qquad \forall e \in E$$

Integer Programming Model
Hard Constraints

H9. preassignments

$$x_{ijrdh} = 1 \qquad \forall i, j, r, d, h \in \text{preassigned}$$

Integer Programming Model
Objective function

Auxiliary variables

$$z_{sd} \in \{0,1\} \qquad \forall s \in S, d \in Days \quad y_{ij} \in \{0,1\} \qquad \forall t \in T, d \in Days$$

$$u', u_e \in \{0,1\} \qquad \forall e \in E$$

$z_{sd} = 1$ if student has more than three classes in a day

$$\sum_{h \in H} \sum_{i \in E} \sum_{j \in L(i)} \sum_{r \in R} x_{ijrdh} + m_{sdh} - 3 \leq z_{sd} \qquad \forall s \in S, d \in Days$$

minimum distance between classes

$$\sum_{i \in E} \sum_{j \in L(i)} \sum_{r \in R} \sum_{h \in H} x_{i,j,r,k,h} + x_{i,j,r,k,h} \leq y_{ij} + 1$$

not at 8 or at 16

$$\sum_{i \in E} \sum_{j \in L(i)} \sum_{r \in R} \sum_{d \in Days} (x_{ijrd8} + x_{ijrd16}) \leq u_e \quad \forall e \in E$$

$$u_e \leq u' \quad \forall e \in E$$

MIP Model
Objective function

$$\min \sum_{i \in E} \sum_{j \in L(i)} \sum_{dh \in P} x_{ij\mathtt{D}dh} +$$

Not in dummy room

$$+ + u' + \sum_{i \in E} u_e$$

not at 8 or at 16

$$- \sum_{i \in E} \sum_{j \in L(i)} y_{ij}$$

lectures in consec. days

$$+ \sum_{s \in S} \sum_{d \in Days} z_{sd}$$

students with > 3 class per day

$$+ \sum_{t \in T} \sum_{d \in Days} v_{td}$$

teacher with > 1 class per day

Results 2011

Quarter	3	4
Students	818	774
Teachers	14	16
Rooms	2	2
Periods	25	25
Mandatories	11	13
Classes mandatories	27	32
Elective courses	6	9
Classes	17	27

Results 2011

Quarter	3	4
Students	818	774
Teachers	14	16
Rooms	2	2
Periods	25	25
Mandatories	11	13
Classes mandatories	27	32
Elective courses	6	9
Classes	17	27

3th Quarter

```
SCIP:
Original variables    : 45927 (4957 int, 40
Original constraints : 2803
Solving Time (sec)    : 0.83
Solving Nodes         : 1
Number of Solutions   : 2
```

CPLEX even faster in 0.15 sec.

Results 2011

Quarter	3	4
Students	818	774
Teachers	14	16
Rooms	2	2
Periods	25	25
Mandatories	11	13
Classes mandatories	27	32
Elective courses	6	9
Classes	17	27

3th Quarter

```
SCIP:
Original variables    : 45927 (4957 int, 4(
Original constraints : 2803
Solving Time (sec)    : 0.83
Solving Nodes         : 1
Number of Solutions   : 2
```

CPLEX even faster in 0.15 sec.

4th Quarter

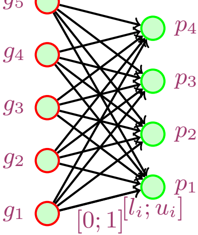
```
SCIP:
Original variables    : 44027 (5247 int, 38
Original constraints : 3241
Solving Time (sec)    : 1.28
Solving Nodes         : 1
Number of Solutions   : 4
```

CPLEX even faster in 0.18 sec.

Outline

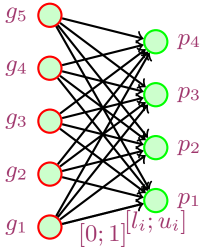
NAT501 Projects Assignment

- a set of project topics $P = \{1, \ldots, n\}$
  with a number of teams t_p for each topic $p \in P$
  Each team has cardinality bounds $[l_i..u_i]$ on the number of studetns
  (Expanision to set of project teams $\mathcal{P}$)
- a set of groups of students $G = \{g_1, \ldots, g_m\}$, $|g_i| \in [1..3]$ wishing to be in the same team



NAT501 Projects Assignment

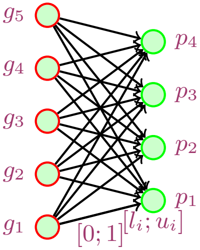
- a set of project topics $P = \{1, \ldots, n\}$
  with a number of teams t_p for each topic $p \in P$
  Each team has cardinality bounds $[l_i..u_i]$ on the number of studetns
  (Expanision to set of project teams $\mathcal{P}$)
- a set of groups of students $G = \{g_1, \ldots, g_m\}$, $|g_i| \in [1..3]$ wishing to be in the same team



Strongly NP-hard. Strongly NP-complete by reduction from 3-partition.

NAT501 Projects Assignment

- a set of project topics $P = \{1, \ldots, n\}$
  with a number of teams t_p for each topic $p \in P$
  Each team has cardinality bounds $[l_i..u_i]$ on the number of studetns
  (Expanision to set of project teams $\mathcal{P}$)
- a set of groups of students $G = \{g_1, \ldots, g_m\}$, $|g_i| \in [1..3]$ wishing to be in the same team
- for each group preferences by ranking a subset of project topics
  $r(g) = (p^{(1)}, \ldots, p^{(q_g)})$, $p^{(i)} \in P$



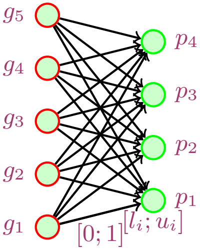
Strongly NP-hard. Strongly NP-complete by reduction from 3-partition.

NAT501 Projects Assignment

- a set of project topics $P = \{1, \ldots, n\}$
  with a number of teams t_p for each topic $p \in P$
  Each team has cardinality bounds $[l_i..u_i]$ on the number of studetns
  (Expanision to set of project teams $\mathcal{P}$)
- a set of groups of students $G = \{g_1, \ldots, g_m\}$, $|g_i| \in [1..3]$ wishing to be in the same team
- for each group preferences by ranking a subset of project topics
  $r(g) = (p^{(1)}, \ldots, p^{(q_g)})$, $p^{(i)} \in P$

Find $\sigma : G \to \mathcal{P}$ s.t. each group to exactly one project

- team bounds and group requirements are satisfied
- $\sigma(g) \in r(g)$
- fairness and collective welfare criteria are taken into account.



Strongly NP-hard. Strongly NP-complete by reduction from 3-partition.

Lottery

A greedy and fair solution: **for** priority $h = 1..\Delta$ **do**

> **for** $p \in \mathcal{P}$ **do**
>> let $C(p)$ be the set of groups with p in their hth priority
>> **for** $g \in C(p)$ in random order **do**
>>> assign g to p if it fits

Running time: $O(|G||\mathcal{P}|t'q)$, $t' = \max\{t_p \mid p \in P\}$

Lottery

A greedy and fair solution: **for** priority $h = 1..\Delta$ **do**

> **for** $p \in \mathcal{P}$ **do**
>> let $C(p)$ be the set of groups with p in their hth priority
>> **for** $g \in C(p)$ in random order **do**
>>> assign g to p if it fits

Running time: $O(|G||\mathcal{P}|t'q)$, $t' = \max\{t_p \mid p \in P\}$

It may end with:

- teams with less students than the lower bound
- unassigned groups

Lottery

A greedy and fair solution: **for** priority $h = 1..\Delta$ **do**

for $p \in \mathcal{P}$ **do**
  let $C(p)$ be the set of groups with p in their hth priority
  for $g \in C(p)$ in random order **do**
    assign g to p if it fits

Running time: $O(|G||\mathcal{P}|t'q)$, $t' = \max\{t_p \mid p \in P\}$

It may end with:

- teams with less students than the lower bound
- unassigned groups

We used:

- Integer Programming
- Constraint Programming

Handling Preferences

Quality of an assignment σ determined by:

- a value vector $\vec{v} = (v_{1,\sigma(1)}, \ldots, v_{m,\sigma(m)})$, $v_{g,\sigma(g)} > 0, \forall g \in G$
- or by the distribution of students over ranks $\vec{\delta} = (\delta_1, \ldots, \delta_\Delta)$

Handling Preferences

Quality of an assignment σ determined by:

- a value vector $\vec{v} = (v_{1,\sigma(1)}, \ldots, v_{m,\sigma(m)})$, $v_{g,\sigma(g)} > 0, \forall g \in G$
- or by the distribution of students over ranks $\vec{\delta} = (\delta_1, \ldots, \delta_\Delta)$

Individual utility groups will prefer assignments over others on the basis of their value $v_{g,\sigma(g)}$

Collective welfare: assignments that are Pareto optimal with respect to their profile vectors $\vec{v}$.

Handling Preferences

Quality of an assignment σ determined by:

- a value vector $\vec{v} = (v_{1,\sigma(1)}, \ldots, v_{m,\sigma(m)})$, $v_{g,\sigma(g)} > 0, \forall g \in G$
- or by the distribution of students over ranks $\vec{\delta} = (\delta_1, \ldots, \delta_\Delta)$

Individual utility groups will prefer assignments over others on the basis of their value $v_{g,\sigma(g)}$

Collective welfare: assignments that are Pareto optimal with respect to their profile vectors $\vec{v}$.

For two feasible assignments σ^1 and σ^2:

- classical utilitarian ordering: assigns a weight to each value, $w : [1..\Delta] \to \mathbb{Z}^+$ and compares $\sum_g w(v_{g,\sigma^1(g)})$ with $\sum_g w(v_{g,\sigma^2(g)})$ here joined with minimax $\min\max\{v_{g,\sigma(g)} \mid g \in G\}$

- egalitarian ordering: leximin order, which consists in reordering the two vectors $\vec{v}^1$ and $\vec{v}^2$ by increasing coordinates and comparing them lexicographically.

Minimax

$$
\begin{aligned}
\min \quad & f \\
\text{s.t.} \quad & \sum_{p \in \mathcal{P}_g} x_{gp} = 1, && \forall g \in G \\
& \sum_{g \in G} |g| x_{gp} \leq u_p y_p, && \forall p \in \mathcal{P} \\
& \sum_{g \in G} |g| x_{gp} \geq l_p y_p, && \forall p \in \mathcal{P} \\
& f \geq v_{gp} x_{gp}, && \forall g \in G, \forall p \in \mathcal{P} \\
& x_{gp} \in \{0,1\}, && \forall g \in G, \forall p \in \mathcal{P} \\
& y_p \in \{0,1\}, && \forall p \in \mathcal{P} \\
& f \geq 0
\end{aligned}
$$

Weighted

$$
\begin{aligned}
\max \quad & \sum_{g} w(v_{g,\sigma^2(g)}) - Mf \\
\text{s.t.} \quad & \sum_{p \in \mathcal{P}_g} x_{gp} = 1, && \forall g \in G \\
& \sum_{g \in G} |g| x_{gp} \leq u_p y_p, && \forall p \in \mathcal{P} \\
& \sum_{g \in G} |g| x_{gp} \geq l_p y_p, && \forall p \in \mathcal{P} \\
& f \geq v_{gp} x_{gp}, && \forall g \in G, \forall p \in \mathcal{P} \\
& x_{gp} \in \{0,1\}, && \forall g \in G, \forall p \in \mathcal{P} \\
& y_p \in \{0,1\}, && \forall p \in \mathcal{P} \\
& f \geq 0
\end{aligned}
$$

Lexicographic procedure

Data: a problem instance Π
Result: a leximin optimal solution X^* to Π
$h' = \Delta$;
$N = \vec{0}$
while $h' > 1$ **do**
    (N^*,X^*)=`{cp|ip}_model`(Π,N^*,h');
    $h' = h' - 1$;

Leximin

Lexicographic minimization of distributions

$$
\begin{aligned}
z^*_{h'} = \min \quad & z_{h'} \\
\text{s.t.} \quad & \sum_{p \in \mathcal{P}_g} x_{gp} = 1, && \forall g \in G \\
& \sum_{g \in G} |g| x_{gp} \leq u_p y_p, && \forall p \in \mathcal{P} \\
& \sum_{g \in G} |g| x_{gp} \geq l_p y_p, && \forall p \in \mathcal{P} \\
& z^*_h = \sum_{(g,p) \in R_h} |g| x_{gp}, && \forall h \in [\Delta..(h'+1)] \\
& z_{h'} = \sum_{(g,p) \in R_{h'}} |g| x_{gp} \\
& x_{gp} \in \{0,1\}, && \forall g \in G, \forall p \in \mathcal{P} \\
& y_p \in \{0,1\}, && \forall p \in \mathcal{P} \\
& z_{h'} \geq 0
\end{aligned}
$$

Constraint Programming

- W_p subset of groups that are assigned to project p; $\mathtt{dom}(W_p) \subseteq G$
- X_g project p to which group g is assigned; $\mathtt{dom}(X_g) = \mathcal{P}_g$
- Y_p is 1 if project p has assigned at least one group, 0 otherwise;
- Z_p number of students assigned to project p
- $\mathbf{a}$ vector of $|G|$ elements that gives $|g|$, $\forall g \in G$

(1) $\forall p \in \mathcal{P}. \quad X_g = p \iff g \in W_p$ [channel]

(2) $\forall p \in \mathcal{P}. \quad Z_p = \text{weights}(W_p, \mathbf{a})$ [weights]

(3) $\forall p \in \mathcal{P}. \quad Z_p \in \{0, l_p..u_p\}$ [dom]

Constraint Programming

- W_p subset of groups that are assigned to project p; $\texttt{dom}(W_p) \subseteq G$
- X_g project p to which group g is assigned; $\texttt{dom}(X_g) = \mathcal{P}_g$
- Y_p is 1 if project p has assigned at least one group, 0 otherwise;
- Z_p number of students assigned to project p
- $\mathbf{a}$ vector of $|G|$ elements that gives $|g|$, $\forall g \in G$

(1) $\forall p \in \mathcal{P}. \quad X_g = p \iff g \in W_p$ [channel]
(2) $\forall p \in \mathcal{P}. \quad Z_p = \text{weights}(W_p, \mathbf{a})$ [weights]
(3) $\forall p \in \mathcal{P}. \quad Z_p \in \{0, l_p..u_p\}$ [dom]

- V_g value obtained by group g under assignment X_g; $\texttt{dom}(V_g) = [1..\Delta]$
- U_h subset of groups that obtain their preference h; $\texttt{dom}(U_h) \subseteq G$
- N_h number of students that obtain their preference h; $\texttt{dom}(N_h) = [1..m]$

(4) $\forall g \in G. \quad V_g = v^{-1}_{X_g}(g)$ [element]
(5) $\forall h \in H. \quad V_g = h \iff g \in U_h$ [channel]
(6) $\forall h \in H. \quad N_h = \text{weights}(U_h, \mathbf{a})$ [weights]

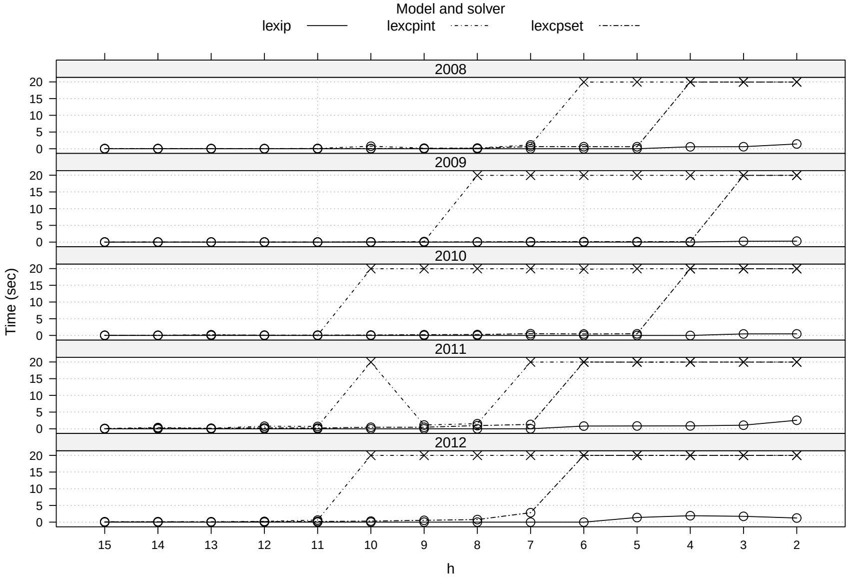
Branch on S_p choosing the variable with smallest domain size divided by accumulated failure count and including largest element.

Computational Results

Instance					Criterion	Students per priorities														Unassigned	Underfull	
year	$\lvert S\rvert$	$\lvert G\rvert$	$\lvert\mathcal{P}\rvert$	$\lvert P\rvert$		14	13	12	11	10	9	8	7	6	5	4	3	2	1	students	projects	sec.
2008	200	173	70	52	lottery	0	0	0	0	0	0	0	6	8	9	9	8	21	135	4	4	0.0
2008	200	173	70	52	minimax	0	0	0	0	0	0	0	0	0	0	12	79	64	45	0	0	0.6
2008	200	173	70	52	leximin	0	0	0	0	0	0	0	0	0	0	2	38	69	91	0	0	4.7
2008	200	173	70	52	weighted	0	0	0	0	0	0	0	0	0	0	15	22	46	117	0	0	0.0
2009	129	107	48	45	lottery	0	0	0	0	0	0	0	0	3	4	2	0	10	108	2	0	0.0
2009	129	107	48	45	minimax	0	0	0	0	0	0	0	0	0	0	0	32	46	51	0	0	0.4
2009	129	107	48	45	leximin	0	0	0	0	0	0	0	0	0	0	0	5	31	93	0	0	1.2
2009	129	107	48	45	weighted	0	0	0	0	0	0	0	0	0	0	0	8	24	97	0	0	0.0
2010	193	158	62	52	lottery	0	0	2	0	0	0	0	0	0	3	3	7	23	147	8	1	0.0
2010	193	158	62	52	minimax	0	0	0	0	0	0	0	0	0	0	0	70	64	59	0	0	0.8
2010	193	158	62	52	leximin	0	0	0	0	0	0	0	0	0	0	0	20	56	117	0	0	2.0
2010	193	158	62	52	weighted	0	0	0	0	0	0	0	0	0	0	0	26	43	124	0	0	0.0
2011	259	219	83	69	lottery	0	0	4	0	2	2	3	2	4	6	8	9	28	171	20	3	0.0
2011	259	219	83	69	minimax	0	0	0	0	0	0	0	0	30	59	47	50	40	33	0	0	2.2
2011	259	219	83	69	leximin	0	0	0	0	0	0	0	0	5	17	23	22	77	115	0	0	9.9
2011	259	219	83	69	weighted	0	0	0	0	0	0	0	0	17	15	16	10	51	150	0	0	0.0
2012	300	247	102	81	lottery	0	0	0	0	0	2	4	6	5	8	12	22	29	188	24	4	0.0
2012	300	247	102	81	minimax	0	0	0	0	0	0	0	0	0	59	66	61	59	55	0	0	2.7
2012	300	247	102	81	leximin	0	0	0	0	0	0	0	0	0	18	28	55	72	127	0	0	11.2
2012	300	247	102	81	weighted	0	0	0	0	0	0	0	0	0	32	32	34	45	157	0	0	0.0

- lottery clearly outperformed
- leximin outperforms minimax
- weighted has many in first but also many in last priority

CP vs IP


Model and solver
lexip
lexcpint
lexcpset
2008
2009
2010
2011
2012
Time (sec)
20
15
10
5
0
15
14
13
12
11
10
9
8
7
6
5
4
3
2
h

Open Issues

- Disruption management: reassignments with minimal changes
- Partial order in preference expression
- What to tell to the students??

Summary

- Several combinatorial problems all year around:
  some are well solved by CP and IP (consider even Solver in Excel),
  some are hard and need heuristics
- Heuristics lack of a modelling framework á la COMET
  $\rightsquigarrow$ hard to apply in practice
- Interesting link with collective welfare and social choice theory
- Good case studies to engage students in education of IP, CP, LS

Thank you for your attention

Planning in Education

Some Challenging Scheduling Problems (and Some Easy Ones)

Marco Chiarandini

Department of Mathematics & Computer Science
University of Southern Denmark

EU/ME Meeting, DTU Copenhagen
Friday, May 11th, 2012