

Timetabling at Purdue University

April 21, 2010

Part III: Interactive Timetabling

Suggestions

Changes with class "PSY 120 Lec 5" are considered

Suggestions

<u>Score</u>	<u>Class</u>	<u>Date</u>	<u>Time</u>	<u>Room</u>	<u>Students</u>
+43	PSY 120 Lec 5	Full Term	MWF 7:30a	WTHR 200 → CL50 224	0
+48.4	PSY 120 Lec 5	Full Term	MWF 7:30a → TTh 7:30a	WTHR 200 → CL50 224	+10
+63.3	PSY 120 Lec 5	Full Term	MWF 7:30a → MWF 4:30p	WTHR 200 → LILY 1105	+14
	POL 130 Lec 2	Full Term	MWF 4:30p → MWF 9:30a	LILY 1105 → RHPH 172	
+63.9	PSY 120 Lec 5	Full Term	MWF 7:30a → MWF 4:30p	WTHR 200 → LILY 1105	+16
	POL 130 Lec 2	Full Term	MWF 4:30p	LILY 1105 → FRNY G140	
+63.9	PSY 120 Lec 5	Full Term	MWF 7:30a → MWF 4:30p	WTHR 200 → LILY 1105	+16
	POL 130 Lec 2	Full Term	MWF 4:30p	LILY 1105 → LYNN 1136	

(all 235 possibilities up to 2 changes were considered, top 5 of 22 suggestions displayed)

See http://www.unitime.org/uct_demo.php for online demo

Interaction Process: Variables

Timetabling problem $P = (V, \mathcal{D}, C, w_c, w_\theta)$

- weighted constraint satisfaction problem

Initial solution δ

initial timetable of the interaction process

Selected assignments μ : changes made with the timetable δ
during current interaction

Selected class v_{bb}

to modify its placement or to be placed into the timetable

Interaction Process: Variables

Timetabling problem $P = (V, \mathcal{D}, C, w_c, w_\theta)$

- weighted constraint satisfaction problem

Initial solution δ

initial timetable of the interaction process

Selected assignments μ : changes made with the timetable δ
during current interaction

Selected class v_{bb}

to modify its placement or to be placed into the timetable

Suggestions Ω : set of generated assignments ω

making the timetable feasible (all hard constraints are satisfied)

Conflicting assignments γ

set of assignments conflicting with selected assignments μ

Simplified Interaction Process

```
procedure INTERACTION( $P, \delta, v_{bb}$ )  
   $\mu = \emptyset$   
   $A = \emptyset$   
  while true do  
     $\Omega = \text{BB}(P \cup A, \delta, \mu, v_{bb})$ 
```

Simplified Interaction Process

procedure INTERACTION(P, δ, v_{bb})

$\mu = \emptyset$

$A = \emptyset$

while true do

$\Omega = \text{BB}(P \cup A, \delta, \mu, v_{bb})$

$S = \text{COMMUNICATION}(\Omega)$

Simplified Interaction Process

```
procedure INTERACTION( $P, \delta, v_{bb}$ )  
   $\mu = \emptyset$   
   $A = \emptyset$   
  while true do  
     $\Omega = \text{BB}(P \cup A, \delta, \mu, v_{bb})$   
     $S = \text{COMMUNICATION}(\Omega)$   
  
    case ( $S$ ) commit( $\omega \in \Omega$ ):  $\delta = \text{join}(\delta, \mu \cup \omega)$ ; return  
      abort: return  
      selectAssignment( $d_n$ ):  $\mu = \mu \cup \{v_{bb}/d_n\}$   
      selectFilter( $\alpha$ ):  $A = \alpha v_{bb}$   
  
    end case  
  end while  
end procedure
```


Simplified Interaction Process

```
procedure INTERACTION( $P, \delta, v_{bb}$ )  
   $\mu = \emptyset$   
   $A = \emptyset$   
  while true do  
     $(\Omega, \gamma) = \text{BB}(P \cup A, \delta, \mu, v_{bb})$   
     $S = \text{COMMUNICATION}(\Omega, \gamma)$   
  
    case ( $S$ ) commit( $\omega \in \Omega$ ):  $\delta = \text{join}(\delta, \mu \cup \omega)$ ; return  
      abort: return  
      selectAssignment( $d_n$ ):  $\mu = \mu \cup \{v_{bb}/d_n\}$   
      selectFilter( $\alpha$ ):  $A = \alpha v_{bb}$   
      selectClass( $c \in \{\mu \cup \gamma \cup \Omega\}$ ):  $v_{bb} = c$   
      removeClass( $c \in \mu$ ):  $\mu = \mu \setminus \{c/d_c\}$   
  
    end case  
  end while  
end procedure
```

Variables

- weighted constraint satisfaction problem with filter $P = P \cup A$
- initial timetable δ
- selected assignments μ
- class to be (re-)placed v_{bb}

Initialization

- compute conflicting assignment caused by μ

Run BB to find assignments of variables for

- class v_{bb}
- classes involved in conflicting assignments

Branch and Bound (continues)

Run BB

- n best suggestions ω are given to user
- search with timeout
- best values based on $\Delta F_s(\delta, v/d)$ explored first
 - conflict-based statistics not taken into account (too expensive)

Bounds

- limited search depth
 - to allow changes of small number of variables only
 - to include changes of one new class
 - it does make sense to change too many other classes
 - M : maximum depth
- F_{wcsp} must be better than the n -th best found suggestion
 - $\Omega[n]$: n -th best suggestion

Branch and Bound (continues)

Run BB

- n best suggestions ω are given to user
- search with timeout
- best values based on $\Delta F_s(\delta, v/d)$ explored first
 - conflict-based statistics not taken into account (too expensive)

Bounds

- limited search depth
 - to allow changes of small number of variables only
 - to include changes of one new class
 - it does make sense to change too many other classes
 - M : maximum depth
- F_{wCSP} must be better than the n -th best found suggestion
 - $\Omega[n]$: n -th best suggestion

Repeat BB: process another run of BB with

- increased search depth or
- increased timeout

Branch & Bound

```
1: function BB( $P, \delta, \mu, v_{bb}$ )
2:   if  $\{v_{bb}/d\} \subseteq \delta$  then  $\delta = \delta \setminus \{v_{bb}/d\}$ 
3:   else  $d = nil$ 
4:    $\gamma = \{v_{bb}/d\}$ 
5:   for  $v_i/d_i \in \mu$  do
6:     if  $\{v_i/d_o\} \subseteq \delta$  then  $o = \{v_i/d_o\}$ 
7:     else  $o = \emptyset$ 
8:      $\gamma = \gamma \cup \text{hardConflicts}(P, \delta, v_i/d_i) \setminus o$ 
9:      $\delta = \delta \setminus o \cup \{v_i/d_i\}$ 
10:  end for
11:  return backtrack( $P, \delta, \mu, \gamma, \emptyset, 0$ )
12: end function
```


Experiments

Problem	pu-fal07-llr		pu-spr07-llr	
Classes	891		803	
Classes fixed in time & room (%)	31.0		33.8	
Classes not fixed in time & room (%)	69.0		66.2	
Time limit (s)	–	5	–	5
Time spent (s)	128.6	4.7	39.9	4.2
Complete space explored	98.4	21.5	99.2	33.3
No suggestion found (%)	1.6	2.3	0.8	0.8
Number of suggestions	232.8	174.9	228.6	184.5
Number of backtracks	66367.9	2886.9	13949.1	2592
Optimal suggestion found (%)	98.4	51.5	99.2	67.0
Improvements in objective function (%)	+1.1	+0.8	+0.9	+0.7