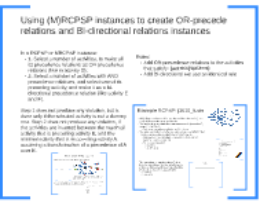
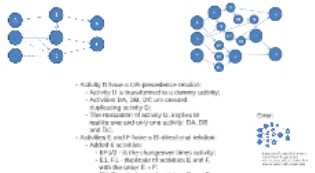
Conclusions and Future Work

- Proposed an approach to solve the RCPSP with logical constraints:
  - OR precedence relations;
  - Bi-directional precedence relations;
- The method uses an extension of multi-mode, where the realization of each activity-mode is controlled by SAT, with more flexibility over the MRCPSP procedures, and the schedule is done by a RCPSP procedure
- The method, applied to MRCPSP instances, reveal to be competitive.
- Future work we plan:
  - Study these type of logical constraints without resources, and with multiple mode;
  - Study mode identity constraints in multiple mode.



Changing of an AND-precedence relation into an OR-precedence relation, and Bi-directional relation



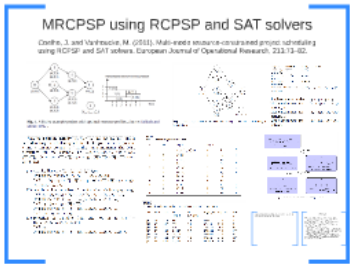
A novel approach to solve various resource-constrained project scheduling problems

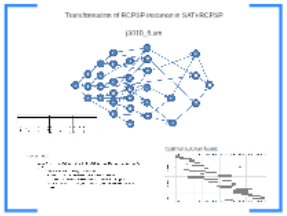
using multi-mode

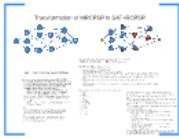
José Coelho* and Mario Vanhoucke**

* - Technical University of Lisbon, Lisbon (Portugal)
- Universidade Aberta, Lisbon (Portugal) Jose.Coelho@uab.pt
** - Faculty of Economics and Business Administration, Ghent University, (Belgium) mario.vanhoucke@ugent.be
- Technology and Operations Management Area, Vlerick Business School (Belgium) mario.vanhoucke@vlerick.com
- Department of Management Science and Innovation, University College London (United Kingdom) m.vanhoucke@ucl.ac.uk

MRCPSP using RCPSP and SAT solvers







Tests and Results

1. Run without OR-precedence relations and Bi-directional relations:
   - Check if our procedure is close to the state-of-the-art in MRCPSP instance sets
2. Run with different density of OR-precedence relations:
   - Quantify the increase needed on the number of activities
   - Quantify the gain in makespan by consider OR-precedence relations, relative to treat those precedence relations as AND-precedence relations
3. Run with bi-directional relations.

A novel approach to solve various resource-constrained project scheduling problems

using multi-mode

José Coelho* and Mario Vanhoucke**

* - Technical University of Lisbon, Lisbon (Portugal)
- Universidade Aberta, Lisbon (Portugal) Jose.Coelho@uab.pt
** - Faculty of Economics and Business Administration, Ghent University, (Belgium) mario.vanhoucke@ugent.be
- Technology and Operations Management Area, Vlerick Business School (Belgium) mario.vanhoucke@vlerick.com
- Department of Management Science and Innovation, University College London (United Kingdom) m.vanhoucke@ucl.ac.uk

MRCPSP using RCPSP and SAT solvers

Coelho, J. and Vanhoucke, M. (2011). Multi-mode resource-constrained project scheduling using RCPSP and SAT solvers. European Journal of Operational Research, 213:73–82.

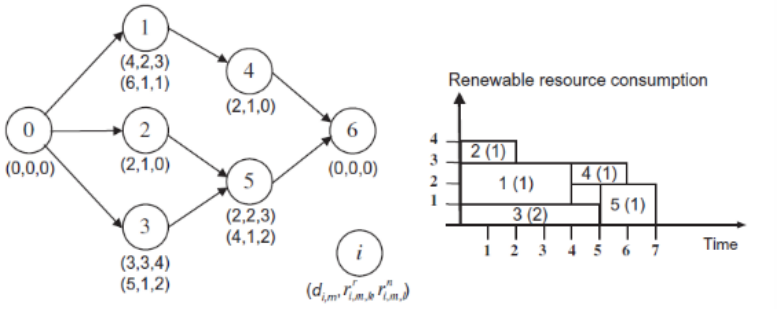


Fig. 1. A fictive example project with optimal resource profiles.. *Source*: Kolisch and Drexl (1997).

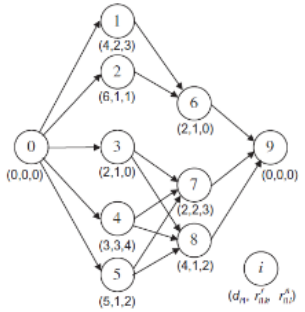


Fig. 2. The single-mode RCPSP network of Fig. 1 without any activity mod restrictions.

Single-mode activities: $x_0 + x_3 + x_6 + x_9 = 4$
Mode assignment for activity 1: $x_1 + x_2 = 1$
Mode assignment for activity 3: $x_4 + x_5 = 1$
Mode assignment for activity 5: $x_7 + x_8 = 1$
Non-renewable resource constraint: $3x_1 + x_2 + 4x_4 + 2x_5 + 3x_7 + 2x_8 \leqslant 8$

Single-mode activities: $x_0 \wedge x_3 \wedge x_6 \wedge x_9$
Mode assignment for activity 1: $(x_1 \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2})$
Mode assignment for activity 3: $(x_4 \vee x_5) \wedge (\overline{x_4} \vee \overline{x_5})$
Mode assignment for activity 5: $(x_7 \vee x_8) \wedge (\overline{x_7} \vee \overline{x_8})$

Assume a simple activity list AL = {0,1,2,3,4,5,6,7,8,9} and a set T denoting the set of assigned literals at a given moment (where x and $\bar{x}$ can not belong to T at the same moment). The DPLL algorithm for the example CNF $= x_0 \wedge x_3 \wedge x_6 \wedge x_9 \wedge (x_1 \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2}) \wedge (x_4 \vee x_5) \wedge (\overline{x_4} \vee \overline{x_5}) \wedge (x_7 \vee x_8) \wedge (\overline{x_7} \vee \overline{x_8}) \wedge (\overline{x_1} \vee \overline{x_4})$ runs as follows:

(1) Activity list AL = {0,1,2,3,4,5,6,7,8,9}.
Unit clause rule x_0, x_3, x_6, x_9: $T = \{x_0, x_3, x_6, x_9\}$.
CNF: $(x_1 \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2}) \wedge (x_4 \vee x_5) \wedge (\overline{x_4} \vee \overline{x_5}) \wedge (x_7 \vee x_8) \wedge (\overline{x_7} \vee \overline{x_8}) \wedge (\overline{x_1} \vee \overline{x_4})$.
(2) Selection of variable from AL, not in T: x_1: $T = \{x_0, x_3, x_6, x_9, x_1\}$.
CNF: $\overline{x_2} \wedge (x_4 \vee x_5) \wedge (\overline{x_4} \vee \overline{x_5}) \wedge (x_7 \vee x_8) \wedge (\overline{x_7} \vee \overline{x_8}) \wedge \overline{x_4}$.
Unit clause rule: $\overline{x_2}, \overline{x_4}$: $T = \{x_0, x_3, x_6, x_9, x_1, \overline{x_2}, \overline{x_4}\}$.
CNF: $x_5 \wedge (x_7 \vee x_8) \wedge (\overline{x_7} \vee \overline{x_8})$.
Unit clause rule: x_5 : $T = \{x_0, x_3, x_6, x_9, x_1, \overline{x_2}, \overline{x_4}, x_5\}$.
CNF: $(x_7 \vee x_8) \wedge (\overline{x_7} \vee \overline{x_8})$.
(3) Selection of variable from AL, not in T: x_7 : $T = \{x_0, x_3, x_6, x_9, x_1, \overline{x_2}, \overline{x_4}, x_5, x_7\}$.
CNF: $\overline{x_8}$.
Unit clause rule: $\overline{x_8}$: $T = \{x_0, x_3, x_6, x_9, x_1, \overline{x_2}, \overline{x_4}, x_5, x_7, \overline{x_8}\}$.

Table 1
The SAT mode assignment solution.

i	m	Fig. 1 $(d_{i,m}, r^r_{i,m,1}, r^n_{i,m,1})$	Fig. 2 i	T	d_{ft}
0	1	(0,0,0)	0	x_0	0
1	1	(4,2,3)	1	x_1	4
	2	(6,1,1)	2	$\overline{x_2}$	0
2	1	(2,1,0)	3	x_3	2
3	1	(3,3,4)	4	$\overline{x_4}$	0
	2	(5,1,2)	5	x_5	5
4	1	(2,1,0)	6	x_6	2
5	1	(2,2,3)	7	x_7	2
	2	(4,1,2)	8	$\overline{x_8}$	0
6	1	(0,0,0)	9	x_9	0

Table 2
A comparison between the pure SAT and the SAT(k) approach.

Set	#Var	#Ins	M (SAT(3))	M (SAT(4))	Avg.Cl(3)	Avg.Cl(4)	#Ins-M
J10	32	536	119	1.5	5883	21.4	0
J12	38	547	801	2.1	32,435	25.4	0
J14	44	551	5342	2.4	186,436	29.4	24
J16	50	550	14,445	2.7	454,055	33.4	162
J18	56	552	23,112	3.0	653,028	37.4	320
J20	62	554	26,759	3.4	694,061	41.4	378
J30	92	640	49,554	5.8	750,030	61.5	480

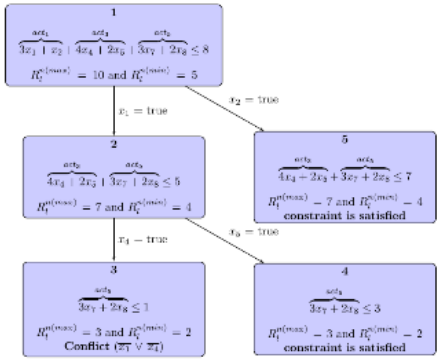
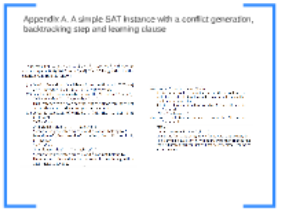
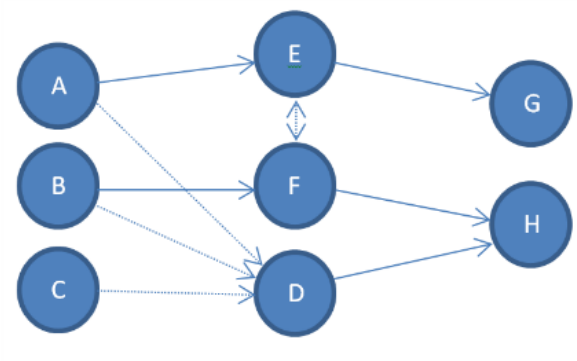
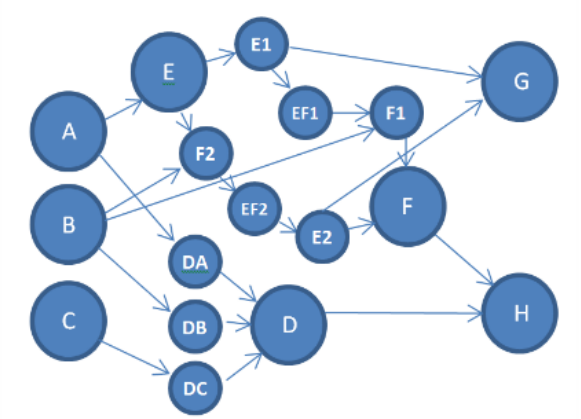


Fig. 3. Enumeration scheme of the example project.



Changing of an AND-precedence relation into an OR-precedence relation, and Bi-directional relation





- Activity D have a OR-precedence relation:
  - Activity D is transformed to a dummy activity;
  - Activities DA, DB, DC are created, duplicating activity D;
  - The realization of activity D, implies to realize one and only one activity: DA, DB and DC.
- Activities E and F have a Bi-directional relation:
  - Added 6 activities:
    - EF1/2 - is the changeover times activity;
    - E1, F1 - duplicate of activities E and F, with the order E » F;
    - E2, F2 - duplicate of activities E and F, with the order F » E.

Error:

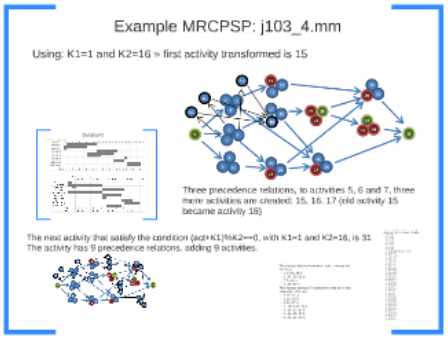
- B precedes F, but in F1 it need to wait for the EF activity also
- G is sucessor of E, but in the E1 it need to wait the EF activity also

Using (M)RCPSP instances to create OR-precede relations and BI-directional relations instances

In a RCPSP or MRCPSP instance:

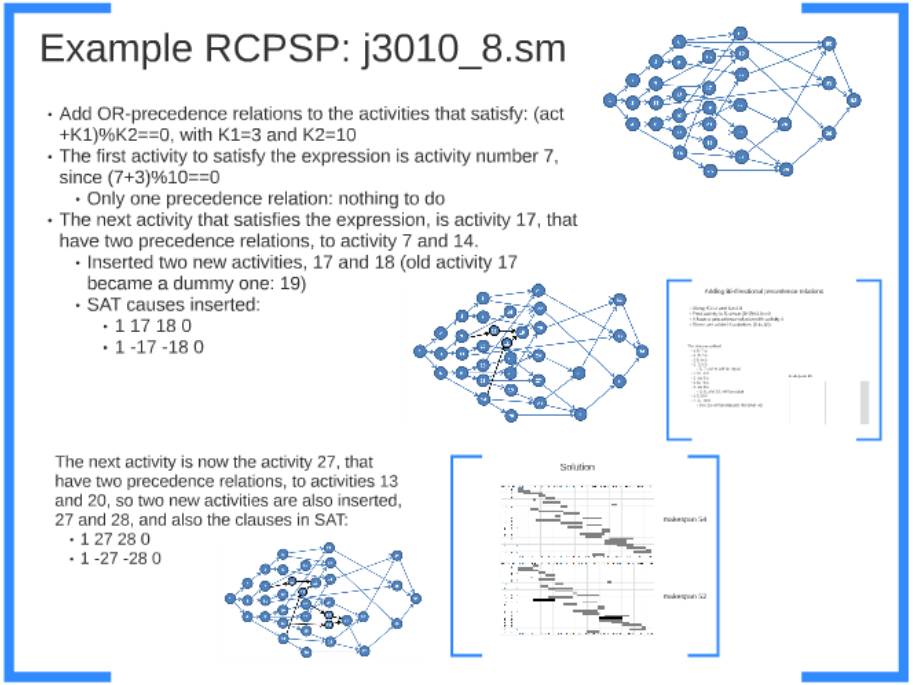
- 1. Select a number of activities, to make all its precedence relations as OR precedence relations (like in activity D);
- 2. Select a number of activities with AND precedence relations, and select one of its preceding activity, and make it as a bi-directional precedence relation (like activity E and F).

Step 1 does not produce any violation, but is done only if the selected activity is not a dummy one. Step 2 does not produce any violation, if the activities are inserted between the maximal activity that is preceding activity B, and the minimal activity that is succeeding activity A, assuming a transformation of a precedence of A over B.

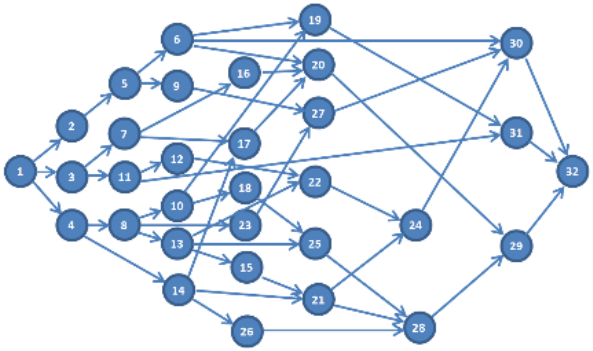


Rules:

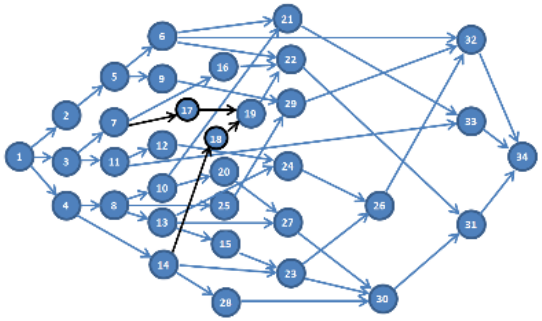
- Add OR-precedence relations to the activities that satisfy: **(act+K1)%K2==0**
- Add BI-directional we use an identical rule

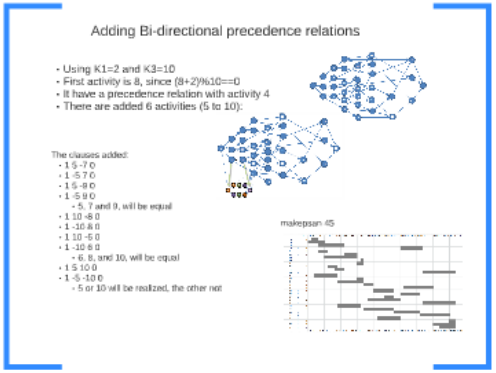


Example RCPSP: j3010_8.sm



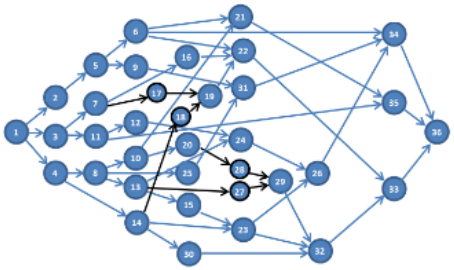
- Add OR-precedence relations to the activities that satisfy: (act +K1)%K2==0, with K1=3 and K2=10
- The first activity to satisfy the expression is activity number 7, since (7+3)%10==0
  - Only one precedence relation: nothing to do
- The next activity that satisfies the expression, is activity 17, that have two precedence relations, to activity 7 and 14.
  - Inserted two new activities, 17 and 18 (old activity 17 became a dummy one: 19)
  - SAT causes inserted:
    - 1 17 18 0
    - 1 -17 -18 0

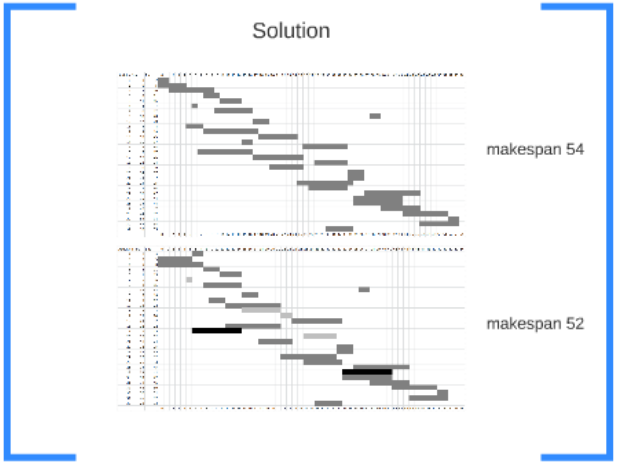




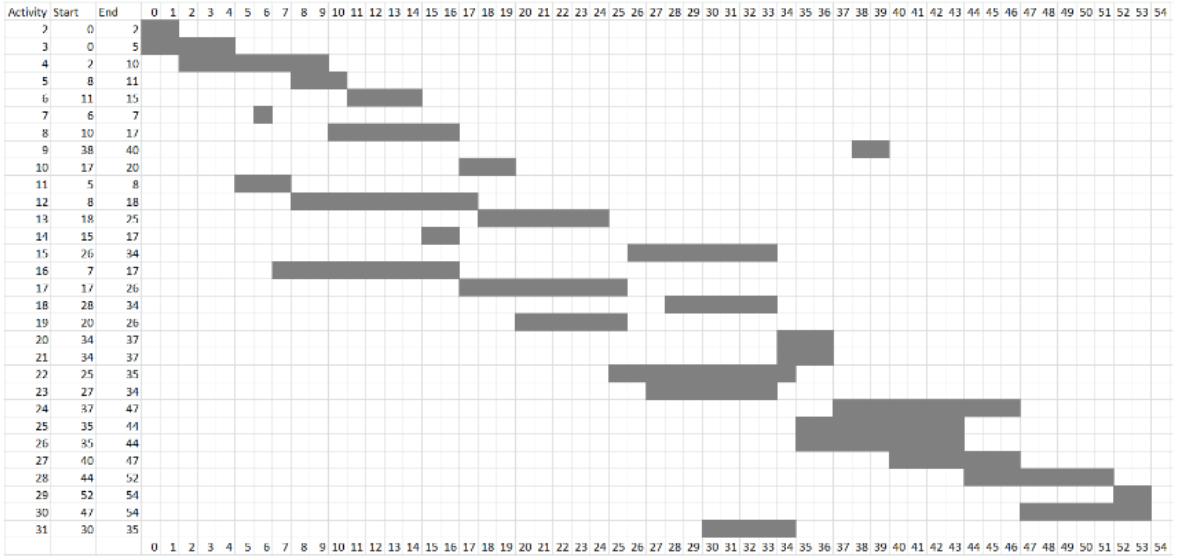
The next activity is now the activity 27, that have two precedence relations, to activities 13 and 20, so two new activities are also inserted, 27 and 28, and also the clauses in SAT:

- 1 27 28 0
- 1 -27 -28 0



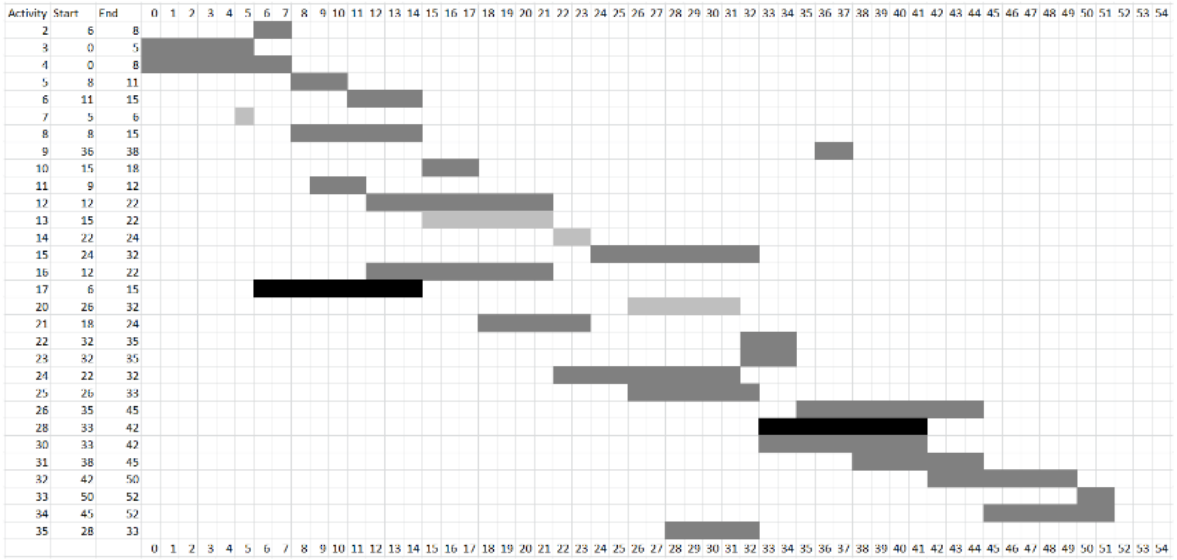


Solution



Activity	Start	End
2	0	2
3	0	5
4	2	10
5	8	11
6	11	15
7	6	7
8	10	17
9	38	40
10	17	20
11	5	8
12	8	18
13	18	25
14	15	17
15	26	34
16	7	17
17	17	26
18	28	34
19	20	26
20	34	37
21	34	37
22	25	35
23	27	34
24	37	47
25	35	44
26	35	44
27	40	47
28	44	52
29	52	54
30	47	54
31	30	35

makespan 54

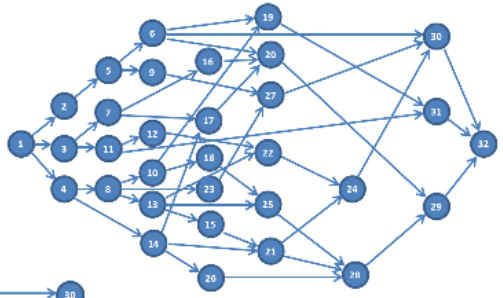


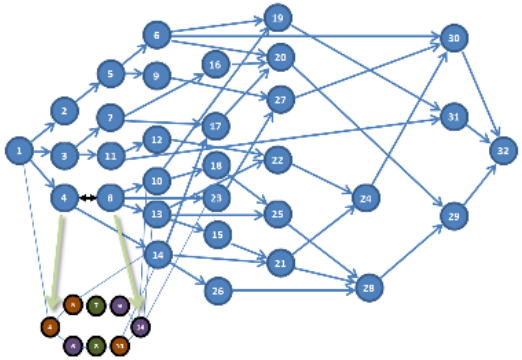
Activity	Start	End
2	6	8
3	0	5
4	0	8
5	8	11
6	11	15
7	5	6
8	8	15
9	36	38
10	15	18
11	9	12
12	12	22
13	15	22
14	22	24
15	24	32
16	12	22
17	6	15
20	26	32
21	18	24
22	32	35
23	32	35
24	22	32
25	26	33
26	35	45
28	33	42
30	33	42
31	38	45
32	42	50
33	50	52
34	45	52
35	28	33

makespan 52

Adding Bi-directional precedence relations

- Using K1=2 and K3=10
- First activity is 8, since (8+2)%10==0
- It have a precedence relation with activity 4
- There are added 6 activities (5 to 10):

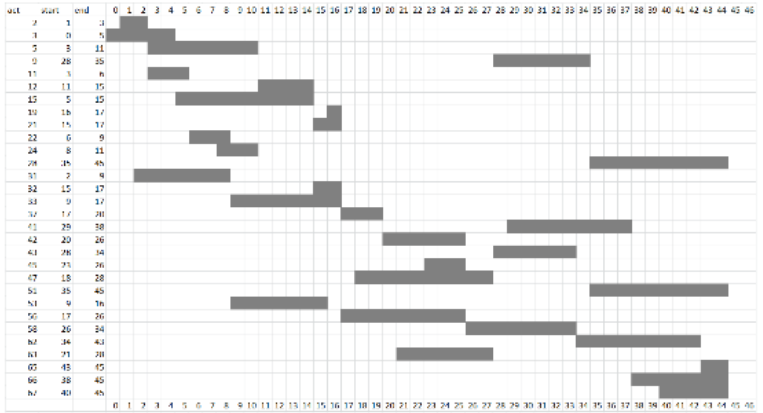




The clauses added:

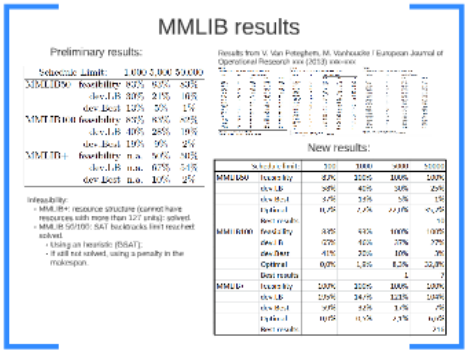
- 1 5 -7 0
- 1 -5 7 0
- 1 5 -9 0
- 1 -5 9 0
  - 5, 7 and 9, will be equal
- 1 10 -8 0
- 1 -10 8 0
- 1 10 -6 0
- 1 -10 6 0
  - 6, 8, and 10, will be equal
- 1 5 10 0
- 1 -5 -10 0
  - 5 or 10 will be realized, the other not

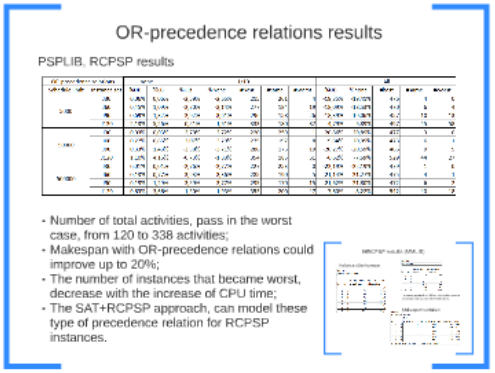
makepsan 45

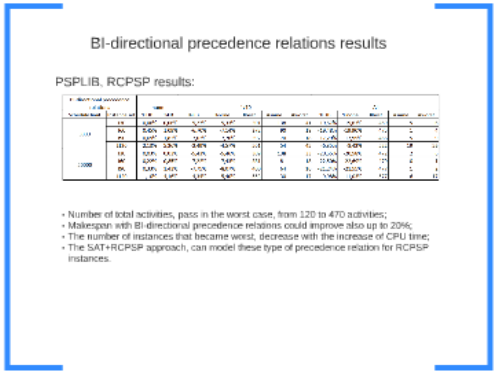


Tests and Results

1. Run without OR-precedence relations and Bi-directional relations:
   - Check if our procedure is close to the state-of-the-art in MRCPSP instance sets
2. Run with different density of OR-precedence relations:
   - Quantify the increase needed on the number of activities
   - Quantify the gain in makespan by consider OR-precedence relations, relative to treat those precedence relations as AND-precedence relations
3. Run with bi-directional relations.







MMLIB results

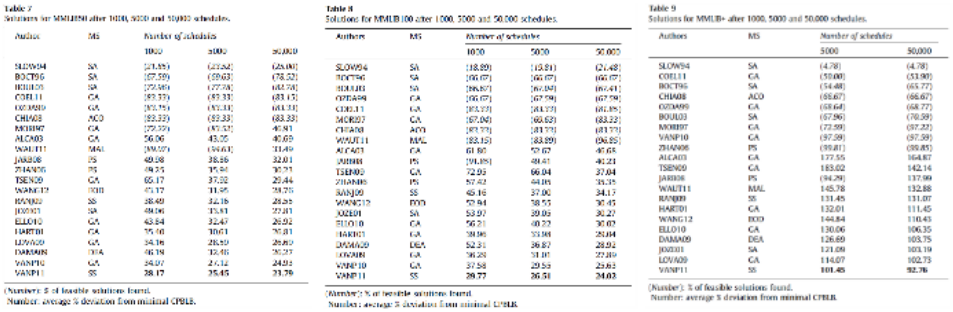
Preliminary results:

Schedule Limit:		1,000	5,000	50,000
MMLIB50	feasibility	83%	83%	83%
	dev.LB	30%	21%	16%
	dev.Best	13%	5%	1%
MMLIB100	feasibility	83%	83%	82%
	dev.LB	40%	28%	19%
	dev.Best	19%	9%	2%
MMLIB+	feasibility	n.a.	50%	50%
	dev.LB	n.a.	67%	54%
	dev.Best	n.a.	10%	2%

Infeasibility:

- MMLIB+: resource structure (cannot have resources with more than 127 units): solved.
- MMLIB 50/100: SAT backtracks limit reached: solved.
  - Using an heuristic (GSAT);
  - If still not solved, using a penalty in the makespan.

Results from V. Van Peteghem, M. Vanhoucke / European Journal of Operational Research xxx (2013) xxx–xxx



New results:

Schedule limit:		100	1000	5000	50000
MMLIB50	feasibility	83%	100%	100%	100%
	dev.LB	58%	40%	30%	25%
	dev.Best	37%	13%	5%	1%
	Optimal	0,2%	7,2%	22,0%	35,2%
	Best results				10
MMLIB100	feasibility	83%	93%	100%	100%
	dev.LB	65%	46%	37%	27%
	dev.Best	41%	20%	10%	3%
	Optimal	0,0%	1,9%	8,3%	32,8%
	Best results			1	7
MMLIB+	feasibility	100%	100%	100%	100%
	dev.LB	195%	147%	121%	104%
	dev.Best	59%	32%	17%	7%
	Optimal	0,0%	0,5%	2,1%	6,6%
	Best results				216

OR-precedence relations results

PSPLIB, RCPSP results

OR-precedence relations		none		1/10					All				
Schedule limit	Instance set	%UB	%LB	%UB	%none	#best	#same	#worst	%UB	%none	#best	#same	#worst
5000	J30	0,06%	0,06%	-2,59%	-2,65%	215	261	4	-19,35%	-19,41%	476	4	0
	J60	0,45%	1,09%	-2,70%	-3,14%	277	184	19	-18,09%	-18,53%	470	6	4
	J90	0,69%	1,82%	-2,92%	-3,61%	296	158	26	-18,39%	-19,06%	457	10	13
	J120	2,13%	5,26%	0,51%	-1,61%	333	180	87	-4,79%	-6,89%	497	45	58
50000	J30	0,03%	0,03%	-2,70%	-2,72%	220	259	1	-20,86%	-20,89%	477	3	0
	J60	0,22%	0,85%	-3,07%	-3,29%	275	197	8	-19,84%	-20,05%	473	6	1
	J90	0,33%	1,43%	-3,38%	-3,71%	286	175	19	-20,27%	-20,59%	466	9	5
	J120	1,10%	4,16%	-0,73%	-1,83%	354	185	61	-6,62%	-7,69%	529	44	27
500000	J30	0,01%	0,01%	-2,76%	-2,77%	227	253	0	-22,19%	-22,19%	479	1	0
	J60	0,13%	0,77%	-3,23%	-3,36%	285	190	5	-21,14%	-21,27%	475	4	1
	J90	0,19%	1,29%	-3,59%	-3,77%	295	170	15	-21,62%	-21,80%	472	6	2
	J120	0,63%	3,66%	-1,30%	-1,93%	353	200	47	-7,60%	-8,22%	542	40	18

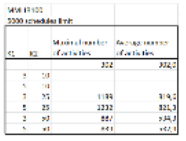
- Number of total activities, pass in the worst case, from 120 to 338 activities;
- Makespan with OR-precedence relations could improve up to 20%;
- The number of instances that became worst, decrease with the increase of CPU time;
- The SAT+RCPSP approach, can model these type of precedence relation for RCPSP instances.

MRCPSP results (MMLIB)

Instance size increase

MMLIB50
5000 schedules limit

K1	K2	Maximal number of activities	Average number of activities
-	-	152	152,0
3	10	734	507,1
5	10	662	505,5
3	25	716	302,0
5	25	689	305,9
3	50	332	200,2
5	50	287	199,0
3	1	2411	1829,1



Instances greater than 900 activity-modes, are not processed due data structure restrictions.

Makespan variation

MMLIB50
5000 schedules limit

K1	K2	Better makespan	Same makespan	Worst makespan	Total
3	10	191	113	147	450
5	10	183	114	154	451
3	25	172	145	133	450
5	25	170	152	128	450
3	50	131	192	127	450
5	50	122	191	140	453

Using only results on instances with feasible solutions

BI-directional precedence relations results

PSPLIB, RCPSP results:

BI-directional precedence relations		none		1/10					All				
Schedule limit	Instance set	%UB	%LB	%UB	%none	#best	#same	#worst	%UB	%none	#best	#same	#worst
5000	J30	0,06%	0,06%	-5,27%	-5,33%	301	138	41	-19,57%	-19,63%	469	5	6
	J60	0,45%	1,09%	-6,70%	-7,14%	372	90	18	-19,48%	-19,90%	475	1	4
	J90	0,69%	1,82%	-7,02%	-7,70%	392	70	18	-17,29%	-17,95%	466	5	9
	J120	2,13%	5,26%	-2,49%	-4,57%	501	54	45	-6,35%	-8,42%	522	19	59
50000	J30	0,03%	0,03%	-5,43%	-5,46%	309	138	33	-20,55%	-20,58%	472	2	6
	J60	0,22%	0,85%	-7,22%	-7,43%	381	81	18	-22,50%	-22,69%	479	0	1
	J90	0,33%	1,43%	-7,75%	-8,07%	400	64	16	-21,24%	-21,55%	477	1	2
	J120	1,10%	4,16%	-4,34%	-5,40%	553	30	17	-9,98%	-11,03%	577	6	17

- Number of total activities, pass in the worst case, from 120 to 470 activities;
- Makespan with BI-directional precedence relations could improve also up to 20%;
- The number of instances that became worst, decrease with the increase of CPU time;
- The SAT+RCPSP approach, can model these type of precedence relation for RCPSP instances.

Conclusions and Future Work

- Proposed an approach to solve the RCPSP with logical constraints:
  - OR precedence relations;
  - BI-directional precedence relations;
- The method uses an extension of multi-mode, where the realization of each activity-mode is controlled by SAT, with more flexibility over the MRCPSP procedures, and the schedule is done by a RCPSP procedure
- The method, applied to MRCPSP instances, reveal to be competitive.
- Future work we plan:
  - Study these type of logical constraints without resources, and with multiple mode;
  - Study mode identity constraints in multiple mode.