Progress on the Progressive Party Problem

Helmut Simonis

Cork Constraint Computation Centre
Computer Science Department
University College Cork
Ireland

CP-AI-OR 2009

Main Points

- We need to handle benchmarks as precious resources
- CP is a competitive technique to solve the Progressive Party Problem

Outline

Problem Definition

Progressive Party

The problem is to timetable a party at a yacht club. Certain boats are to be designated hosts, and the crews of the remaining boats in turn visit the host boats for several successive half-hour periods. The crew of a host boat remains on board to act as hosts while the crew of a guest boat together visits several hosts. Every boat can only host a limited number of guests at a time (its capacity) and crew sizes are different. The party lasts for 6 time periods. A guest boat cannot not revisit a host and guest crews cannot meet more than once. The problem facing the rally organizer is that of minimizing the number of host boats.

Data

Boat	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Capacity	6	8	12	12	12	12	12	10	10	10	10	10	8	8
Crew	2	2	2	2	4	4	4	1	2	2	2	3	4	2
Boat	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Capacity	8	12	8	8	8	8	8	8	7	7	7	7	7	7
Crew	3	6	2	2	4	2	4	5	4	4	2	2	4	5
Boat	29	30	31	32	33	34	35	36	37	38	39	40	41	42
Capacity	6	6	6	6	6	6	6	6	6	6	9	0	0	0
Crew	2	4	2	2	2	2	2	2	4	5	7	2	3	4

Related Work (1)

- Brailsford, Hubbard, Smith and Williams (1995/1996)
- Initial description of problem and model
- 6 time period problem can not be solved with MIP
- .. but can be solved in 28 minutes with CP

Partial Search in CHIP

- Beldiceanu, Bourreau, Chan and Rivreau (1997)
- 8 time periods 672 (deep) backtracks
- 9 time periods require > 6 million backtrack, hours

Local Search

- Galinier and Hao (1998)
  - 9 time periods in 4 seconds
- Walser (1998)
  - Introduced problem variants
  - Solved with Pseudo-Boolean 0/1 local search procedure
- Van Hentenryck and Michel (2005)
  - Comet Book: Constraint-based local search

MIP Works

- Kalvelagen (2001)
- GAMS/CPLEX
- 6 time periods 9000 seconds
- Sequential decomposition
- 6 time periods 4 seconds
- 7 time periods 5 seconds

Where is CP in comparison?

- CSPLIB problem 13, does not mention any improvements since Brailsford et al from 1996
- This is not the editor's fault!
- Is this the right message from the CP community?

Contribution

- “Reasonable CP model” for this problem
- Results compared to best local search
- Not the best CP result possible

Standard Decomposition

- Host boats and their capacity given
- Ignore host teams, only consider free capacity
- Assign guest teams to host boats

Model

- Assign guest boats to hosts for each time period
- Matrix (size *NrGuests* $\times$ *NrPeriods*) of domain variables x_{ij}
- Variables range over possible hosts 1..*NrHosts*

Constraints

- Each guest boat visits a host boat atmost once
- Two guest boats meet at most once
- All guest boats assigned to a host in a time period fit within spare capacity of host boat

Each guest visits a hosts atmost once

- The variables for a guest and different time periods must be pairwise different
- `alldifferent` constraint on rows of matrix
- $\text{alldifferent}(\{x_{ij} | 1 \leq j \leq \text{NrPeriods}\})$

Two guests meet at most once

- The variables for two guests can have the same value for atmost one time period
- Constraints on each pair of rows i_1, i_2 in matrix
- $\sum_{1 \leq j \leq \text{NrPeriods}} (x_{i_1 j} = x_{i_2 j}) \leq 1$
- $x_{i_1 j} = x_{i_2 j} \Rightarrow x_{i_1 k} \neq x_{i_2 k}, j < k \leq \text{NrPeriods}$

All guests assigned to a host in a time period fit within spare capacity of host boat

- Capacity constraint expressed as bin packing for each time period
- Each host boat is a bin with capacity from 0 to its unused capacity
- Each guest is an item to be assigned to a bin
- Size of item given by crew size of guest boat

Bin Packing Constraint (Shaw 2004)

- Global constraint
  `bin_packing(Assignment,Sizes,Capacity)`
- Items of different sizes are assigned to bins
- Assignment of item modelled with domain variable (first argument)
- Size of items fixed: integer values (second argument)
- Each bin may have a different capacity
- Capacity of each bin given as a domain variable (third argument)

Outline

Search

- Layered search, assign each time period completely
- Incomplete, credit based search within each time period
- First fail variable selection in each period
- Randomized variable selection
- Restart from top when credit runs out

Changing time periods

Problem	Size	Naive	FF	Layered	Credit	Random
10	5	0.812	1.453	1.515	0.828	1.922
10	6	14.813	2.047	2.093	1.219	2.469
10	7	79.109	3.688	50.250	3.234	3.672
10	8	-	-	141.609	55.156	6.328
10	9	-	-	-	-	10.281

Outline

Further Improvement

- Idea: There is no real effect of including later time periods in constraint model
- Clearly shown in visualization
- Only current time period matters
- Decomposition: Set up model for one period at a time

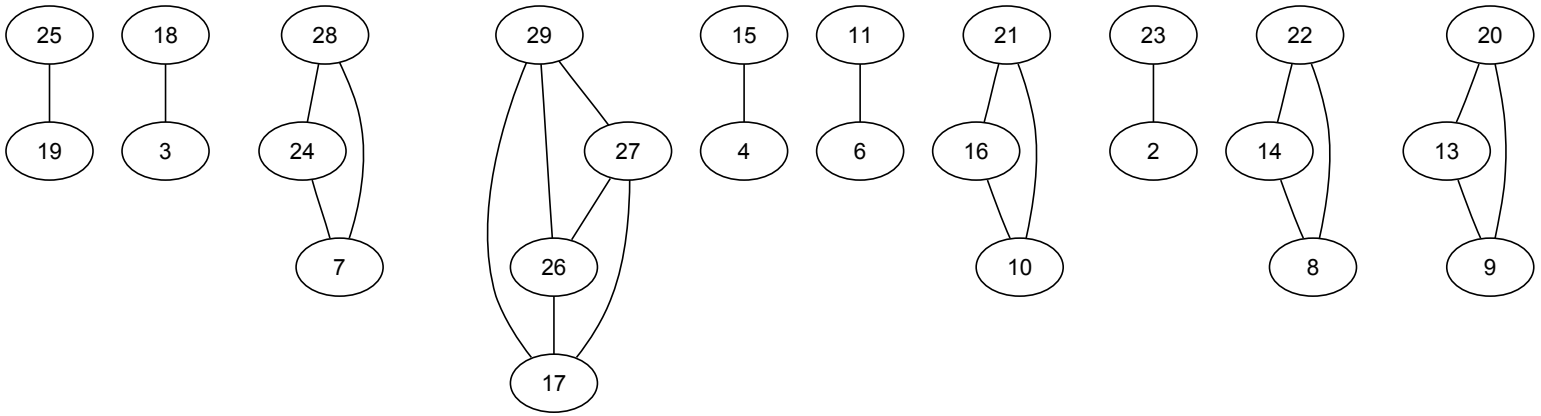
Fine Grained Decomposition

Old	New
Bin packing	Bin packing
Alldifferent	Domain restrictions
Meet at most once	Disequalities between guest boats

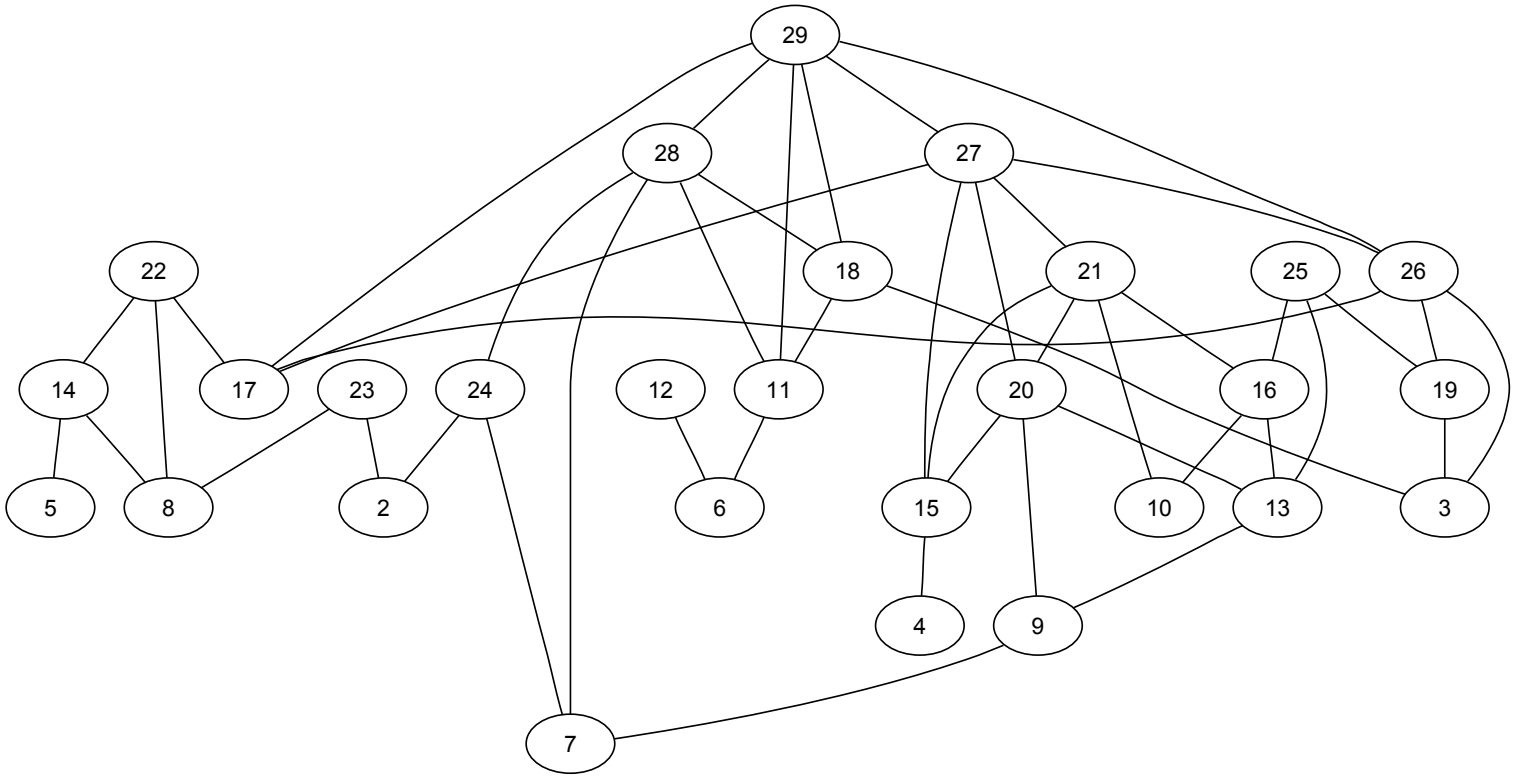
Generated Graph Coloring Problem

- Guest boats = Nodes
- Host boats = Colors
- Disequality constraints = Edges in graph

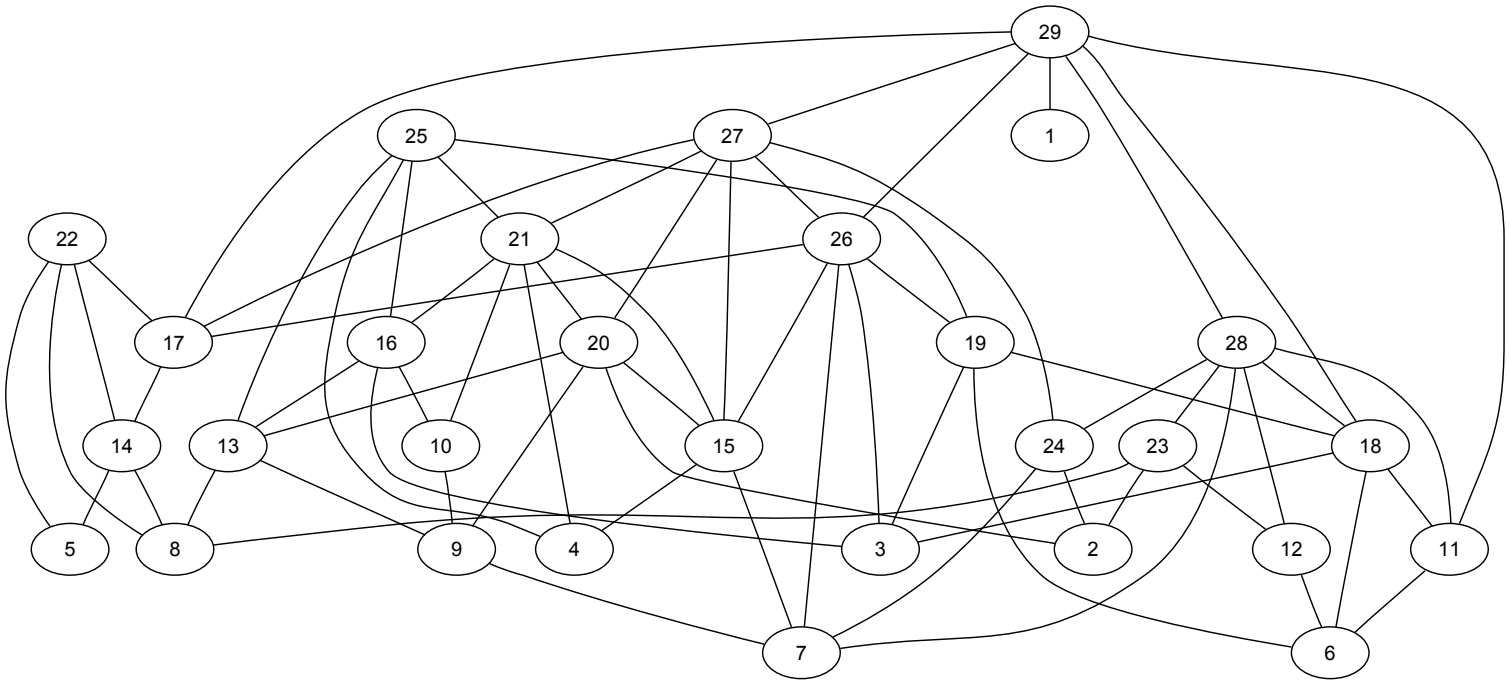
Visualization (Time period 2)



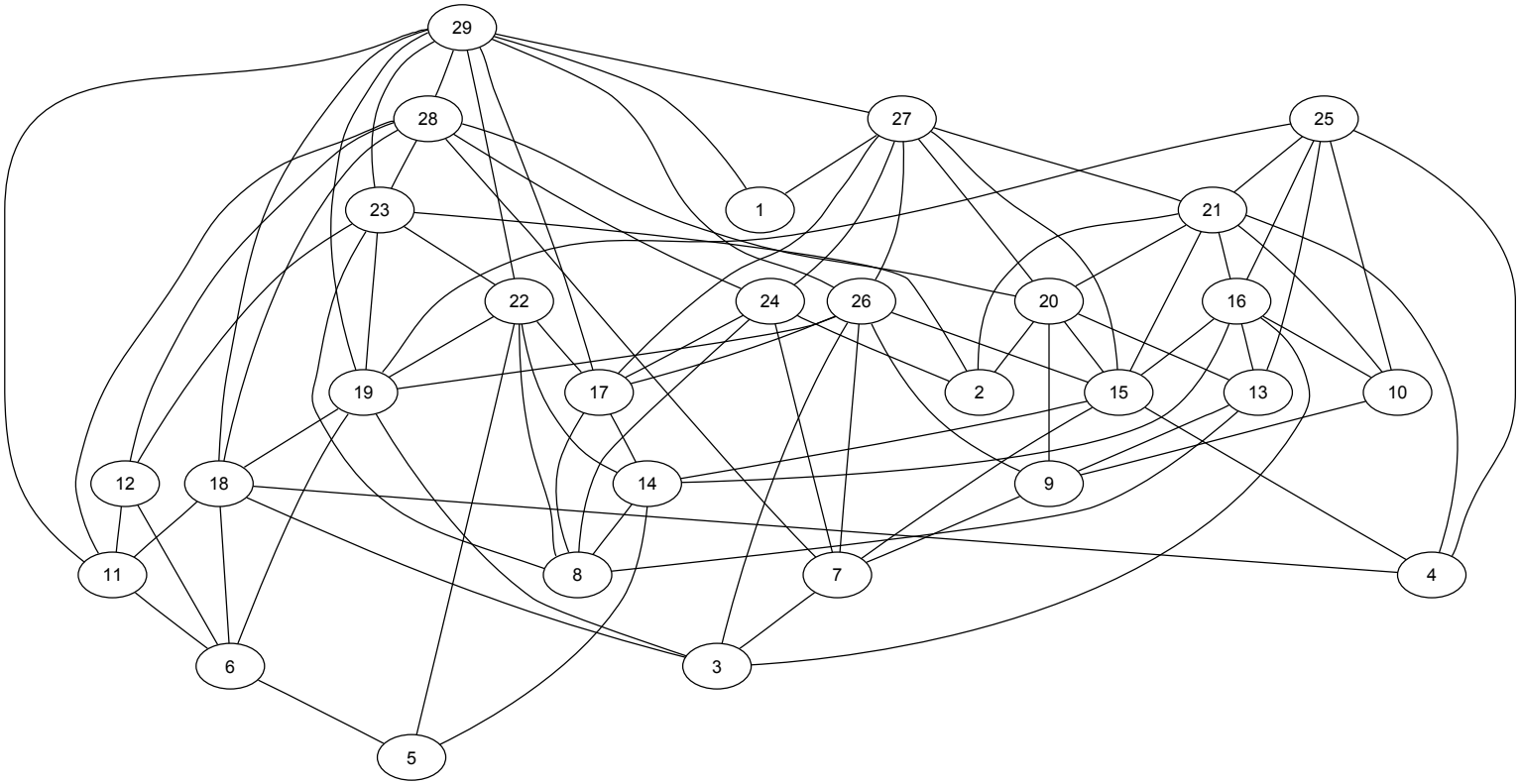
Visualization (Time period 3)



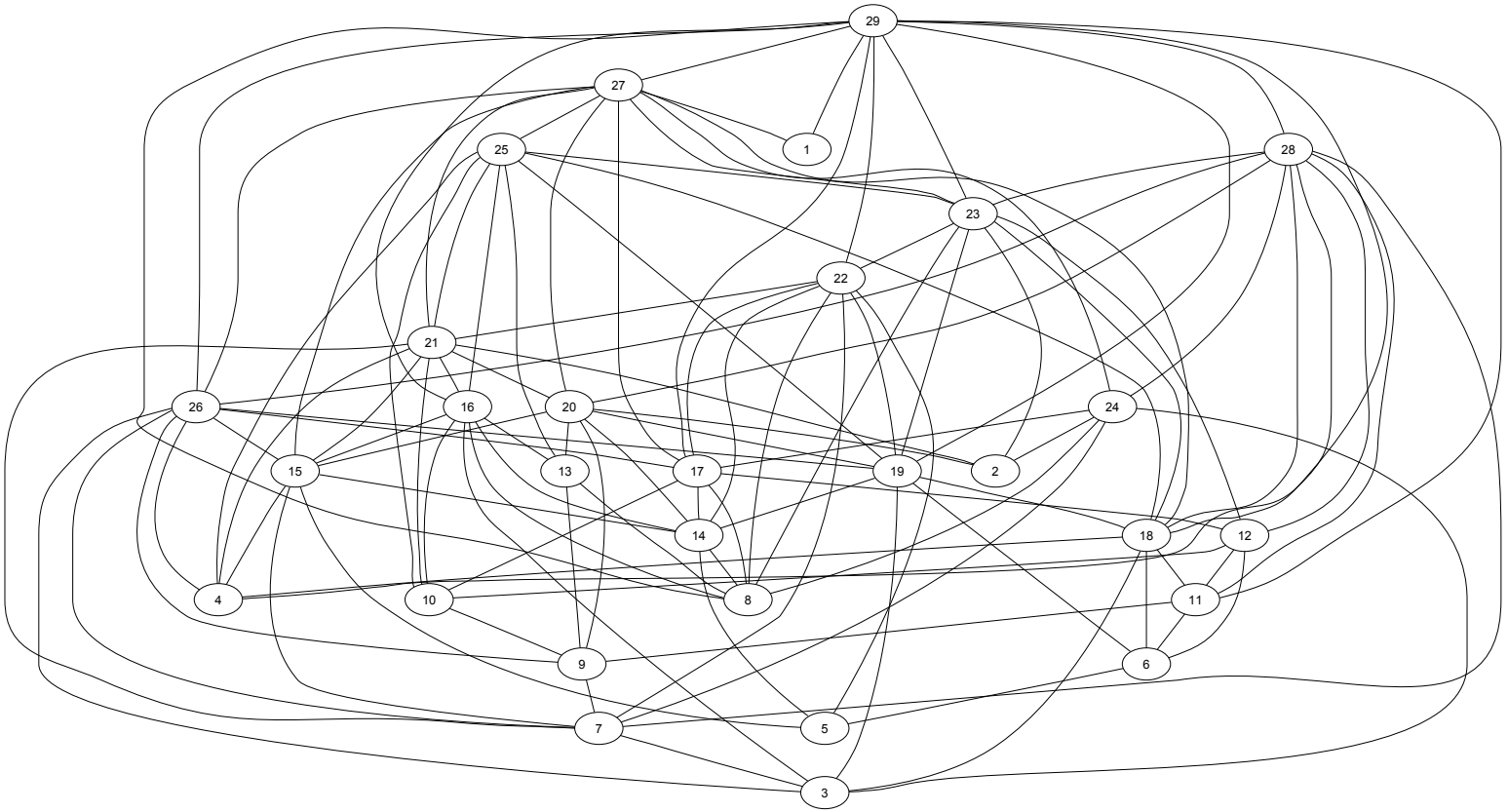
Visualization (Time period 4)



Visualization (Time period 5)



Visualization (Time period 6)

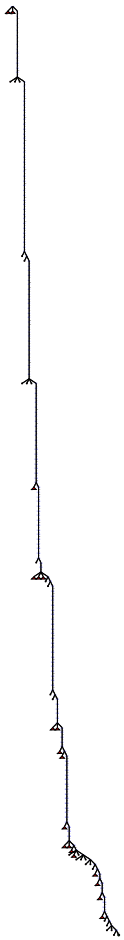


Solving the Graph Coloring Problem

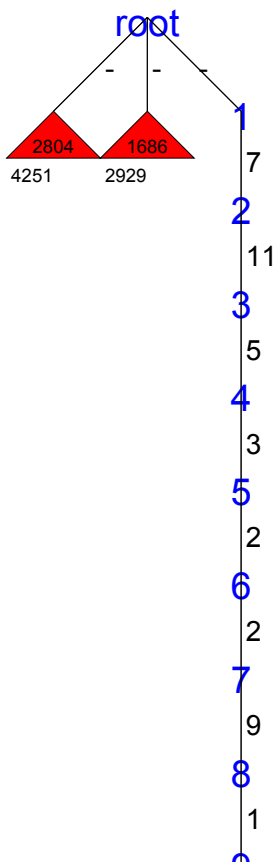
- Use disequality constraints
  - Weak propagation
- Extract `alldifferent` constraints
  - Edge clique cover problem
  - Is this folklore?
  - Choice of consistency method
- Use `somedifferent` global constraint (Richter et al 2006, Galinier et al 2008)
  - Heavy
  - Interaction with `bin packing` constraint

Outline

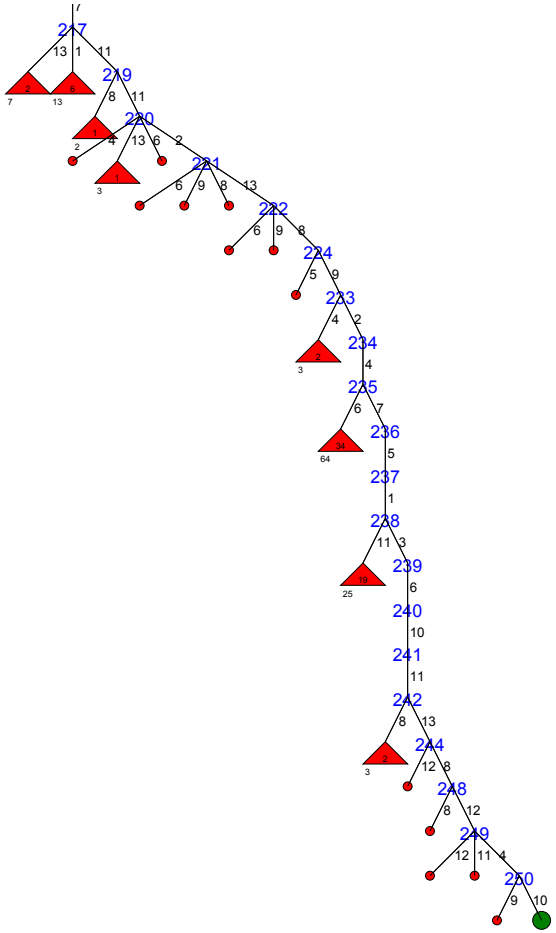
9 Time Periods, GAC



9 Time Periods, GAC (Zoomed)



9 Time Periods, GAC (Zoomed)



Comparison to Comet

		ECLiPSe 6.0				Comet			
Nr	Size	Solved	Min	Max	Avg	Solved	Min	Max	Avg
1	6	100	0.187	0.343	0.226	100	0.33	0.38	0.35
1	7	100	0.218	0.515	0.271	100	0.39	0.49	0.44
1	8	100	0.250	2.469	0.382	100	0.50	0.72	0.57
1	9	100	0.266	9.906	1.253	100	0.74	1.46	1.01
1	10	100	0.375	136.828	23.314	100	1.47	41.72	4.68
2	6	100	0.218	2.375	0.624	100	0.37	0.52	0.43
2	7	100	0.266	3.453	1.117	100	0.47	1.64	0.73
2	8	100	0.297	15.421	2.348	100	0.75	7.16	2.69
2	9	100	0.469	107.187	20.719	99	4.41	162.96	33.54
3	6	100	0.219	3.266	0.551	100	0.37	0.56	0.43
3	7	100	0.250	3.734	0.889	100	0.49	1.45	0.74
3	8	100	0.296	21.360	2.005	100	0.84	11.64	2.85
3	9	100	1.078	173.266	34.774	96	4.41	164.44	40.10
4	6	100	0.219	9.922	2.443	100	0.39	0.72	0.47
4	7	100	0.360	25.297	3.531	100	0.55	2.33	0.87
4	8	100	0.438	53.547	8.848	100	1.23	11.38	3.68
4	9	63	3.062	494.109	206.324	94	8.35	166.90	59.55
5	6	100	0.203	7.922	1.498	100	0.53	5.29	1.67
5	7	100	0.266	28.000	5.889	100	1.77	132.82	29.72
6	6	100	0.219	15.219	2.147	100	0.58	31.84	2.74
6	7	100	0.407	64.312	11.328	88	3.24	152.37	56.92

Summary

- We have to keep up with the competition
- Benchmarks are important resource, don't waste it
- We should do something about CSPLIB!
- Simple, up-to-date model is competitive
- Improvements still possible