Solving the Kirkman's Schoolgirl Problem in a Few Seconds

Nicolas Barnier & **Pascal Brisset**
CENA/ENAC, Toulouse, France

September 9th, 2002

Solving the Kirkman's Schoolgirl Problem in a Few Seconds

Nicolas Barnier & **Pascal Brisset**
CENA/ENAC, Toulouse, France

September 9th, 2002

The Kirkman's Schoolgirl Problem (1850)

A school mistress has fifteen girl pupils and she wishes to take them on a daily walk. The girls are to walk in five rows of three girls each. It is required that no two girls should walk in the same row more than once per week. Can this be done?

Monday	0 1 2	3 4 5	6 7 8	9 10 11	12 13 14

The Kirkman's Schoolgirl Problem (1850)

A school mistress has fifteen girl pupils and she wishes to take them on a daily walk. The girls are to walk in five rows of three girls each. It is required that no two girls should walk in the same row more than once per week. Can this be done?

Monday	0 1 2	3 4 5	6 7 8	9 10 11	12 13 14
Tuesday	0 3 6	1 4 9	2 5 12	7 10 13	8 11 14

The Kirkman's Schoolgirl Problem (1850)

A school mistress has fifteen girl pupils and she wishes to take them on a daily walk. The girls are to walk in five rows of three girls each. It is required that no two girls should walk in the same row more than once per week. Can this be done?

Monday	0 1 2	3 4 5	6 7 8	9 10 11	12 13 14
Tuesday	0 3 6	1 4 9	2 5 12	7 10 13	8 11 14
Wednesday	0 4 13	1 3 14	2 7 11	5 8 10	6 9 12
...					

The Kirkman's Schoolgirl Problem (1850)

A school mistress has fifteen girl pupils and she wishes to take them on a daily walk. The girls are to walk in five rows of three girls each. It is required that no two girls should walk in the same row more than once per week. Can this be done?

Monday	0 1 2	3 4 5	6 7 8	9 10 11	12 13 14
Tuesday	0 3 6	1 4 9	2 5 12	7 10 13	8 11 14
Wednesday	0 4 13	1 3 14	2 7 11	5 8 10	6 9 12
...					

Constraint community reformulation (10 in CSPLib): the Social Golfer Problem

32 golfers want to play in 8 groups of 4 each week, in such way that any two golfers play in the same group at most once. How many weeks can they do this for?

Generalization to w weeks of g groups, each one containing s golfers : g-s-w

Symmetries

- Players can be exchanged inside groups: $\phi_P \in \mathcal{P}_3$ for 35 groups;
- Groups can be exchanged inside weeks: $\phi_G \in \mathcal{P}_5$ for 7 weeks
- Weeks can be ordered arbitrarily: $\phi_W \in \mathcal{P}_7$;
- Players can be renamed among $15!$ permutations: $\phi_X \in \mathcal{P}_{15}$;
- ... and combinations of previous ones.

Problem: finding all non-isomorphic solutions

Kirkman found the 7 unique solutions for 5-3-7 instance in 1850!

Outline

- Related Work
- Model
- Isomorphism Checking
- Deep Pruning
- SBDD+
- Results

Related Work

- [Gervet, Constraints 97]: Modelisation with set variables for *Steiner Systems*,
- [Smith, CPAIOR'01]: Modelisation, symmetry breaking (SBDS)
- [Fahle, Schamberger, Sellmann, CP'01]: symmetry breaking (SBDD); 2 hours for 5-3-7
- [Preswitch, CPAIOR'02]: Randomised Backtracking; new results
- [Harvey, Sellmann, CPAIOR'02]: Heuristic Propagation; 6 minutes for 5-3-7
- [Puget, CP'02]: Symmetry Breaking; 8 seconds for 5-3-7
- Combinatorics community:
  - Social Golfer Problem $\equiv$ Resolvable Steiner System
  - Solutions found for 7-3-10, 7-4-9 ...
  - 8-4-10 is not pure enough (one player cannot meet all others)
- Solutions found with constraints: Warwick Harvey's page (`www.icparc.ac.ac.uk`)

Model

Using "set" variables [Gervet, 97] automatically removes symmetries inside groups: $G_{i,j}$ with i indexing weeks and j indexing $groups$.

Constraints: cardinal, partition inside weeks, no common couples

Redundant constraints: specialized "atmost1" for sets of cardinal s [Gervet, 01]

Ordering groups inside weeks: $\min G_{i,j} < \min G_{i,j+1}$

Ordering weeks: $\min(G_{i,1} \setminus \{0\}) < \min(G_{i+1,1} \setminus \{0\})$

Symmetry among players cannot be removed by constraints.

However

- First week is fixed
- First group of second week is fixed (with smallest players)
- "First" players are put in "First" groups: $j \in G_{i,j'}$ for $j' \leq j \leq g$
- Players together in the first week are in ordered groups in second week
- Order of groups in first week is kept in second week.

Efficient Isomorphism Checking

- Looking for pairs
- Discovering the non-isomorphism as soon as possible

P

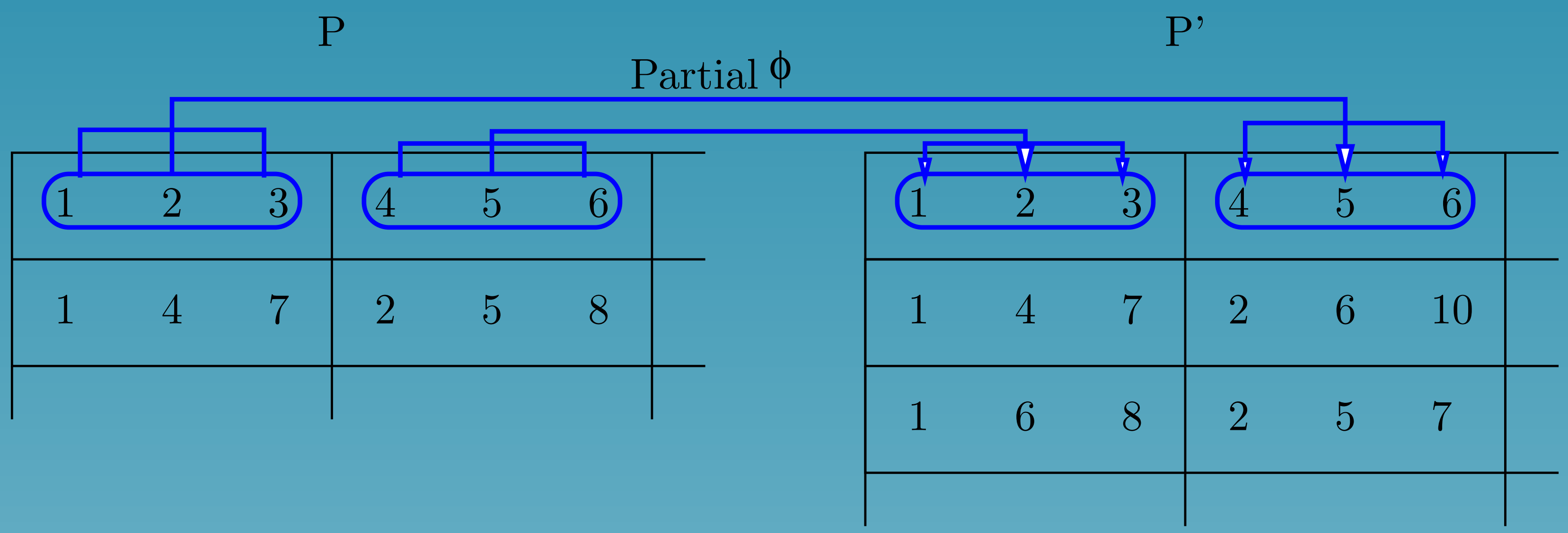
1	2	3	4	5	6
1	4	7	2	5	8

P'

1	2	3	4	5	6
1	4	7	2	6	10
1	6	8	2	5	7

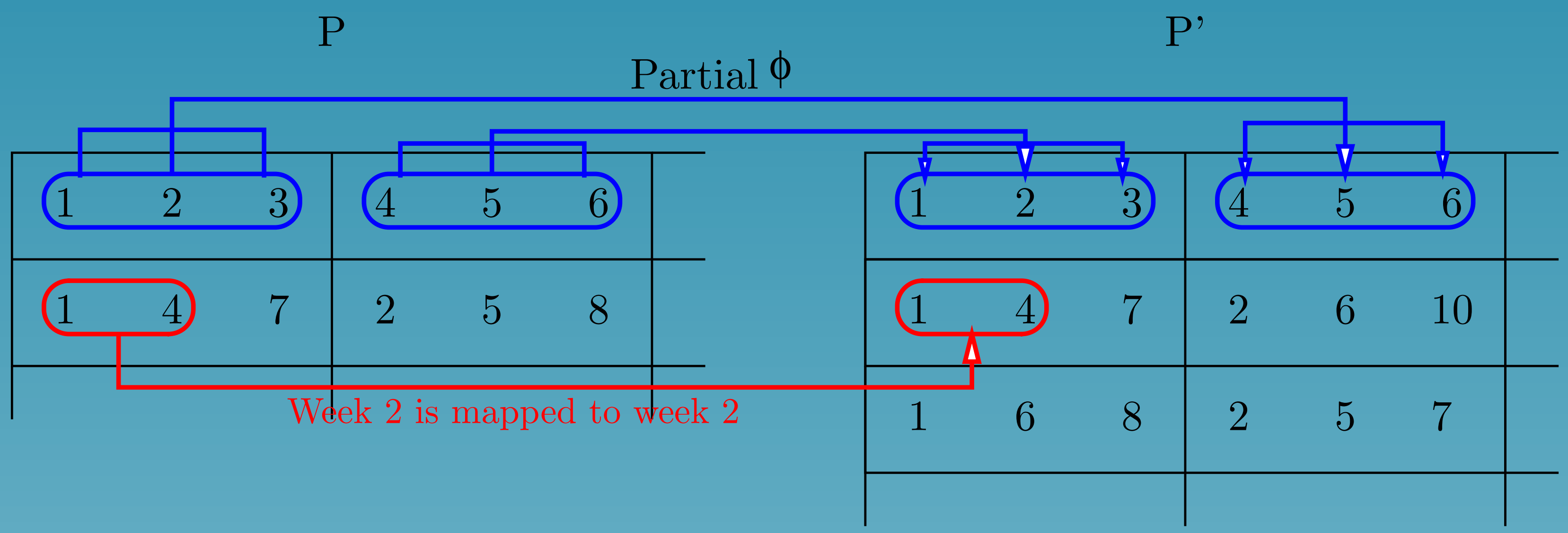
Efficient Isomorphism Checking

- Looking for pairs
- Discovering the non-isomorphism as soon as possible



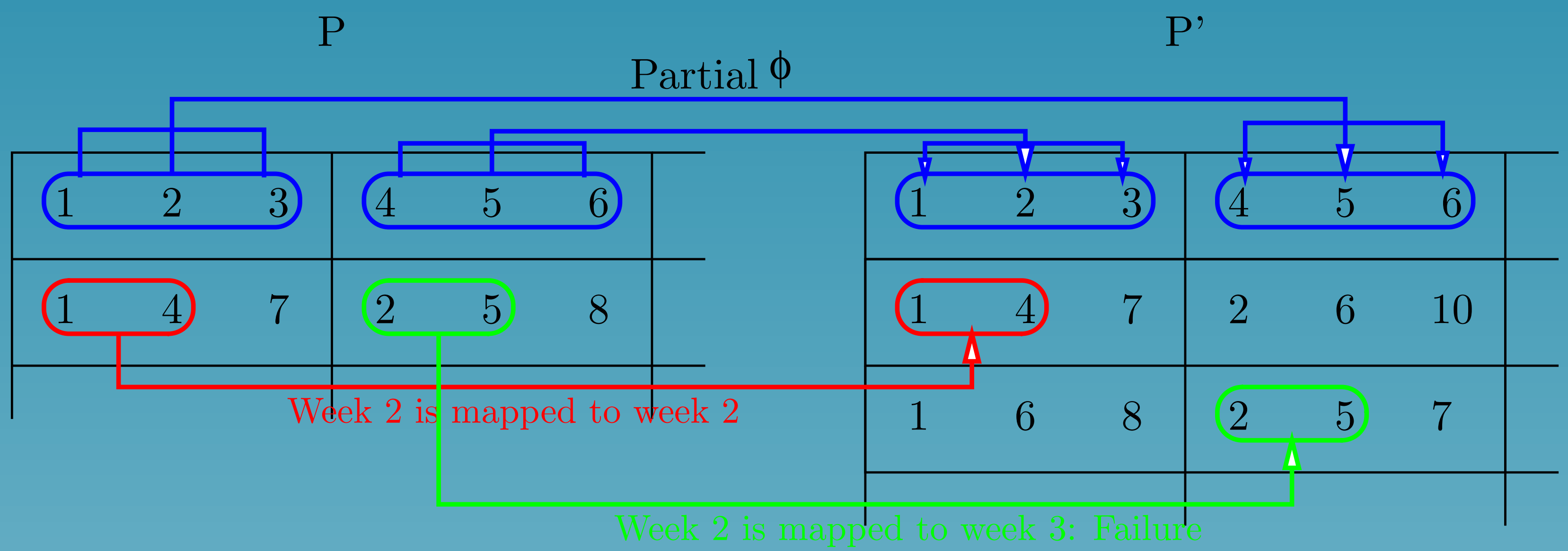
Efficient Isomorphism Checking

- Looking for pairs
- Discovering the non-isomorphism as soon as possible



Efficient Isomorphism Checking

- Looking for pairs
- Discovering the non-isomorphism as soon as possible



First results to find the 7 non-isomorphic solutions of 5-3-7

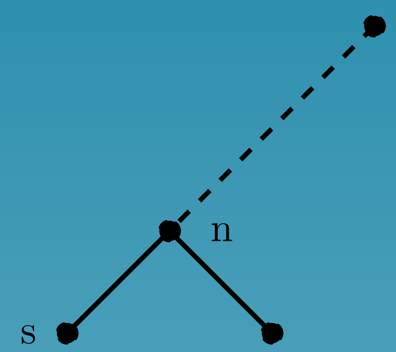
Choice points	Fails	Solutions	CPU(s)
20062206	19491448	20640	5925

(using FaCiLe on a PIII 700MhZ)

Comparable with CPU-time announced in [Sellmann, CP'01] using symmetry breaking.

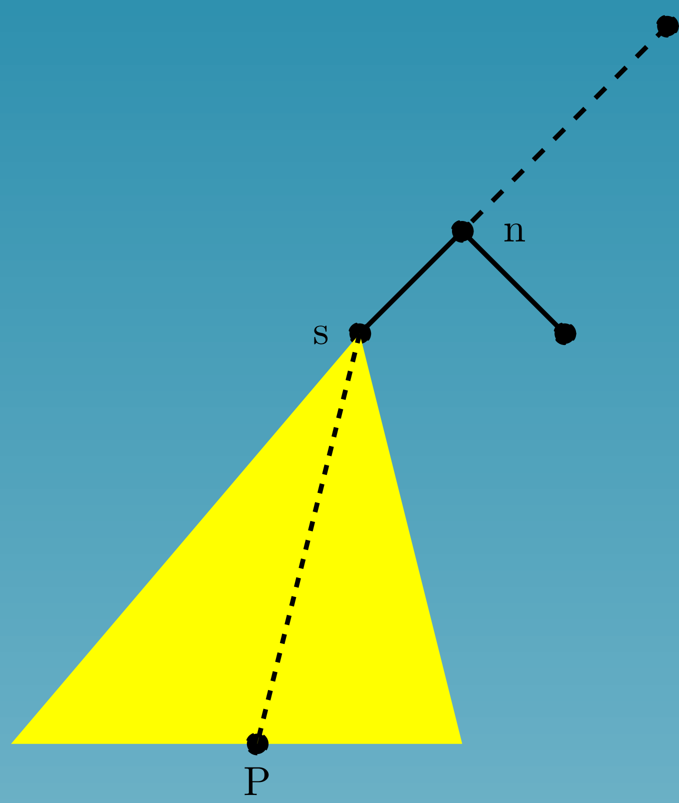
MC KAY **Pruning**

Suggested in [Mc Kay, 81] for search for graph isomorphism



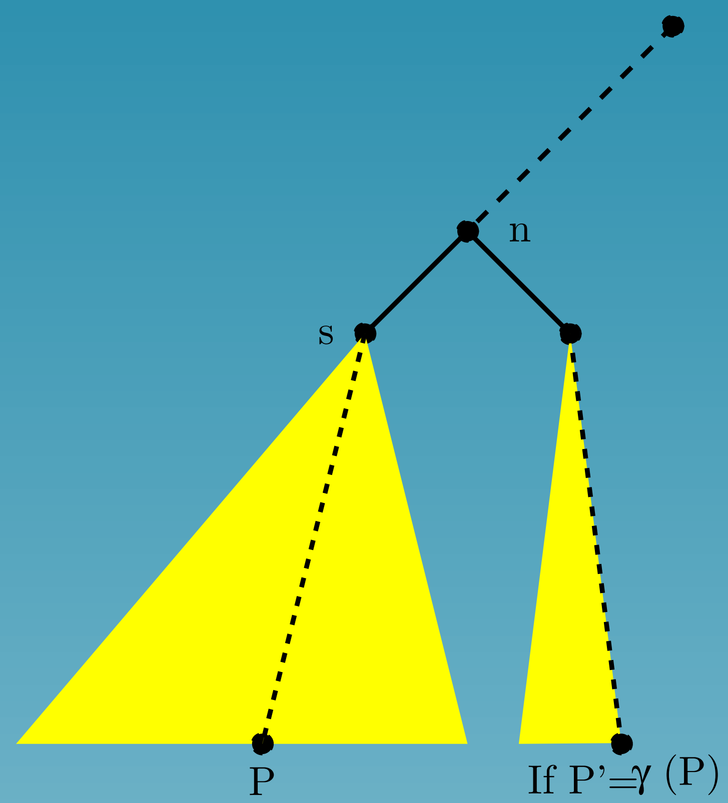
MC KAY Pruning

Suggested in [Mc Kay, 81] for search for graph isomorphism



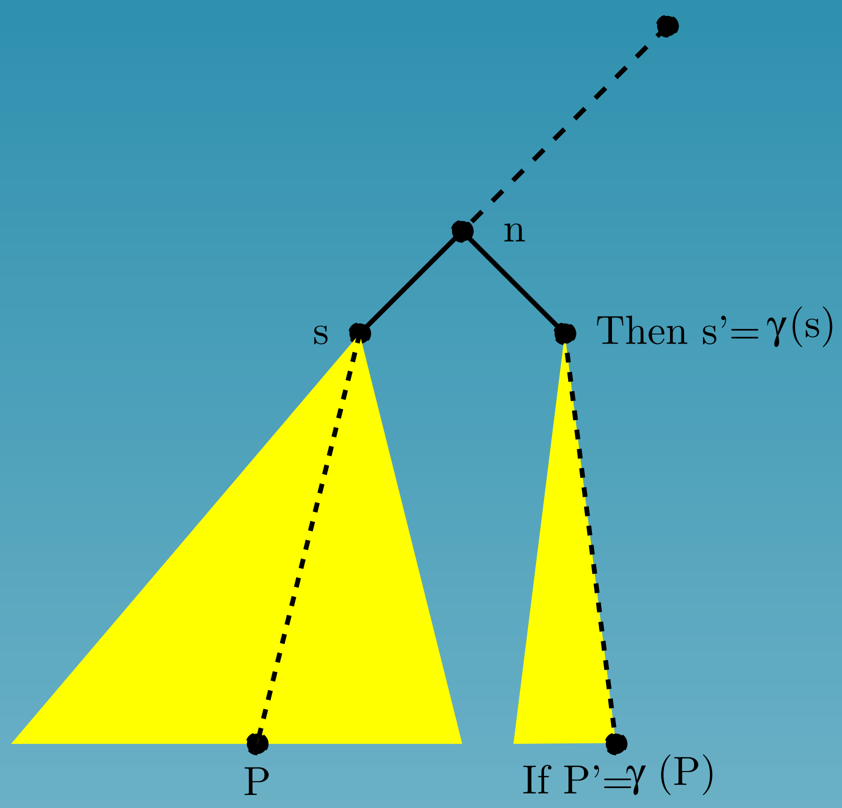
Mc Kay Pruning

Suggested in [Mc Kay, 81] for search for graph isomorphism



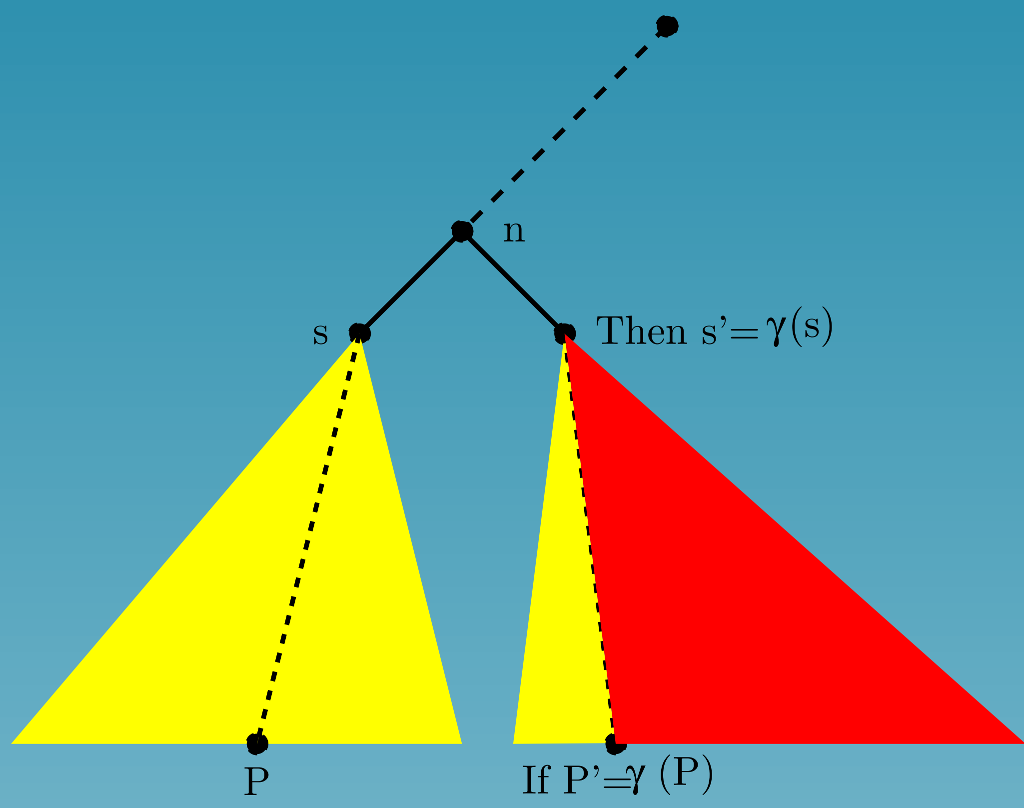
MC KAY Pruning

Suggested in [Mc Kay, 81] for search for graph isomorphism



Mc Kay Pruning

Suggested in [Mc Kay, 81] for search for graph isomorphism



Mc Kay Pruning for Golfer Problem

"Mc Kay property" requires a "compatibility" between

- Symmetry structure
- Search tree structure

Mc Kay **Pruning for Golfer Problem**

"Mc Kay property" requires a "compatibility" between

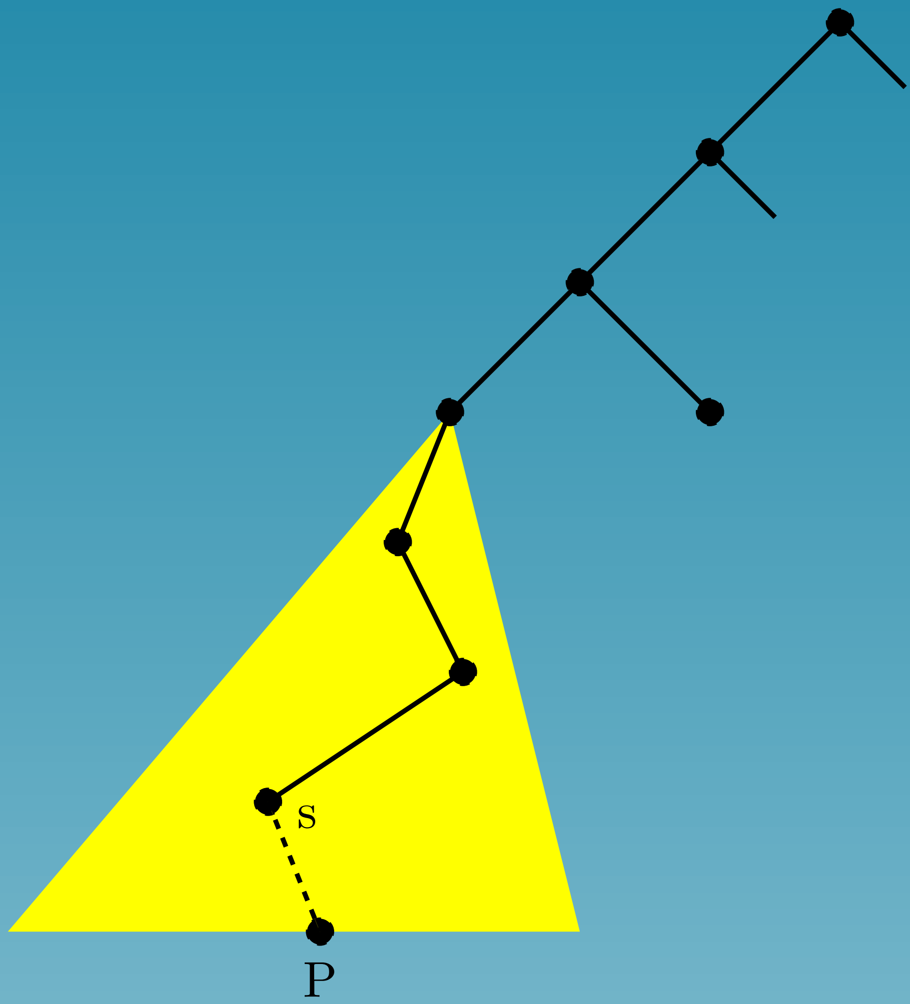
- Symmetry structure
- Search tree structure

For the Golfer Problem:

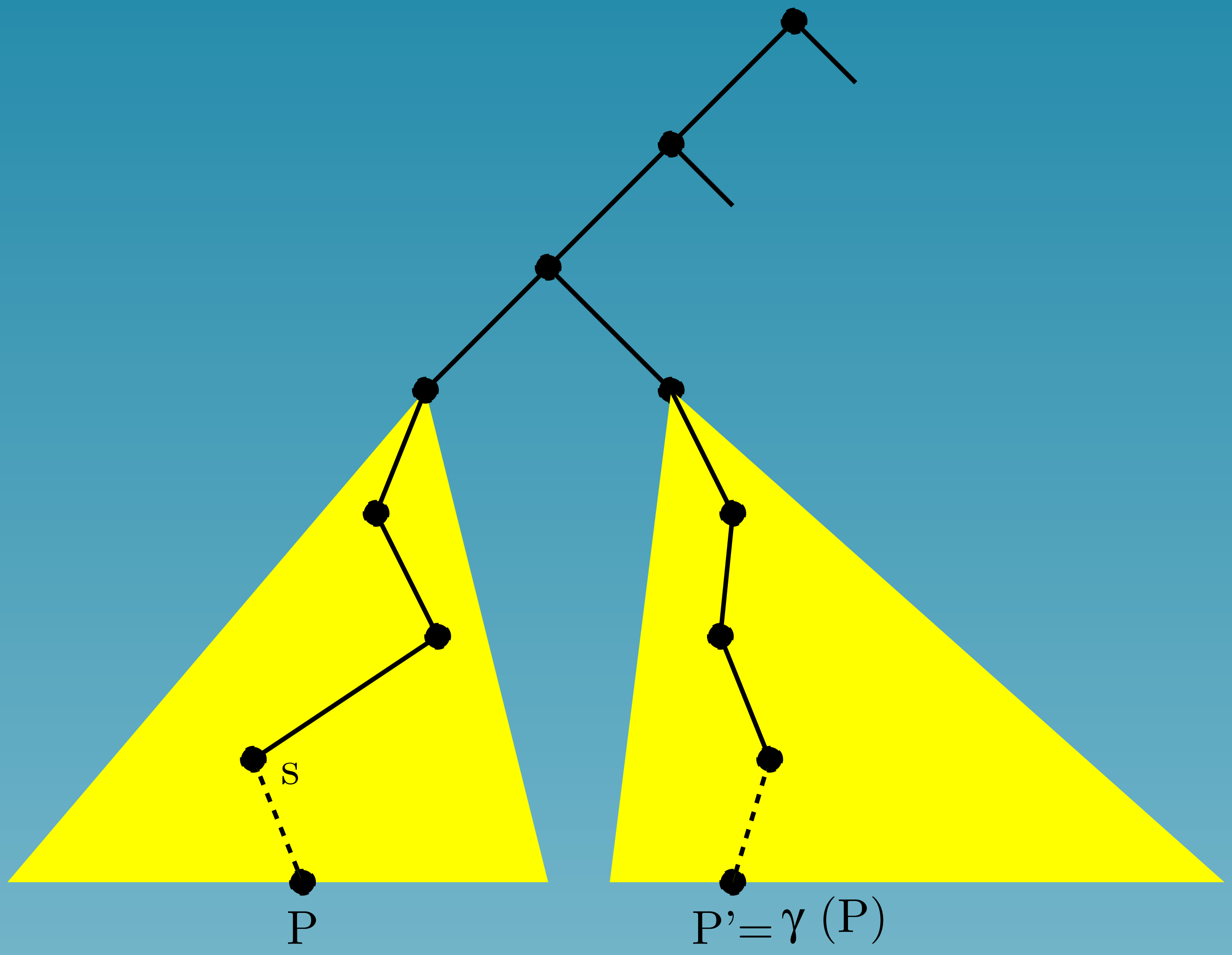
- Symmetry among golfer's "name"
- Labeling per golfers

Necessary but not enough: set of golfers above the choice point must be *stable* through the symmetry γ

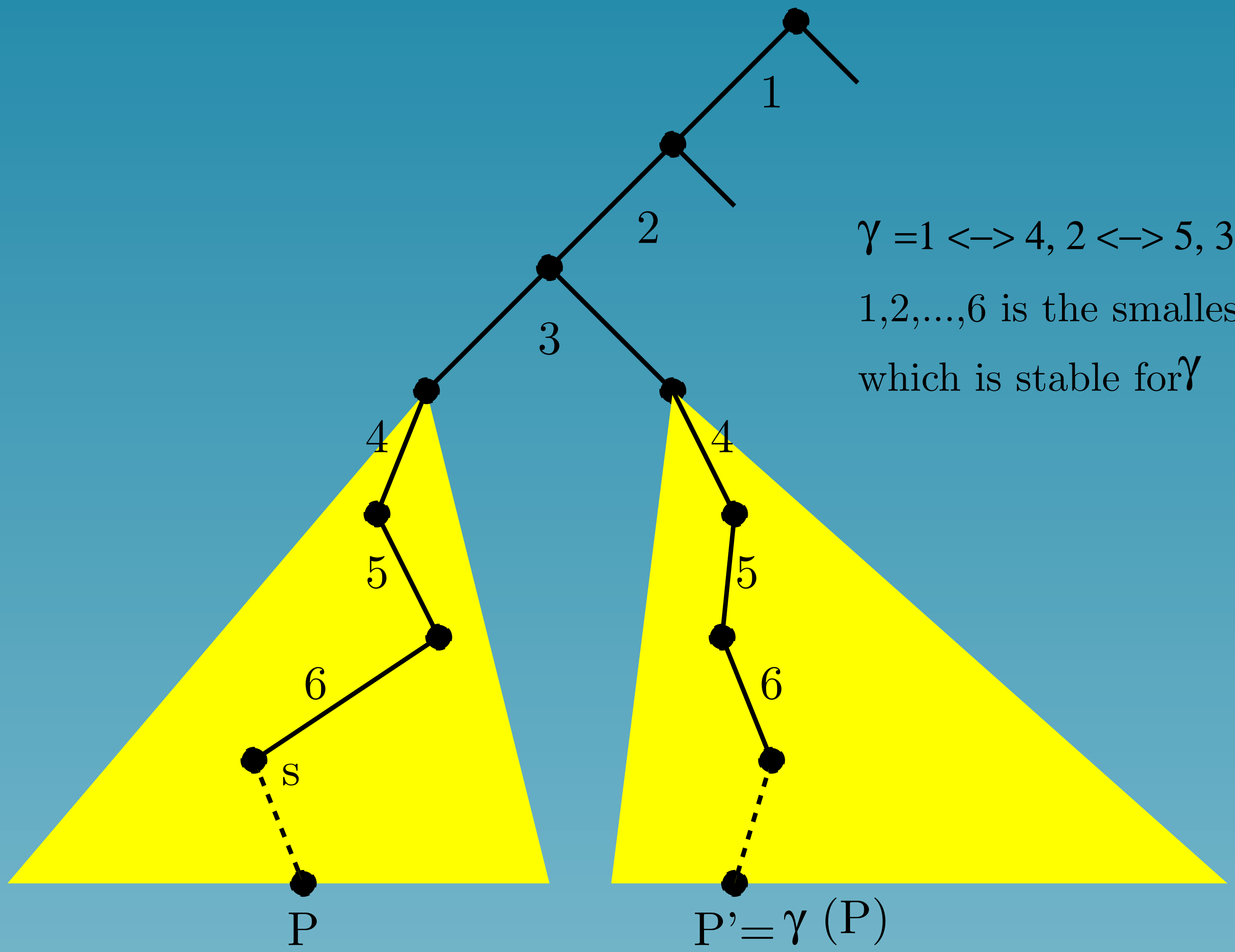
McKay Pruning Revisited



McKay Pruning Revisited



MC KAY **Pruning Revisited**

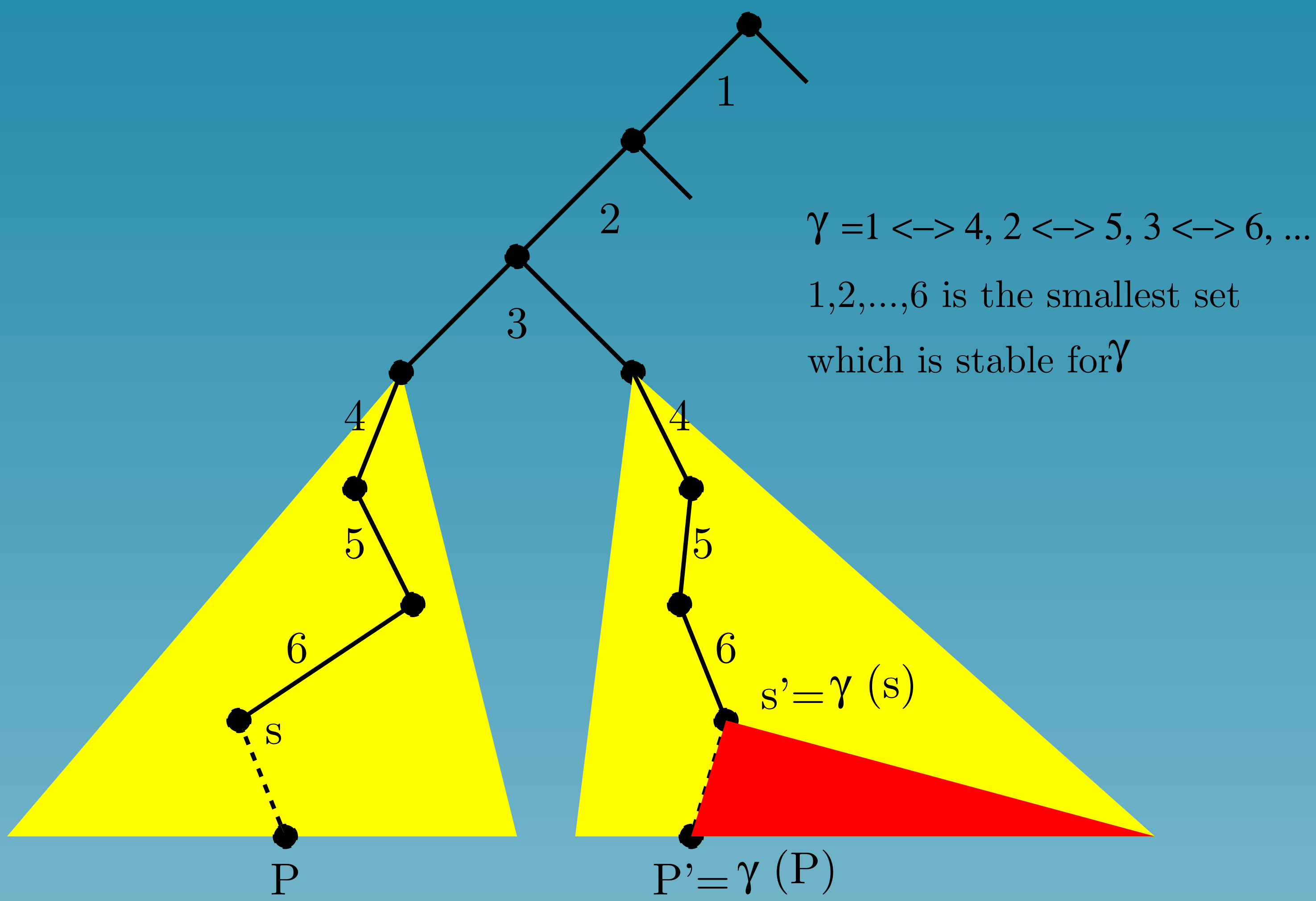


γ =1 <–> 4, 2 <–> 5, 3 <–> 6, ...

1,2,...,6 is the smallest set

which is stable for γ

Mc Kay **Pruning Revisited**



Results

Strong reduction of the search tree and the CPU-time.

	Leaves	McKay
Choice points	20 062 206	1 845 543
Fails	19 491 448	1 803 492
Solutions	20 640	934
CPU(s)	5 925	484

SBDD [Sellmann, CP'01]

Symmetry Breaking via Dominance Detection: A (new) state P' is *dominated* by an (old) state P if P' is subsumed by $\phi(P)$ where ϕ is a symmetry mapping function.

SBDD [Sellmann, CP'01]

Symmetry Breaking via Dominance Detection: A (new) state P' is *dominated* by an (old) state P if P' is subsumed by $\phi(P)$ where ϕ is a symmetry mapping function.

$\longrightarrow$ A state which is dominated by an already explored one can be discarded.

SBDD [Sellmann, CP'01]

Symmetry Breaking via Dominance Detection: A (new) state P' is *dominated* by an (old) state P if P' is subsumed by $\phi(P)$ where ϕ is a symmetry mapping function.

$\longrightarrow$ A state which is dominated by an already explored one can be discarded.

$\longrightarrow$ Explored nodes must be stored

SBDD [Sellmann, CP'01]

Symmetry Breaking via Dominance Detection: A (new) state P' is *dominated* by an (old) state P if P' is subsumed by $\phi(P)$ where ϕ is a symmetry mapping function.

$\longrightarrow$ A state which is dominated by an already explored one can be discarded.

$\longrightarrow$ Explored nodes must be stored

If P dominates P' then father of P dominates P'

SBDD [Sellmann, CP'01]

Symmetry Breaking via Dominance Detection: A (new) state P' is *dominated* by an (old) state P if P' is subsumed by $\phi(P)$ where ϕ is a symmetry mapping function.

$\longrightarrow$ A state which is dominated by an already explored one can be discarded.

$\longrightarrow$ Explored nodes must be stored

If P dominates P' then father of P dominates P'

$\longrightarrow$ If all sons of P have been explored (and stored), they can be removed from the store and replaced by P

SBDD [Sellmann, CP'01]

Symmetry Breaking via Dominance Detection: A (new) state P' is *dominated* by an (old) state P if P' is subsumed by $\phi(P)$ where ϕ is a symmetry mapping function.

$\longrightarrow$ A state which is dominated by an already explored one can be discarded.

$\longrightarrow$ Explored nodes must be stored

If P dominates P' then father of P dominates P'

$\longrightarrow$ If all sons of P have been explored (and stored), they can be removed from the store and replaced by P

$\longrightarrow$ "Only" gsg^w states to store for the g-s-w instance in our case (12 890 625 for 5-3-7)

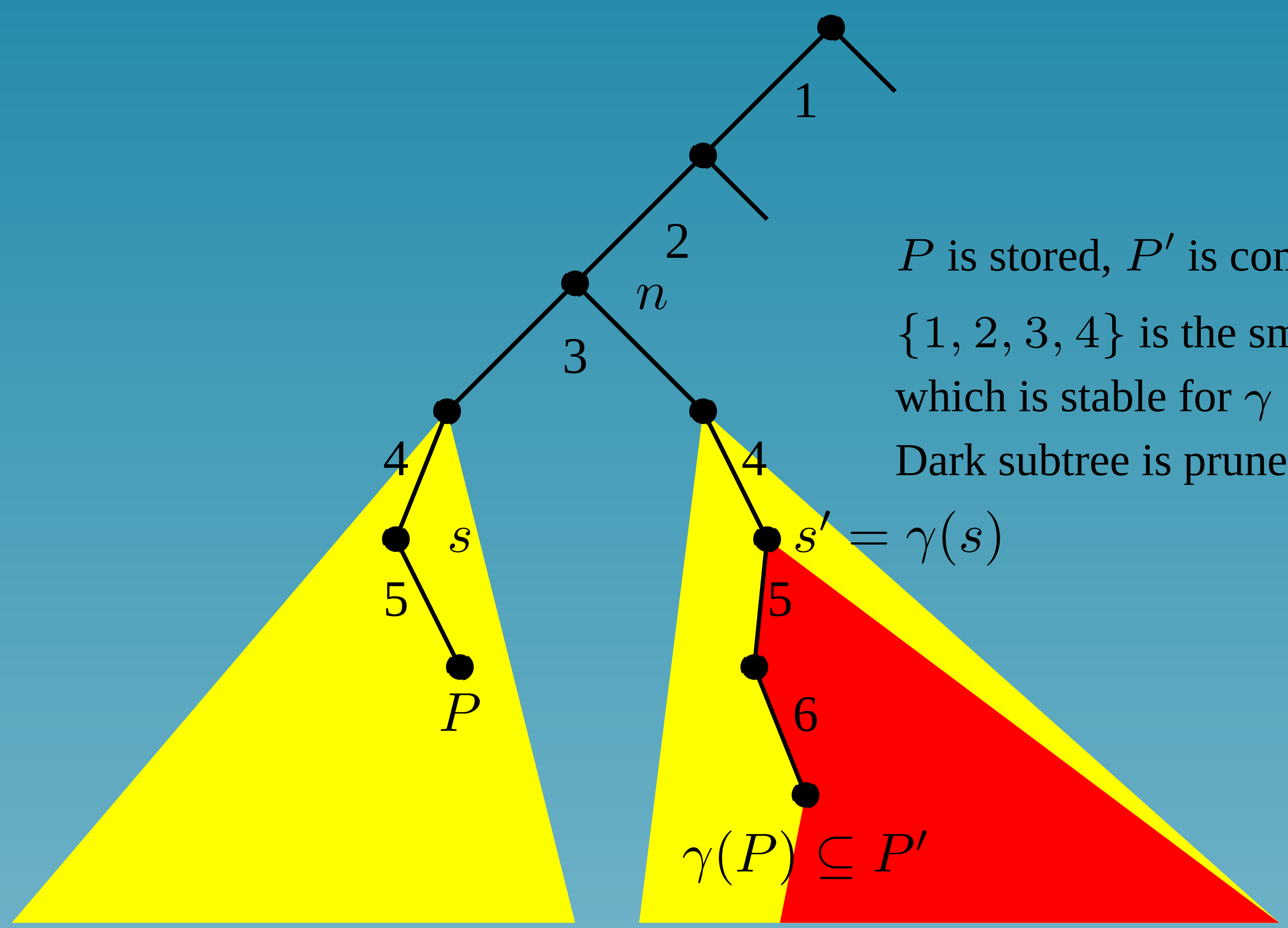
SBDD for Golfer Problem

Dominance checking remains expensive

- Check only symmetries which map first week on itself
- Check frequency must be related to the structure of the problem
  - Store nodes only after all choices for one golfer are made;
  - Check dominance for nodes only against stored nodes of smaller depth;
  - Check dominance only for nodes at depth multiple of s (size of groups)

$\longrightarrow$ Never more than 15 nodes in the store.

SBDD+: SBDD + McKay



P is stored, P' is compared with P

$\{1, 2, 3, 4\}$ is the smallest set which is stable for γ

Dark subtree is pruned

Results

	Leaves	McKay	SBDD	SBDD+
Choice points	20062206	1845543	107567	29954
Fails	19491448	1803492	104134	28777
Solutions	20640	934	11	11
Dominance checks			5373	456
CPU(s)	5925	484	24	7.8

Results

	Leaves	McKay	SBDD	SBDD+
Choice points	20062206	1845543	107567	29954
Fails	19491448	1803492	104134	28777
Solutions	20640	934	11	11
Dominance checks			5373	456
CPU(s)	5925	484	24	7.8

"New Results": $6\text{-}4\text{-}6$, $7\text{-}3\text{-}9$, $8\text{-}3\text{-}7$, $7\text{-}4\text{-}6$, $6\text{-}5\text{-}7$

Less Choice-Points

A player plays only once per week:

$$1 \leq i \leq w, 1 \leq p \leq n \qquad \sum_{1 \leq j \leq g} (p \in G_{i,j}) = 1 \tag{1}$$

Players of a group appear in exactly s groups in other weeks (W. Harvey):

$$1 \leq i \neq i' \leq w,\ 1 \leq j \leq g \qquad \sum_{1 \leq j' \leq g} (G_{i,j} \cap G_{i',j'} \neq \emptyset) = s \tag{2}$$

	SBDD+	+(1)	+(2)
Choice points	29954	18705	18470
Fails	28777	16370	16169
Solutions	11	11	11
Dominance checks	456	456	443
CPU(s)	7.8	9.4	36

Conclusion

Improving CPU-time requires:

- The right model
- Redundant constraints
- Breaking statically symmetries with constraints
- Breaking dynamically symmetries:
  - Efficient detection
  - Dominance detection
  - Analysis of the search tree for deep pruning

Conclusion

Improving CPU-time requires:

- The right model
- Redundant constraints
- Breaking statically symmetries with constraints
- Breaking dynamically symmetries:
  - Efficient detection
  - Dominance detection
  - Analysis of the search tree for deep pruning

Future work

- 8-4-10 instance
- Application of SBDD+ to "real" problems