Almost Square Packing

Helmut Simonis and Barry O'Sullivan

Cork Constraint Computation Centre
Computer Science Department
University College Cork
Ireland

CSCLP 2010

Points to Remember

- Almost Square Packing: Generalization of Square Packing (Simonis and O'Sullivan, CP 2008)
- Allows rotation of items, extra degree of freedom
- Interleave orientation of item with interval choice for X
- Redundant constraints similar to square packing
- Further decomposition possible, treat X variables on their own
- Does not need `disjoint2` and second `cumulative` constraint
- Problems up to N=26 solved optimally (very similar to square packing)

Almost Square Packing

- Consider almost squares, rectangles $n \times (n+1)$
- Pack all items 1×2, 2×3, ..., $n \times (n+1)$
- Into smallest possible rectangle
- Items are non-overlapping

Background

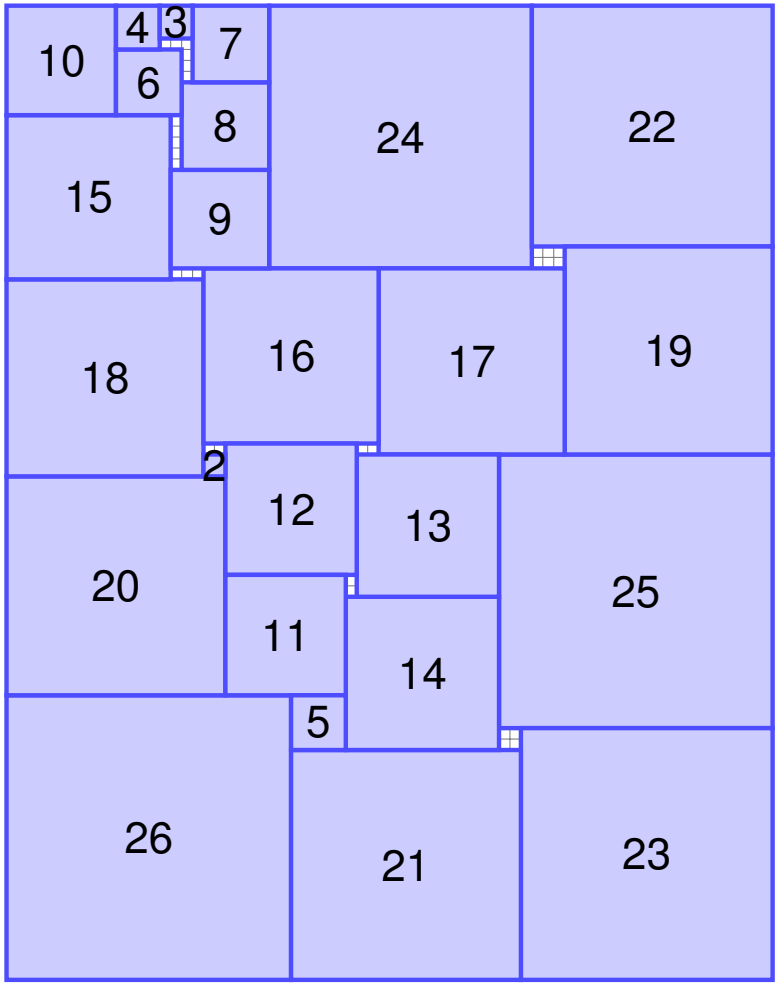
- Suggested by Prof. Des MacHale, Math, UCC
- Solved packing up to 13×14 by hand
- Starting point for Barry's and my interest in packing
- My objection: "This is too complicated"
  - Look at square packing
  - Known results, benchmark comparison

Simpler Problem: Square Packing

- Consider squares, rectangles $n \times n$
- Pack all items 1×1, 2×2, ..., $n \times n$
- Into smallest possible rectangle
- Items are non-overlapping

Outline

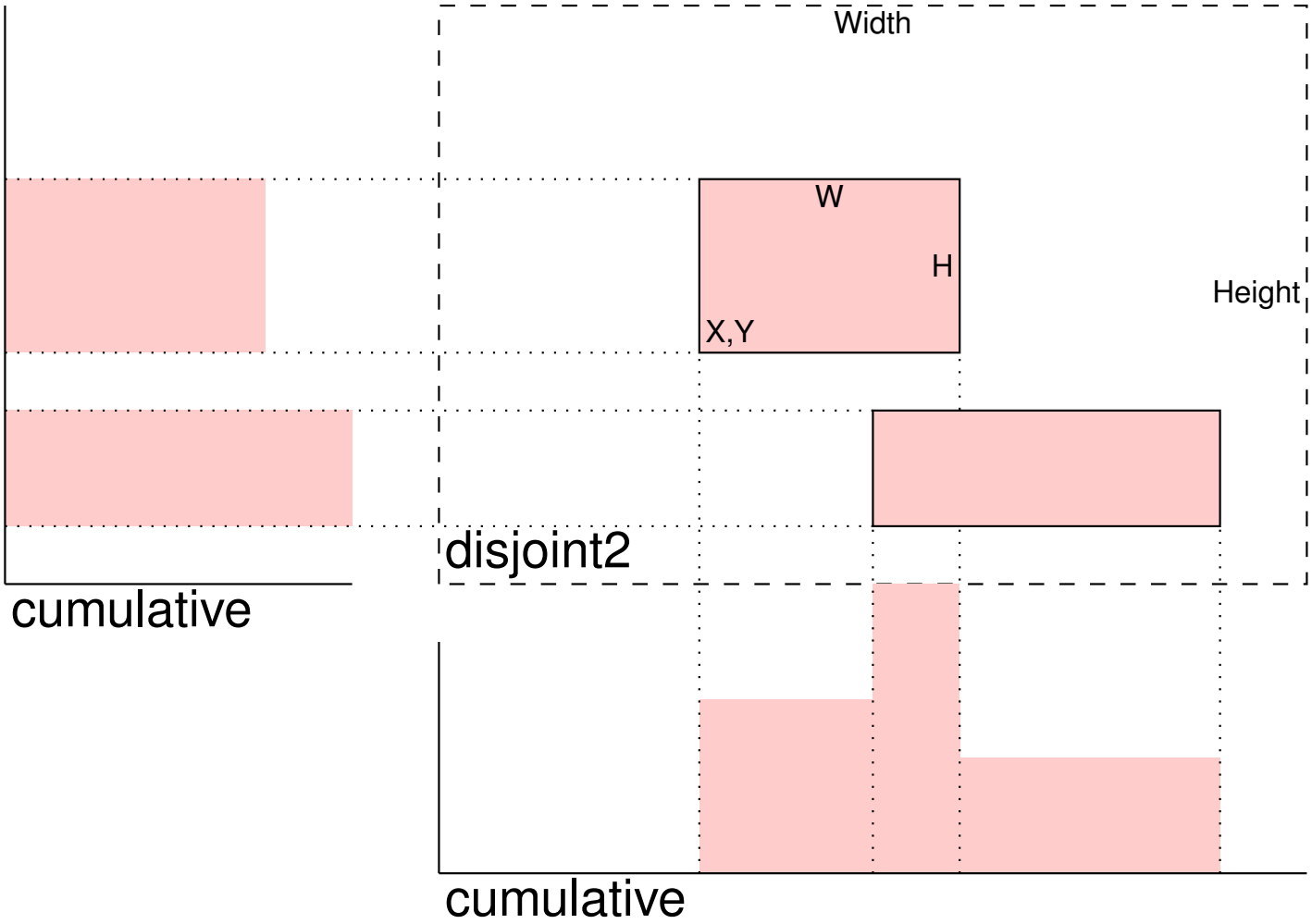
Problem (N=26)



Problem Decomposition

- Search for candidate enclosing rectangle
- Area must be larger than sum of items to be placed
- Search in order of increasing area
  - and increasing “squareness”
- Check each candidate for (in)feasibility until first solution is found
- Observation: Only limited number of candidates explored

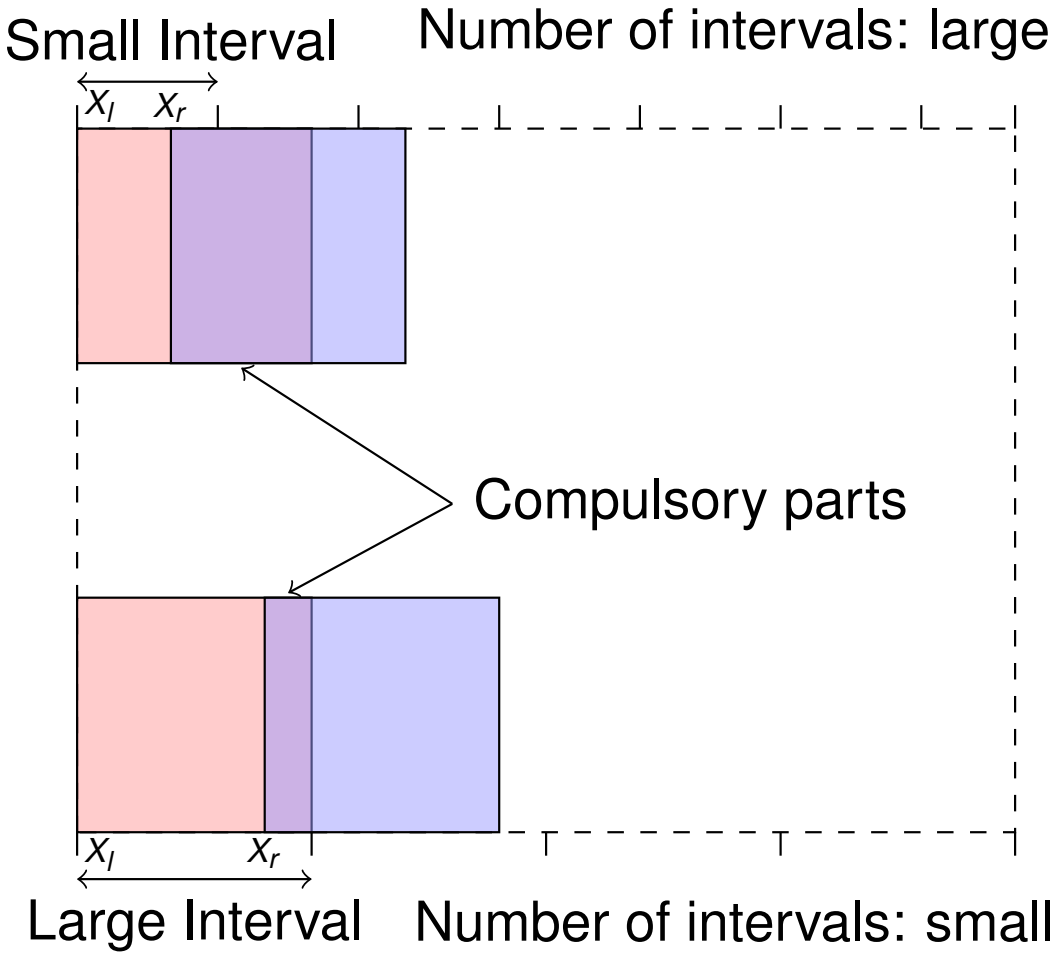
Basic Model



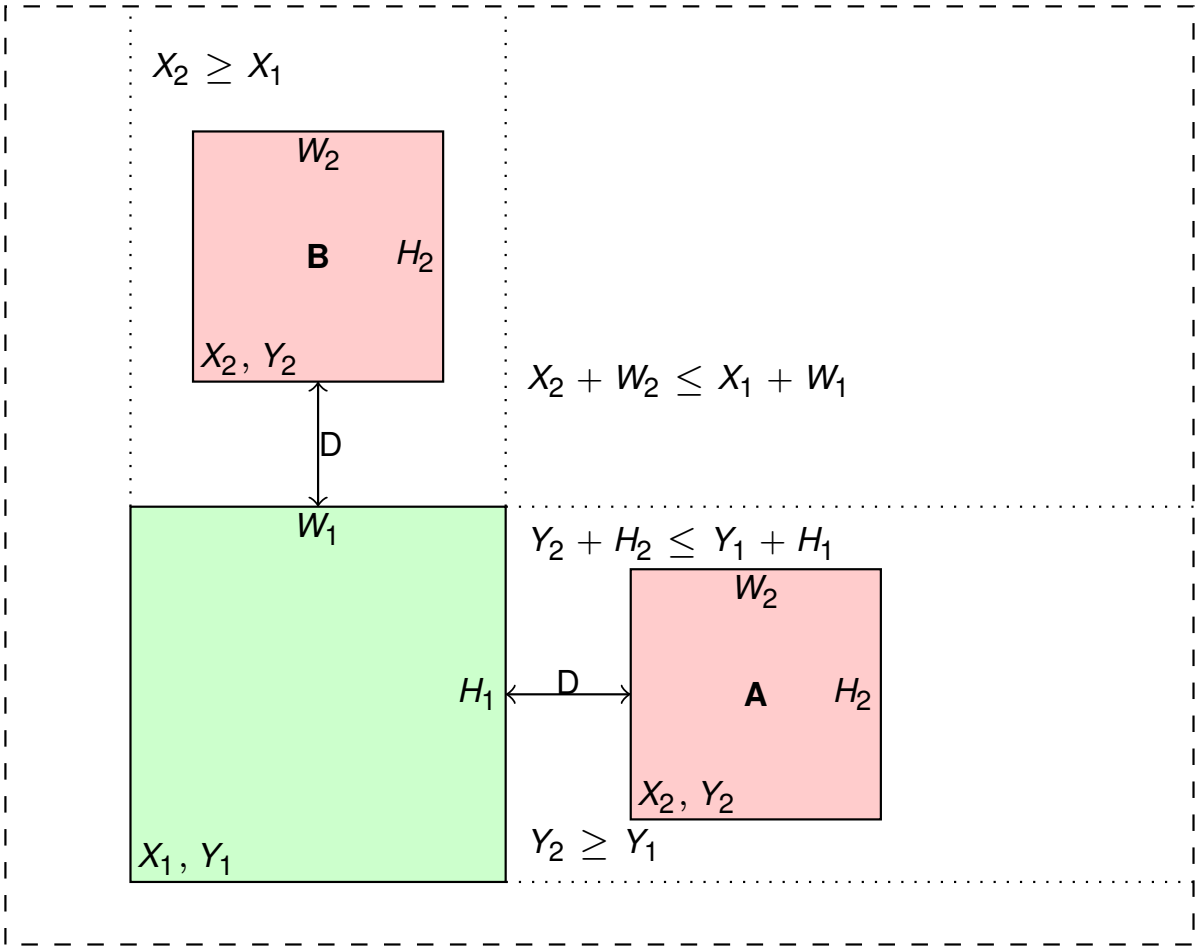
Interval Based Strategies

- Key Idea: Fixing intervals, not values
- Fixing variables to values is too restrictive
- Select “area” in which item is placed
- Allows items to shift slightly
- Restrict domain to intervals
- Only at end fix actual values

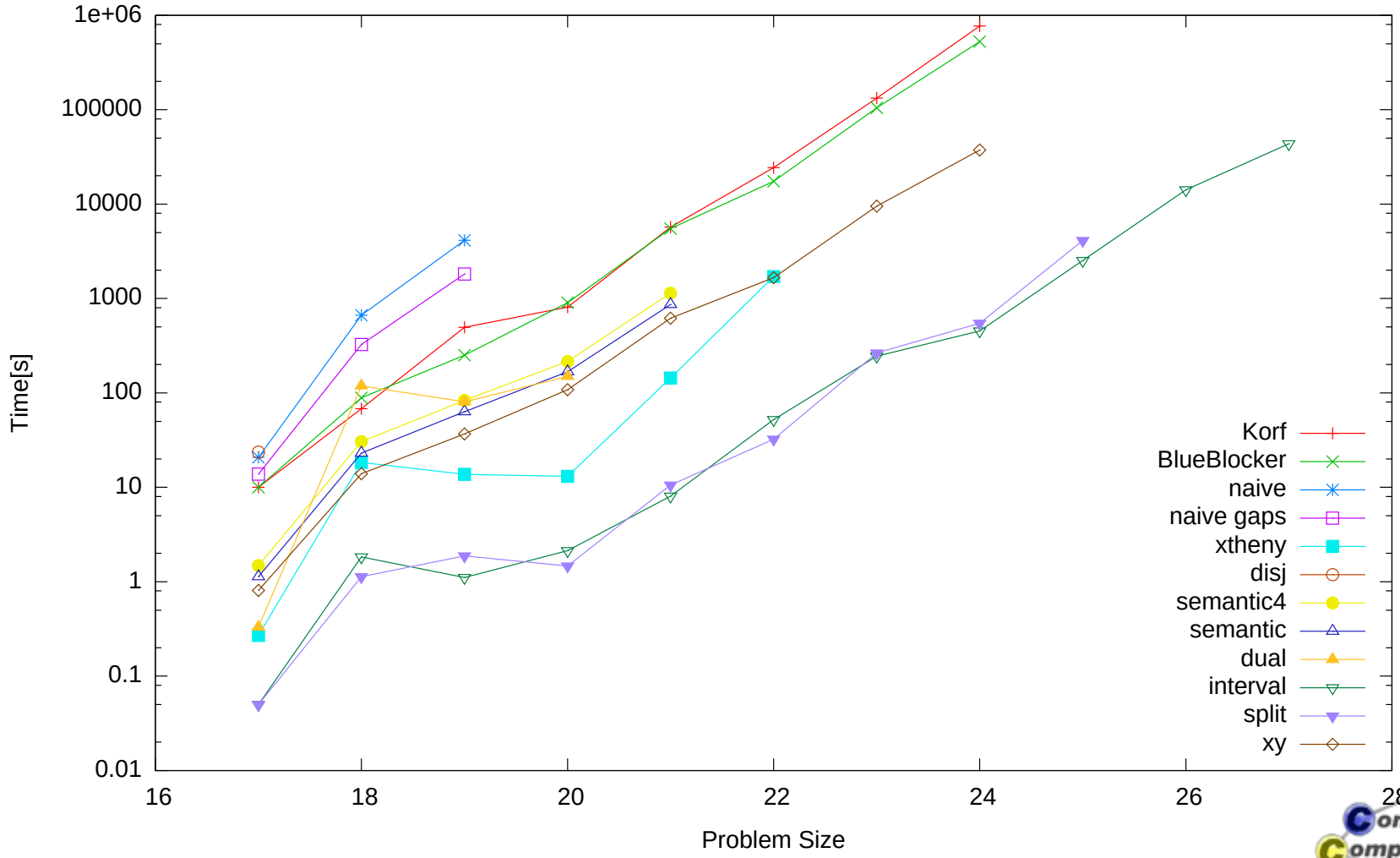
Forcing Compulsory Parts



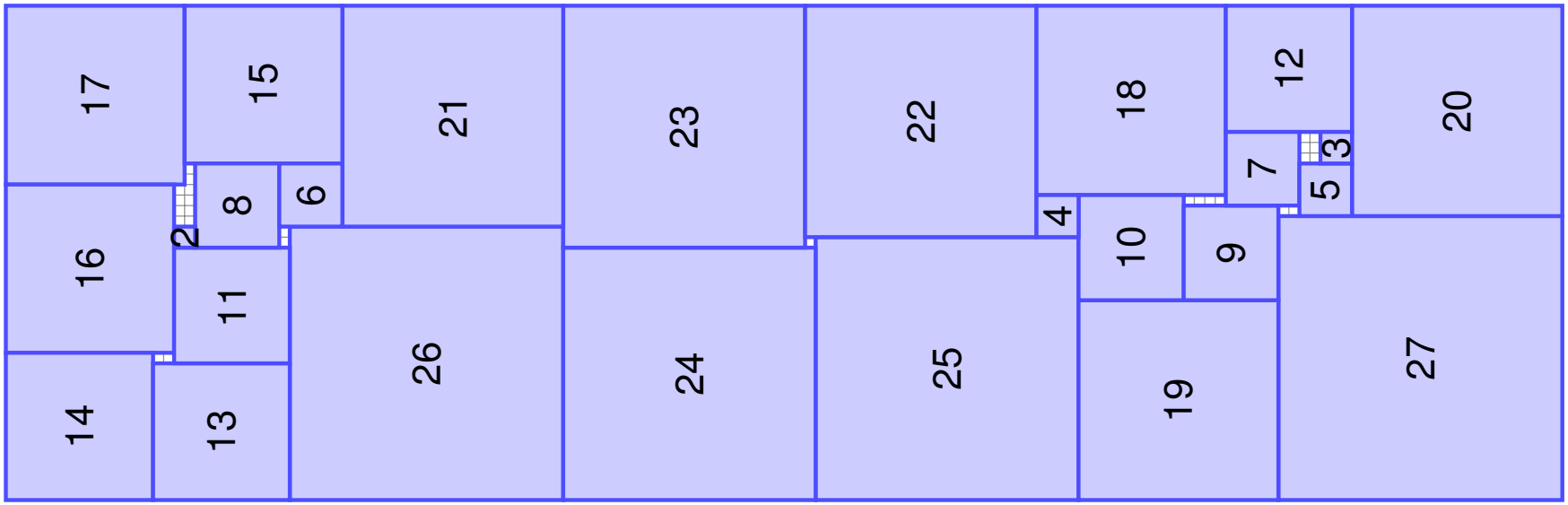
Model Improvement: Dominance Criterion



Strategies Comparison



Optimal Solution (N=27, CP 2008)



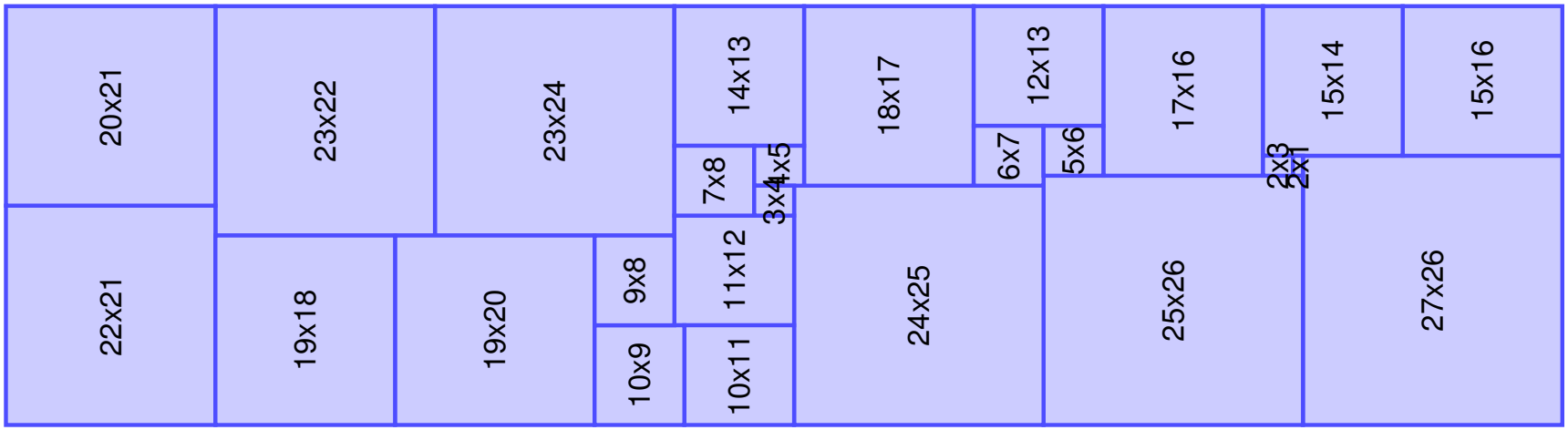
- Even better results by R. Korf in IJCAI 2009

Outline

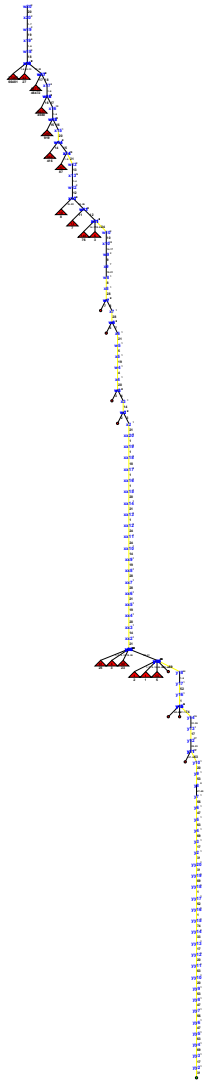
Back to the Future

- Apply lessons learned to almost square packing
- Added degree of freedom, rotation of items
- Weak impact, length only changes by one
- But 2^n additional choices

Almost Square, Optimal Solution (N=26)



Visualization: Search Tree (N=20)



Problem with Tree View

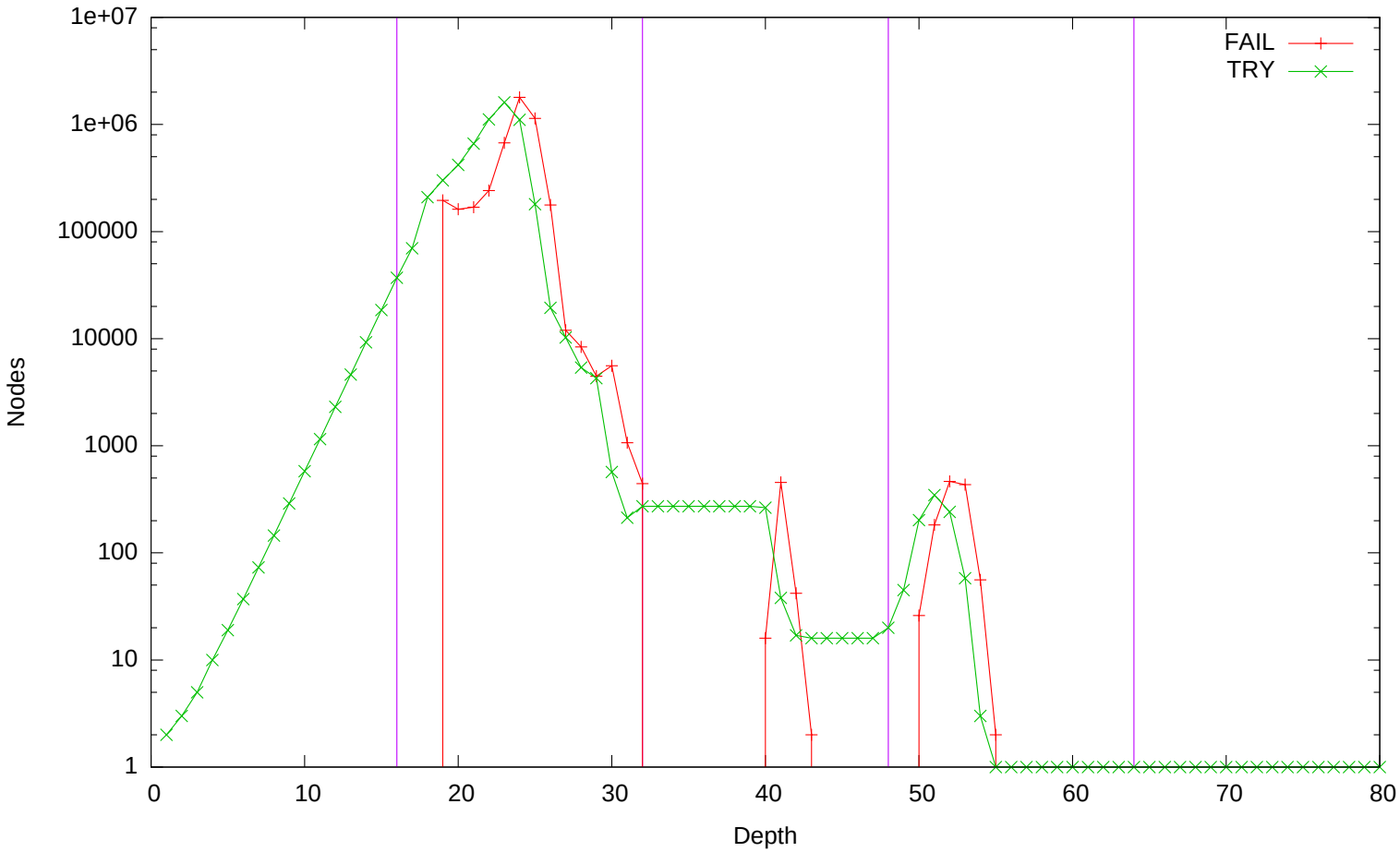
- Showing complete tree is clearly infeasible
- We only show path to solution
- But most time is spent in non-solution parts of tree

Search Choices

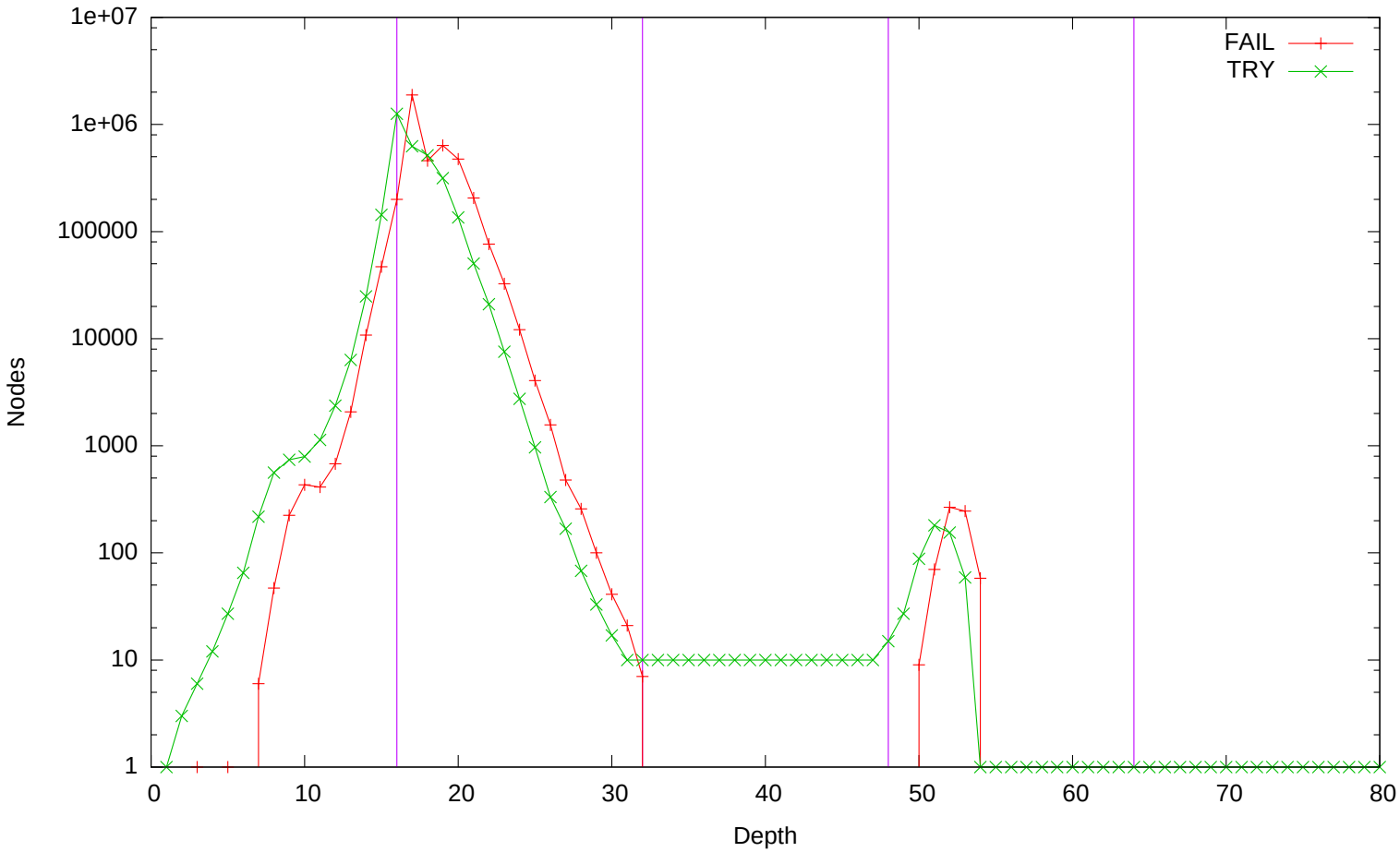
- Based on best method for square packing
- Assign X intervals, fix X values, assign Y intervals, fix Y values
- When to fix orientation?

Eager	Before assigning X intervals
Lazy	After assigning X intervals
Interleaved	Mixed with X interval assignment

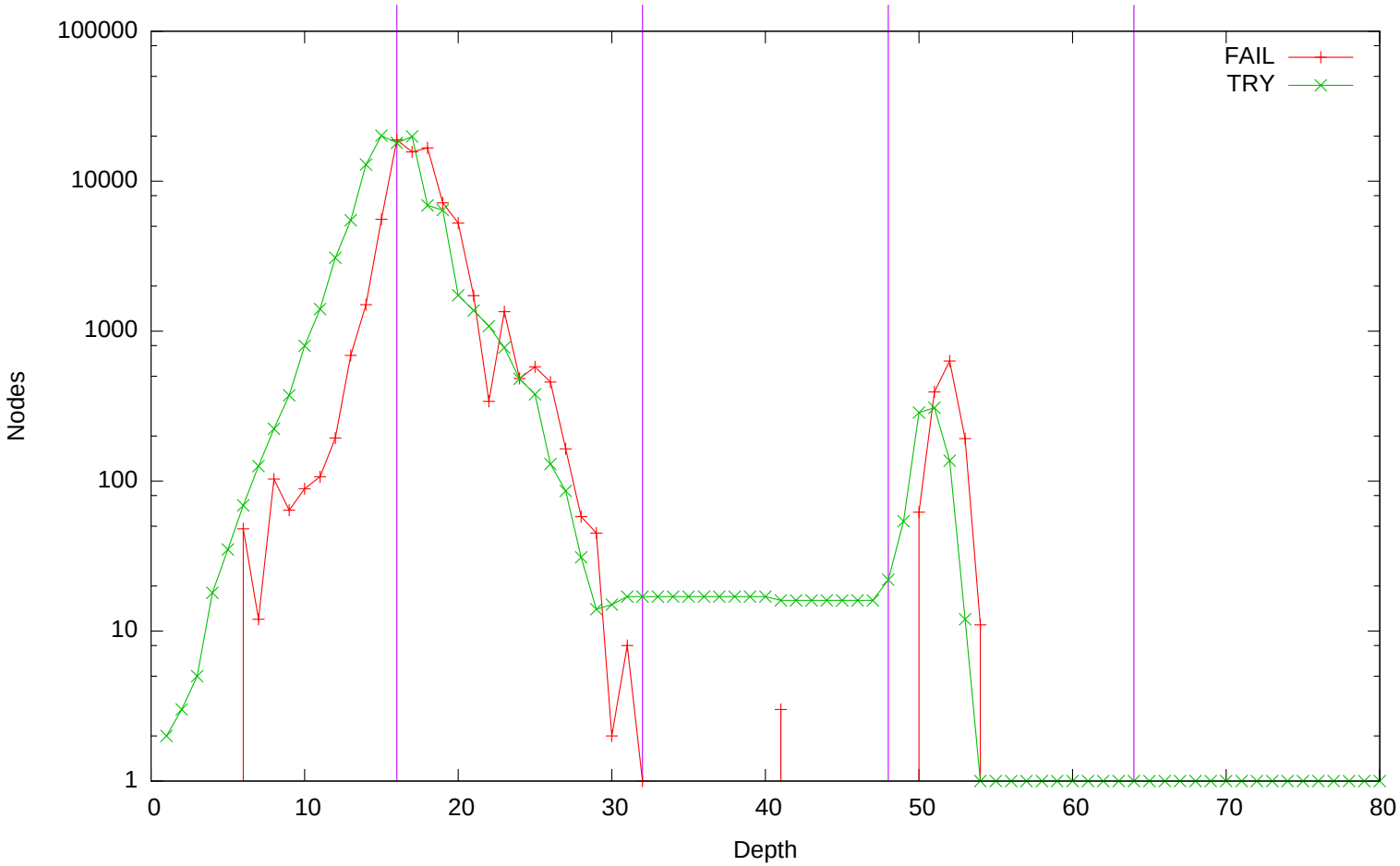
Eager Orientation (N=17)



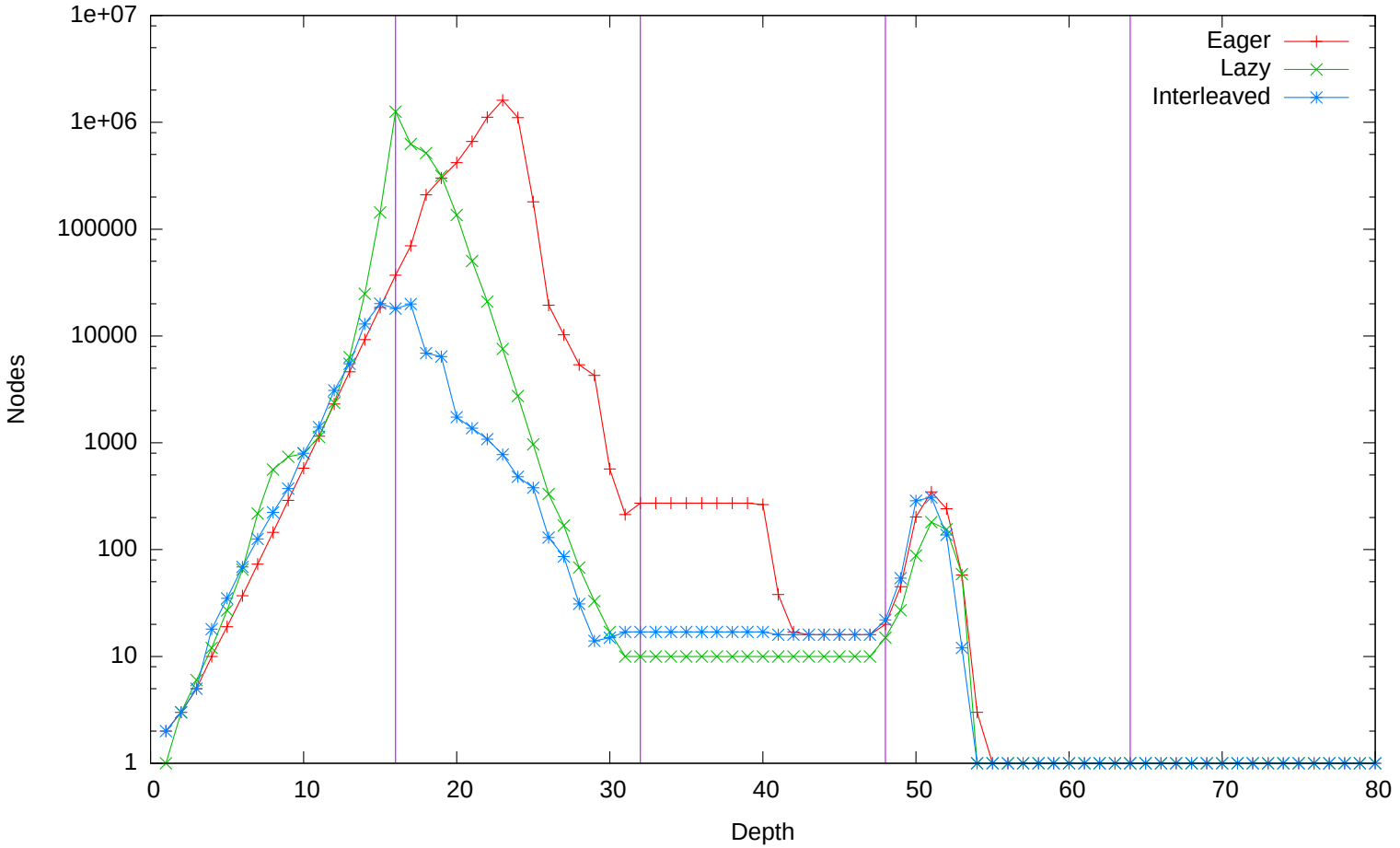
Lazy Orientation (N=17)



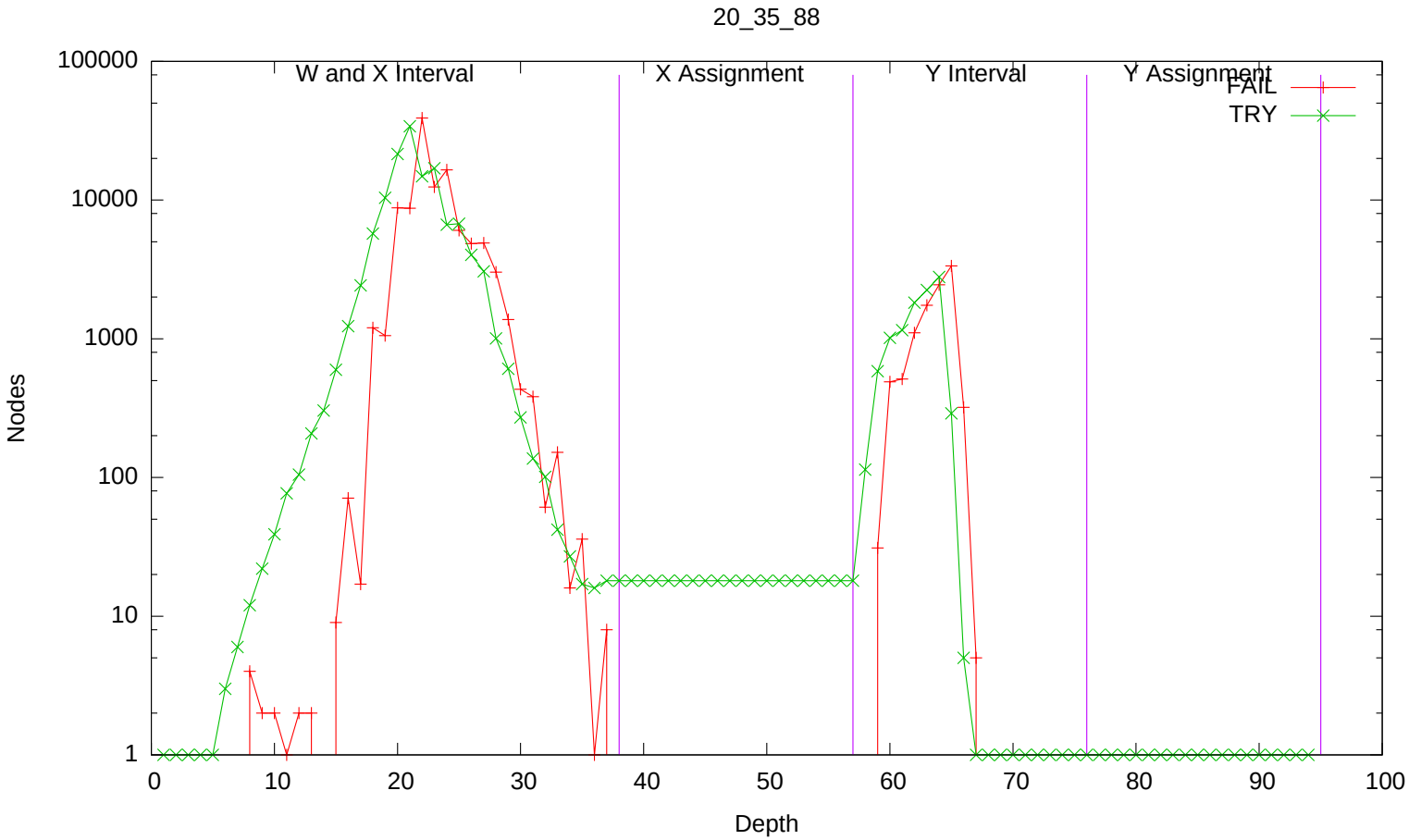
Interleaved Orientation (N=17)



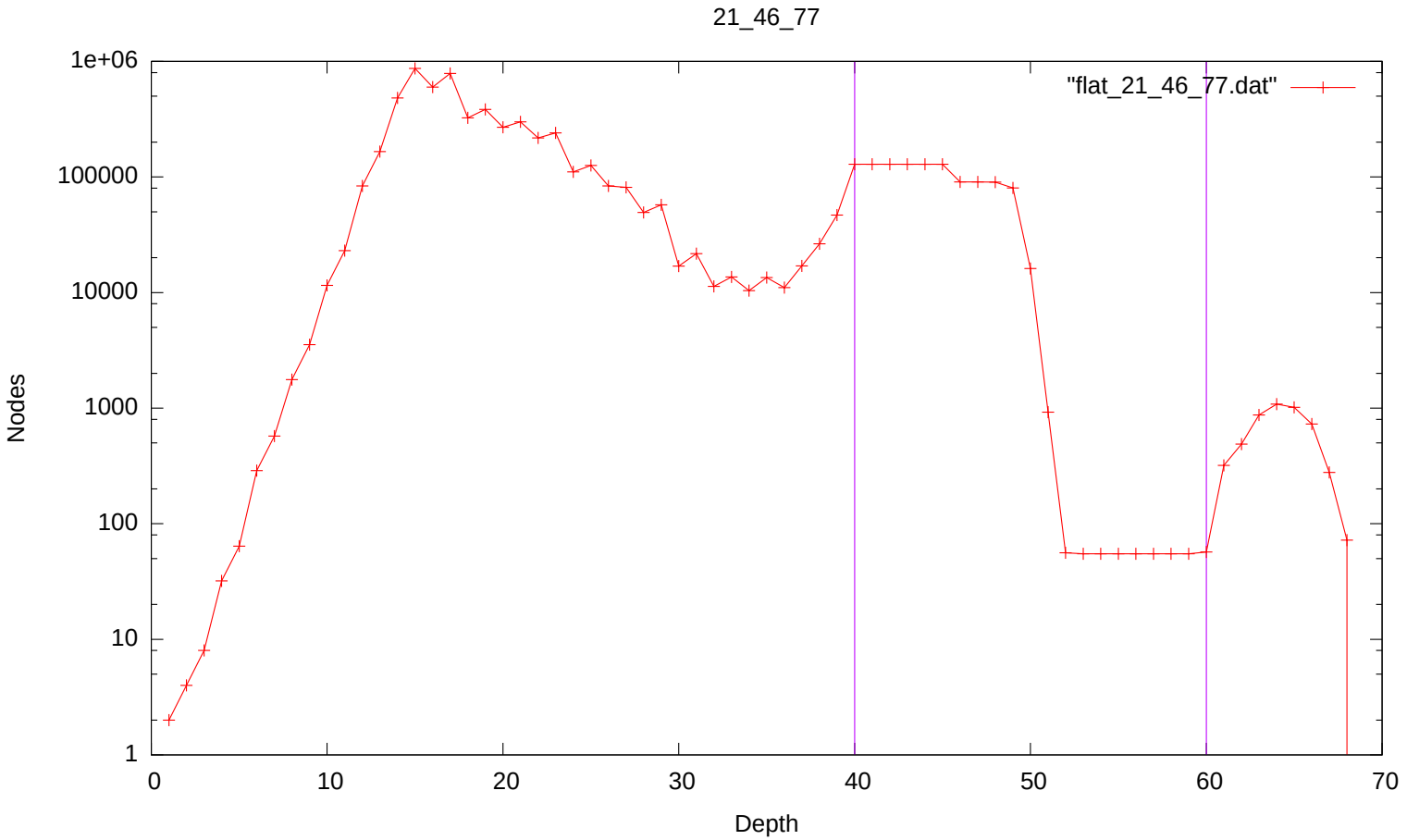
Comparison (N=17)



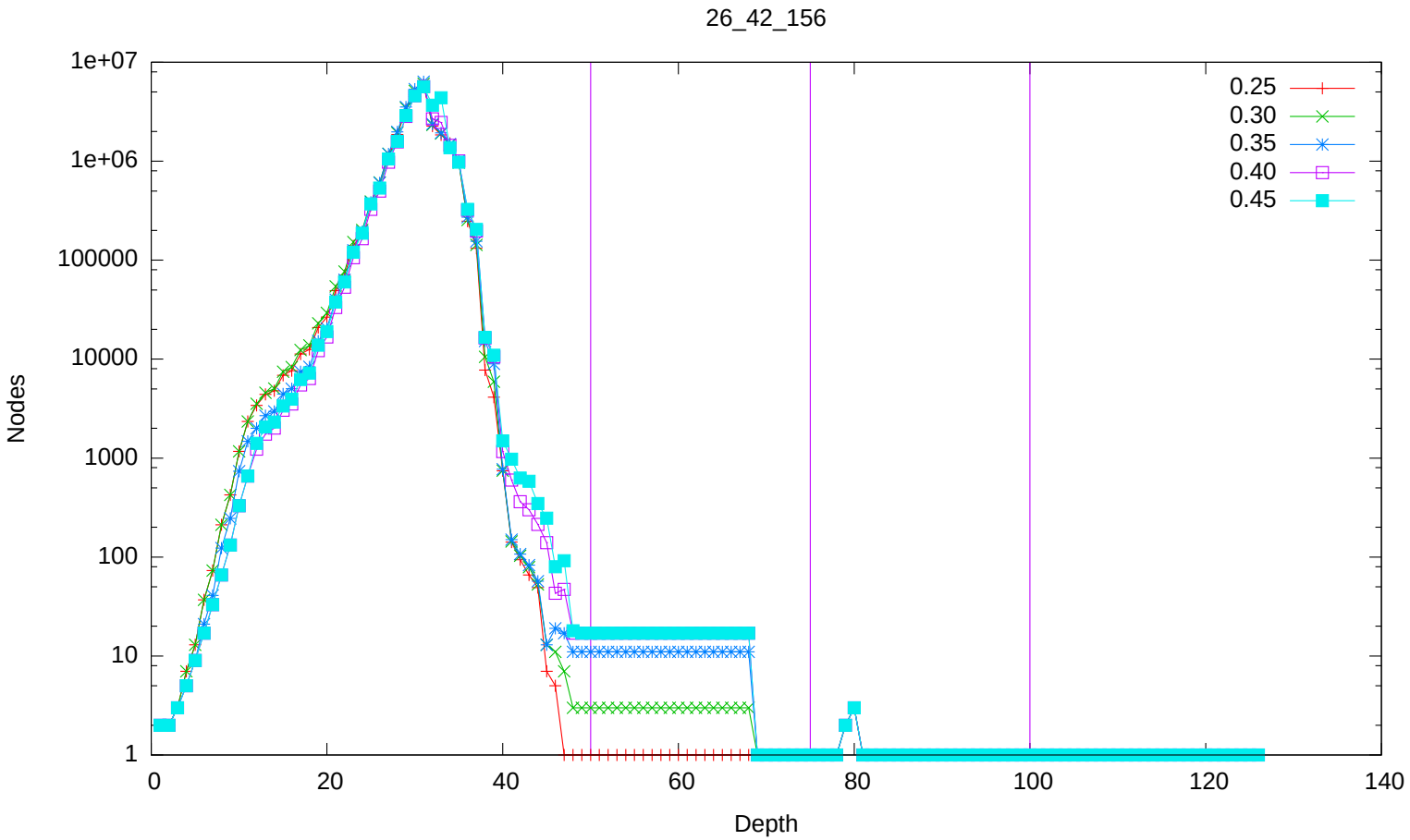
Node Distribution (N=20)



Problem with Slack (N=21)



Impact of Interval Size (N=26)



Basic Model Results (SICStus Prolog 4.04), 2008 Desktop

N	Loss	First Solution		All Solutions		
		Back	Time	Sols	Back	Time
4	0.00	2	00:00	8	6	00:00
5	0.00	4	00:00	16	14	00:00
6	1.79	16	00:00	216	24	00:00
7	0.00	19	00:00	65	76	00:00
8	0.00	6	00:00	12	83	00:00
9	1.82	54	00:00	9170	3137	00:00
10	0.45	323	00:00	1854	1379	00:00
11	0.00	99	00:00	4	268	00:00
12	0.96	546	00:00	25180	13795	00:02
13	0.00	1900	00:00	42	6197	00:00
14	0.00	2937	00:00	4	9604	00:00
15	0.00	14440	00:00	4	50592	00:03
16	0.00	15967	00:01	544	48711	00:03
17	0.00	210878	00:14	16	398759	00:27
18	0.00	9734	00:00	110288	152032	00:24
19	0.00	102235	00:08	526	3240741	04:26
20	0.00	351659	00:34	1988	3612859	05:52
21	0.20	14036353	21:38	3250117	720146935	25:13:20
22	0.00	58206362	01:37:30	688	122563947	03:23:19
23	0.00	14490682	30:12	6784	136039535	04:38:40
24	0.00	27475258	55:05	96	99731414	03:20:37
25	0.00	35282646	01:23:12	*1007780*		
26	0.00	92228265	03:28:20	*1056*		

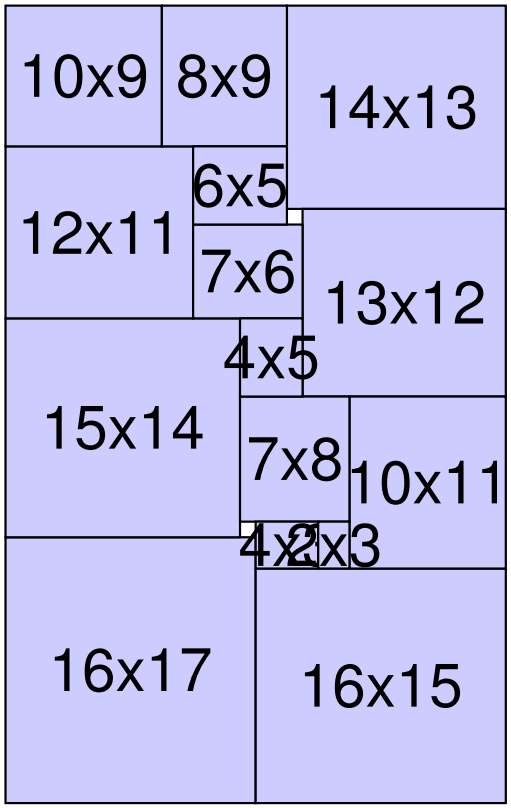
Redundant Constraints

- Forbidden gaps
- Ignore small blocks
- Symmetries of aligned blocks(?)

Forbidden Gaps

- Large items can be placed either directly on border or away from border
- Otherwise gap must be filled with small items and slack
- For each gap size, pre-compute length of item that can not be covered
- Slight complication due to rotation of items, use constructive disjunction
  - Remove most forbidden gaps at setup by reducing domains
  - Further reduction when item is oriented
- Does not handle interaction of multiple gaps

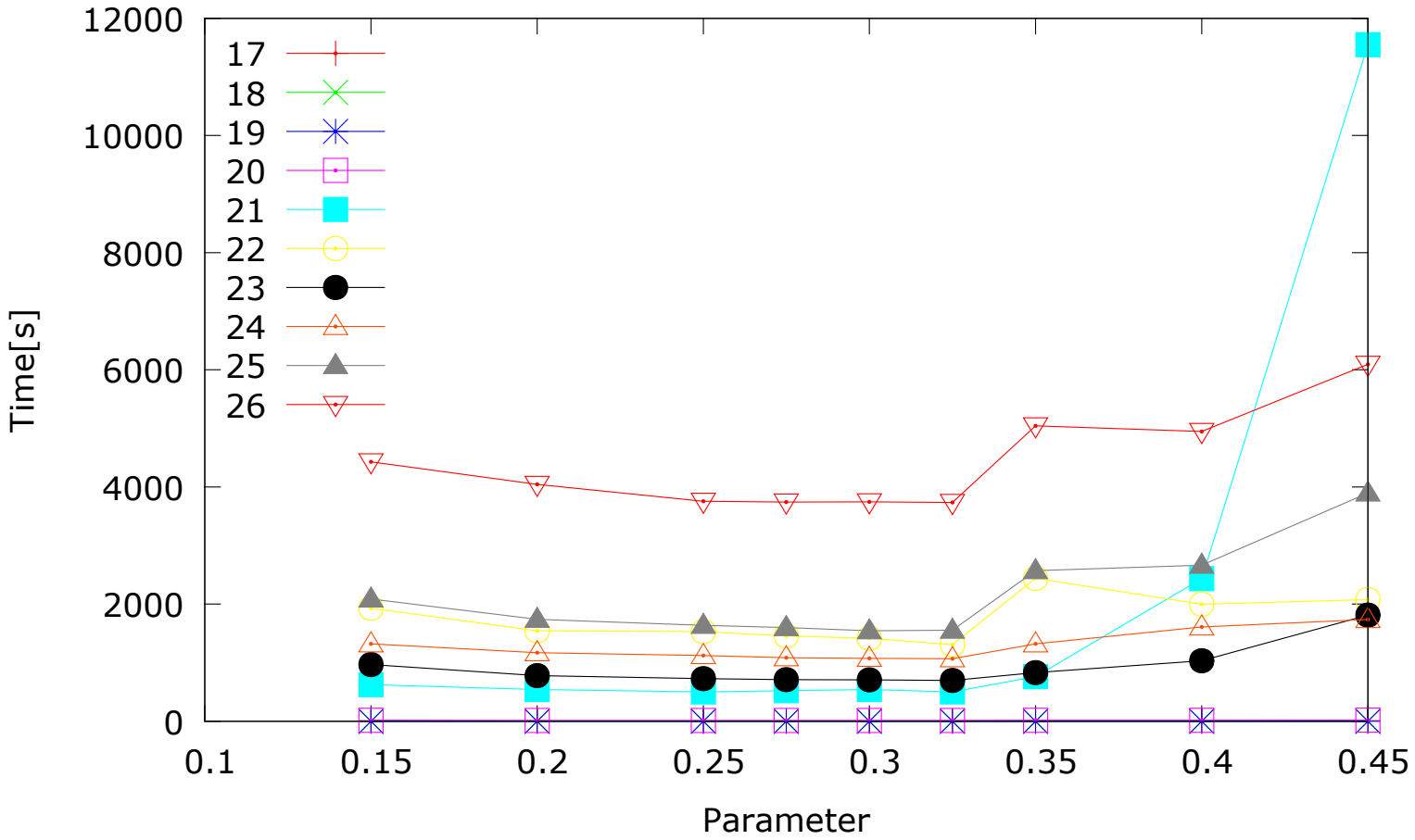
Pseudo Solution Removing 1 × 2 Block



Redundant Constraint Result

N	Basic Model		Not One		Gap		Both	
	Back	Time	Back	Time	Back	Time	Back	Time
4	2	00:00	2	00:00	2	00:00	2	00:00
5	4	00:00	3	00:00	2	00:00	1	00:00
6	16	00:00	16	00:00	6	00:00	6	00:00
7	19	00:00	18	00:00	10	00:00	9	00:00
8	6	00:00	5	00:00	17	00:00	10	00:00
9	54	00:00	54	00:00	27	00:00	27	00:00
10	323	00:00	323	00:00	159	00:00	159	00:00
11	99	00:00	99	00:00	54	00:00	54	00:00
12	546	00:00	546	00:00	274	00:00	274	00:00
13	1900	00:00	1900	00:00	1040	00:00	1040	00:00
14	2937	00:00	2936	00:00	1505	00:00	1501	00:00
15	14440	00:00	14425	00:00	7632	00:00	7617	00:00
16	15967	00:01	9338	00:00	7264	00:00	3989	00:00
17	210878	00:14	210850	00:13	107639	00:07	107611	00:07
18	9734	00:00	9734	00:00	5550	00:00	5550	00:00
19	102235	00:08	102235	00:08	13694	00:01	13690	00:01
20	351659	00:34	355964	00:33	157312	00:14	161410	00:14
21	14036353	21:38	10859861	16:01	9499957	14:14	6524396	09:13
22	58206362	01:37:30	58214183	01:33:03	17312971	24:37	17319946	23:54
23	14490682	30:12	14490682	29:16	6400629	11:01	6400629	10:33
24	27475258	55:05	27475258	53:11	9801577	16:39	9801577	16:10
25	35282646	01:23:12	35502799	01:21:25	13030167	25:16	13232221	25:15
26	92228265	03:28:20	92228259	03:22:33	29432477	55:38	29432467	54:08

Impact of Interval Size



Further Problem Decomposition

- Do we really need the full basic model?
- Most search in X variable search phase
- Postpone setting up Y variables, `disjoint2` and 2nd `cumulative`
- Only create them when complete solution for X variables found
- Uses just single `cumulative` constraint
- We can even replace `disjoint2` by reified `inequalities`

Decomposition Results

N	With Redundant Constraints					
	Basic Model		Decomposed Model		Decomposed Reified	
	Back	Time	Back	Time	Back	Time
4	2	00:00	2	00:00	2	00:00
5	1	00:00	1	00:00	1	00:00
6	6	00:00	6	00:00	6	00:00
7	9	00:00	9	00:00	9	00:00
8	10	00:00	10	00:00	10	00:00
9	27	00:00	27	00:00	27	00:00
10	159	00:00	176	00:00	176	00:00
11	54	00:00	54	00:00	54	00:00
12	274	00:00	301	00:00	301	00:00
13	1040	00:00	1040	00:00	1040	00:00
14	1501	00:00	1501	00:00	1501	00:00
15	7617	00:00	7617	00:00	7617	00:00
16	3989	00:00	3989	00:00	3989	00:00
17	107611	00:07	107611	00:05	107611	00:06
18	5550	00:00	5550	00:00	5550	00:00
19	13690	00:01	13690	00:00	13690	00:00
20	161410	00:14	161410	00:11	161410	00:16
21	6524396	09:13	6524396	07:10	6524396	08:05
22	17319946	23:54	17319946	19:13	17319946	21:50
23	6400629	10:33	6400629	08:04	6400629	08:58
24	9801577	16:10	9801577	12:11	9801577	13:07
25	13232221	25:15	13232221	20:07	13232773	23:34
26	29432467	54:08	29432467	40:26	29432467	43:51

Comparison with (Korf et al., 2010)

Size	Area	Korf, Moffitt and Pollack		Redundant		Decomposition	
		Nodes	Times	Back	Times	Back	Times
17	34×57	6,889,973	:07	107611	00:07	107611	00:05
18	30×76	22,393,428	:26	5550	00:00	5550	00:00
19	35×76	11,918,834	:11	13690	00:01	13690	00:00
20	35×88	608,635,198	12:50	161410	00:14	161410	00:11
21	39×91	792,197,287	23:21	6524396	09:13	6524396	07:10
22	44×92	4,544,585,807	1:49:32	17319946	23:54	17319946	19:13
23	40×115	32,222,677,089	15:06:56	6400629	10:33	6400629	08:04
24	40×130	41,976,042,836	18:39:34	9801577	16:10	9801577	12:11
25	45×130	557,540,262,189	12:11:30:32	13232221	25:15	13232221	20:07
26	42×156	-	-	29432467	54:08	29432467	40:26

Future Work

- SICStus Prolog cumulative does not know about volume of items
- Domain variables $[n, n+1]$ for duration and resource use
- Does not know that $w * h$ is $n * (n+1)$
- Underestimates volume as $n * n$
- Improved energetic reasoning possible (CHIP)

Points to Remember

- Almost Square Packing: Generalization of Square Packing (CP 2008)
- Interleave orientation of item with interval choice for X
- Further decomposition possible, treat X variables on their own
- Does not need `disjoint2` and second `cumulative` constraint
- Problems up to N=26 solved optimally (very similar to square packing)
- Outperforms best published results by 10^3