Search Strategies for Rectangle Packing

Helmut Simonis and Barry O'Sullivan

Cork Constraint Computation Centre
Computer Science Department
University College Cork
Ireland

CP 2008
Sydney, Australia

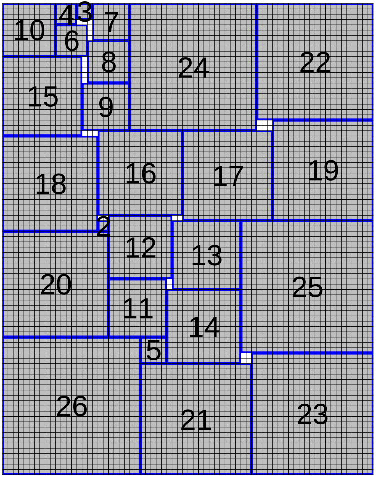
Outline

Outline

Overview

- Use of off-the-shelf constraint technology
- SICStus Prolog 4.02
- Evaluate different search strategies
- Problem: Packing all squares from 1×1 to $n \times n$ in smallest rectangle
- Difficulty: Aspect ratio of enclosing rectangle not known

Problem (N=26)



Motivation

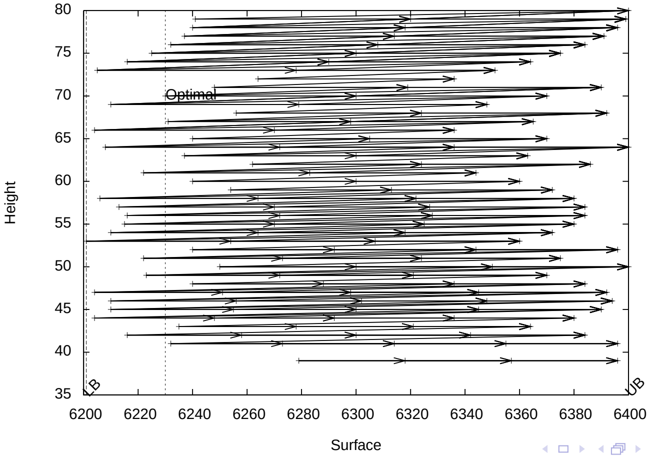
- Related to large-scale, real-life problems
  - IC floor planning, PCB design, architectural design
- Mix of feasible and infeasible subproblems
- Amount of slack changes with subproblem
- New results for open problems
  - Packing in Rectangle (N=26, 27)
  - Packing in Square (N=26, 27, 29, 30, 31, 35)

Outline

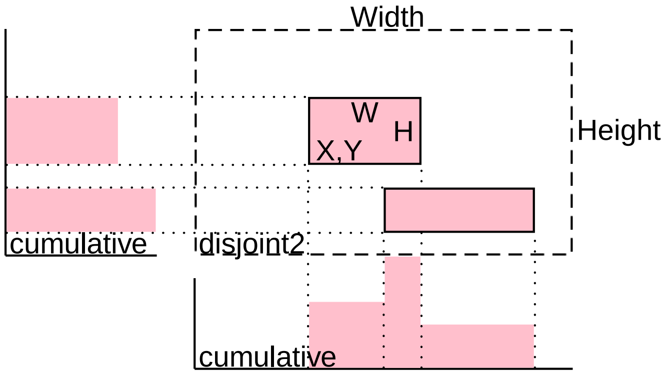
Problem Decomposition

- 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

Candidates


Height
80
75
70
65
60
55
50
45
40
35
Optimal
LB
UB
6200
6220
6240
6260
6280
6300
6320
6340
6360
6380
6400
Surface

Basic Model



Outline

Alternatives

- naive
- x then y
- disjunctive
- semantic disjunctive
- dual
- interval
- split
- xy interval

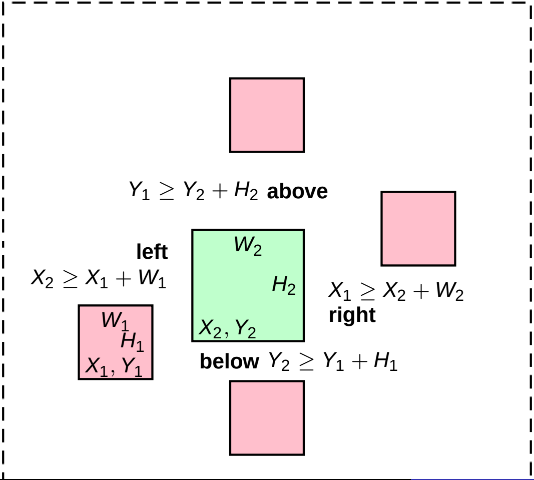
Naive

- Place items in order of decreasing size
- Fix X and Y value for each item
- Depth-first search to explore search space
- Problem: Large number of alternatives considered

X then Y

- Fix X value for all items, before assigning Y values
- Intuition: Once all X values are fixed, the Y values are very constrained
- Ideal: Search tree N, not $2 \times N$ levels deep
- Problem: May lead to deep thrashing if propagation too weak

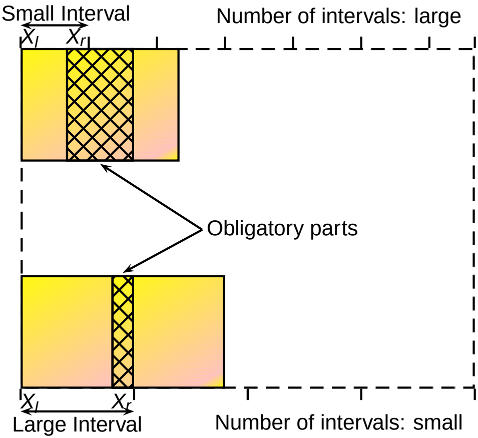
Disjunctive



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 Obligatory Parts



Variants

- (X) Interval
  - Split all X variables into intervals
  - Then fix X values
  - Then treat all Y variables the same way
- Split
  - Split X variables into intervals
  - Split Y variables into intervals
  - Then fix values
- XY Interval
  - For each item, split X and Y variables into intervals
  - Then fix values

Outline

Ignore 1×1 items

- They can be placed anywhere
- No need to include in propagation
- Interact with search routine, creating useless branches

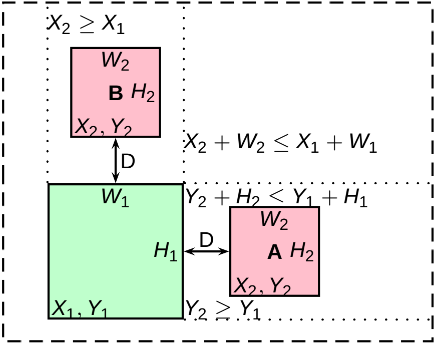
Adaptation of Korf's Dominance Criterion

- Certain placements are dominated by others
- No need to explore both
- Two variants
  - Ignore placement close to the border
  - Interaction of two items

Forbidden Gaps due to Dominance

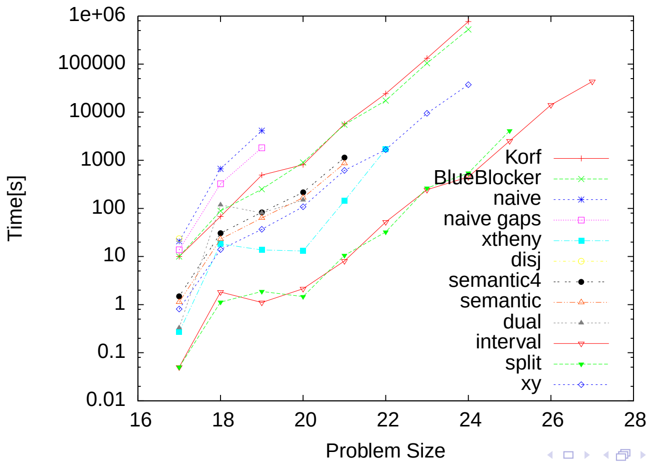
size	2	3	4	5-8	9-11	12-17	18-21	22-29	30-34	34-44	45
generic	1	2	2	3	4	5	6	7	8	9	10
specific	2	3									

Dominance Criterion



Outline

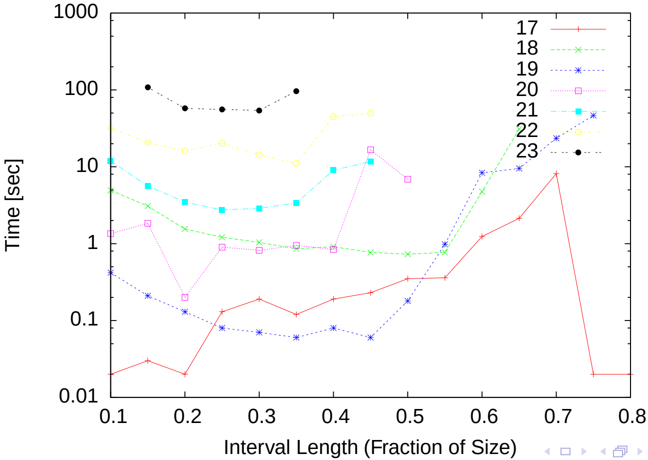
Strategies Comparison



Strategy Comparison

N	naive	xtheny	disj	semantic4	semantic	dual	interval 0.3	split 0.2	xy 0.75
15	2.92	0.09	12.12	0.55	0.45	2.63	-	0.05	-
16	10.44	0.11	98.25	1.31	1.03	0.89	-	0.05	-
17	20.75	0.27	23.57	1.48	1.13	0.33	0.05	0.05	0.81
18	667.33	18.37	-	30.53	23.05	118.58	1.83	1.13	13.94
19	4140.09	13.73	-	83.42	63.25	80.66	1.11	1.88	36.78
20	-	13.08	-	216.07	167.61	149.79	2.14	1.47	108.28
21	-	143.72	-	1138.98	865.13	-	8.09	10.59	619.45
22	-	1708.89	-	-	-	-	52.21	32.36	1668.59
23	-	-	-	-	-	-	245.07	265.54	9521.73
24	-	-	-	-	-	-	452.73	545.82	37506.20
25	-	-	-	-	-	-	2533.64	4127.41	-
26	-	-	-	-	-	-	14158.15	-	-
27	-	-	-	-	-	-	43529.87	-	-

Impact of Interval Length



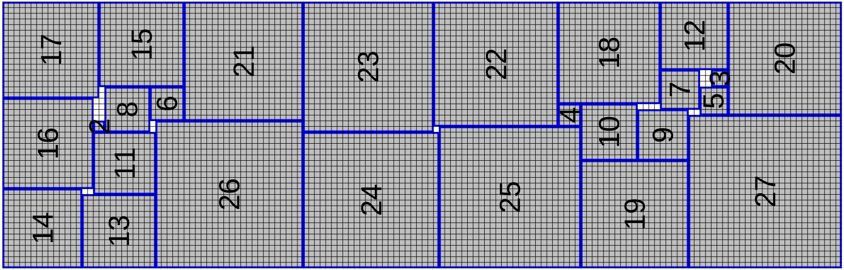
Method Comparison

N	pure	gap	domain	notone	all	best
18	100.00	99.37	78.96	12.93	9.77	9.78
19	100.00	101.61	87.14	48.55	38.26	37.31
20	100.00	105.26	92.24	18.93	16.20	15.39
21	100.00	100.94	81.90	63.57	50.82	49.58
22	100.00	100.24	90.56	23.66	19.46	19.00
23	100.00	99.81	78.92	30.33	23.18	22.80
24	100.00	101.77	77.69	36.43	29.16	28.58

Rectangle Placement Overview

N	Surface	K	Width	Height	Area	Loss	Time	Clautiaux	Korf	BlueBlocker
18	2109	14	31	69	2139	1.42	00:01	31:33	1:08	1:29
19	2470	12	47	53	2491	0.85	00:01	72:53:18	8:15	4:11
20	2870	14	34	85	2890	0.70	00:02	-	13:32	15:03
21	3311	19	38	88	3344	1.00	00:07	-	1:35:08	1:32:01
22	3795	15	39	98	3822	0.71	00:51	-	6:46:15	4:51:23
23	4324	19	64	68	4352	0.65	03:58	-	36:54:50	29:03:49
24	4900	18	56	88	4928	0.57	05:56	-	213:33:00	146:38:48
25	5525	17	43	129	5547	0.40	40:38	-	see paper	-
26	6201	21	70	89	6230	0.47	03:41:43	-	-	-
27	6930	21	47	148	6956	0.38	11:30:02	-	-	-

Optimal Solution (N=27)



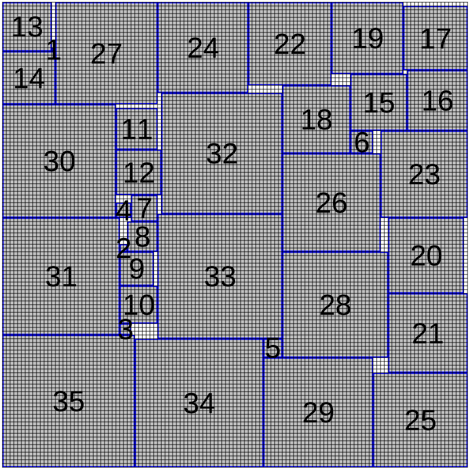
Packing In Square

- Pack all squares 1×1 to $N \times N$ into smallest square
- Fewer candidates to check, start with lower bound $\lceil \sqrt{\sum_{i=1}^{N} i^2} \rceil$
- Often, lower bound is reached $\implies$ optimal
- More slack in optimal solution than for rectangle packing
- Improved symmetry breaking possible

Optimal Solution for Packing in Square

Problem Size	26	27	29	30	31	35
Optimal Solution	80	84	93	98	103	123
$T_{\texttt{opt}}$	12:26	00:04	11:06	2:07	00:18	1:10:07
$T_{\texttt{proof}}$	1:25:22	-	-	-	-	-

Optimal Solution (N=35)

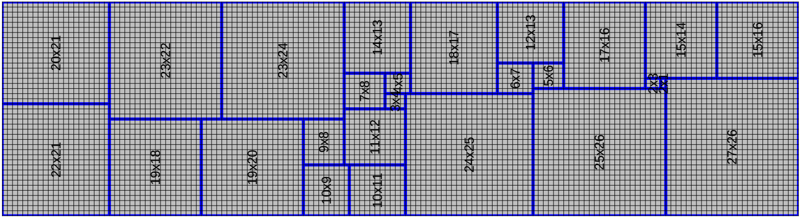


Outline

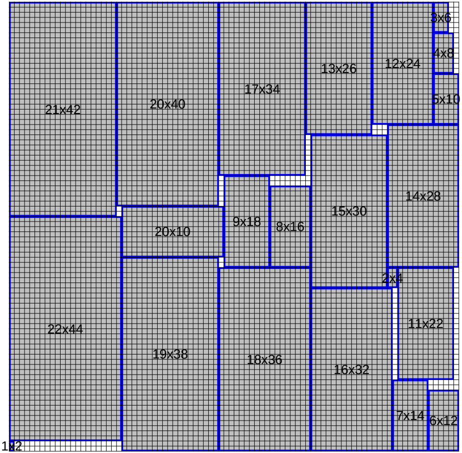
Packing Rectangles, not Squares

- Rectangles may or may not be rotated
- Introduces new degree of freedom
- Does not change constraint model
- Rotation handled as part of search
- Surprising: impact quite small

Almost Square, Optimal Solution (N=26)



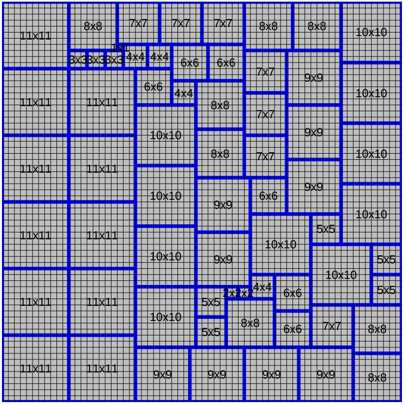
Dominoes, Optimal Solution (N=22)



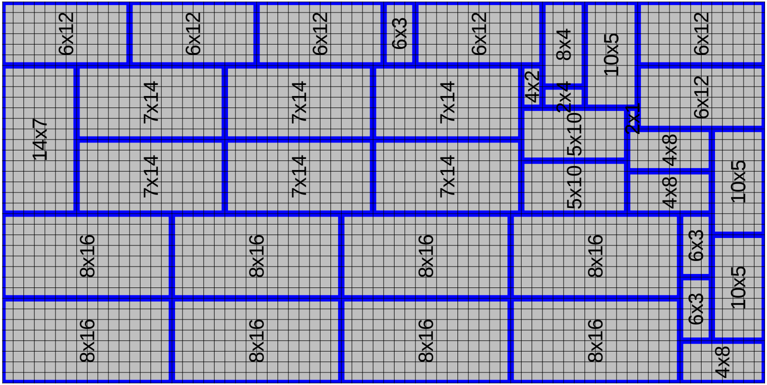
Multiple items of same size

- Introduces additional symmetries
- Removing symmetry by lexicographic constraints
- Constrain all items of same size
- Works for square or rectangle items

Partridge, Optimal Solution (N=11)



Partridge x2, Optimal Solution (N=8)



Summary

- Rectangle packing problems can be solved by off-the-shelf constraint technology.
- Choice of search strategy of prime importance.
- Out-performs previous best solution by factor 1000.

- Future Research
  - Which impact has slack in the problem?
  - Faster heuristics for feasible problems?