Anytime AND/OR Depth-first Search for Combinatorial Optimization

Lars Otten and Rina Dechter

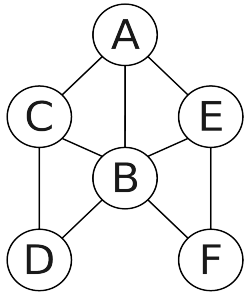
Dept. of Computer Science
University of California, Irvine

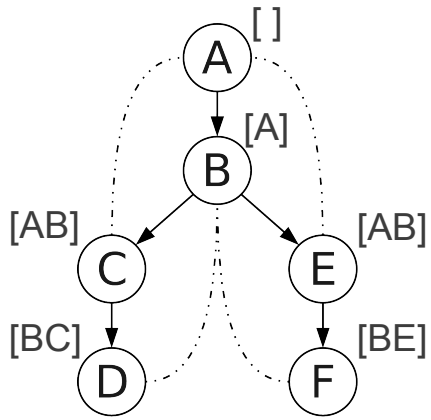
Outline

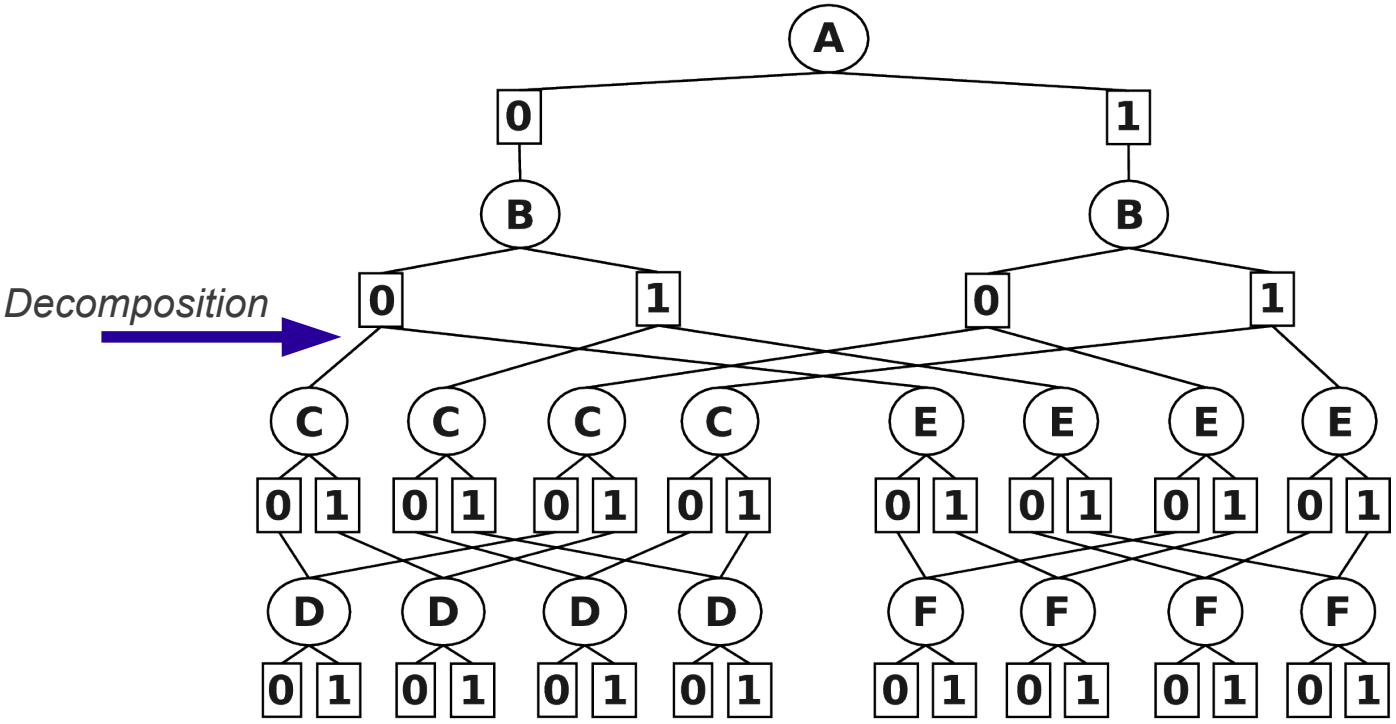
- AND/OR Search Spaces.
  - AND/OR Branch and Bound (AOBB).
- Conflict: Decomposition vs. Anytime.
  - Empirical evidence & analysis.
- Breadth-Rotating AOBB.
  - Example & analysis.
- Experimental Results.
  - Substantial improvements.

AND/OR Branch and Bound

- Guided by pseudo tree:
  - Subproblem decomposition.

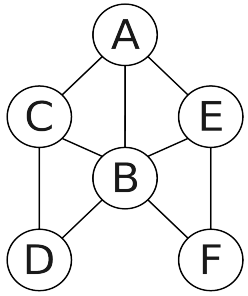


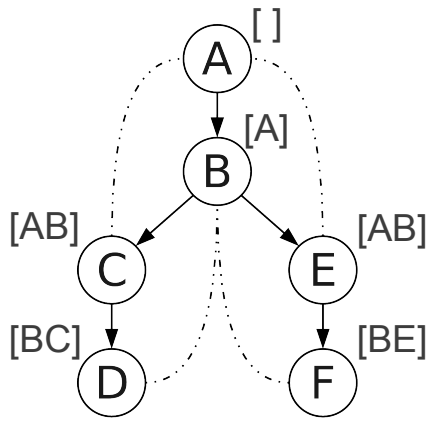


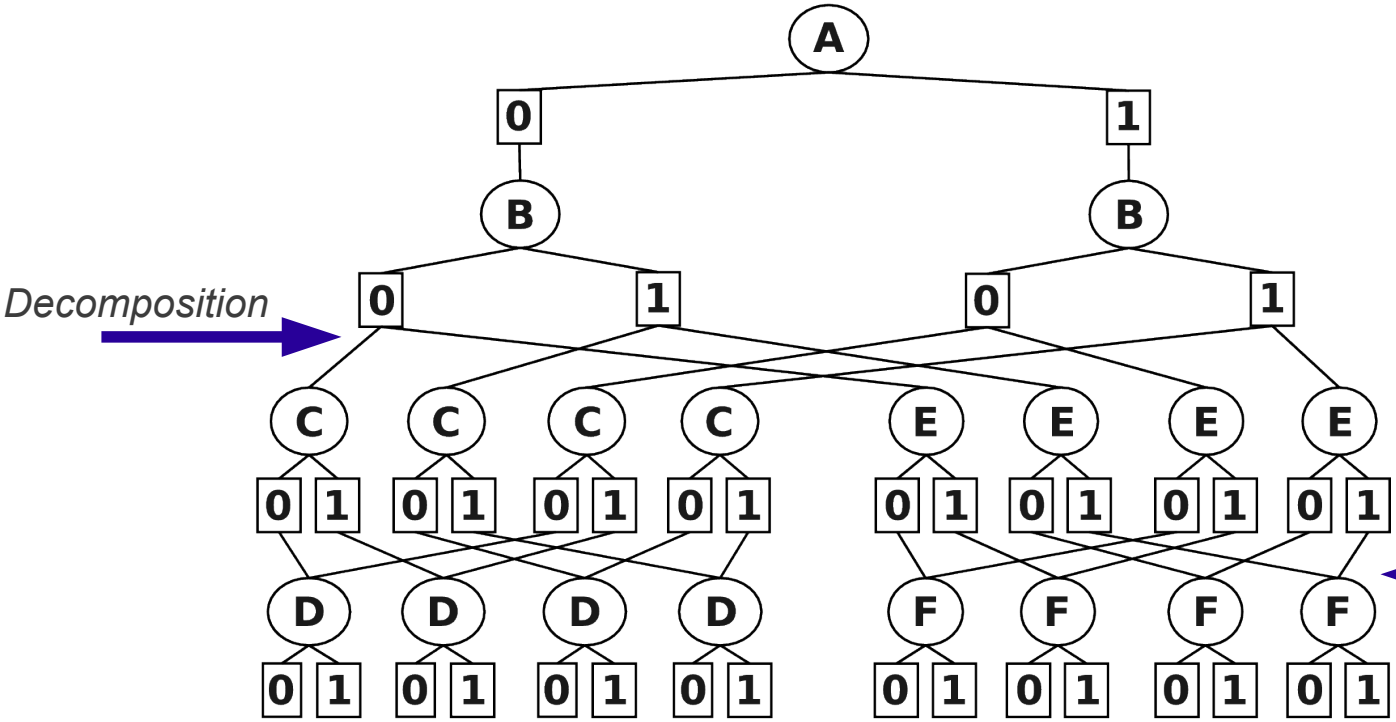


AND/OR Branch and Bound

- Guided by pseudo tree:
  - Subproblem decomposition.
  - Merge unifiable subproblems.





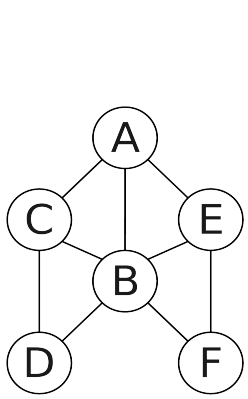


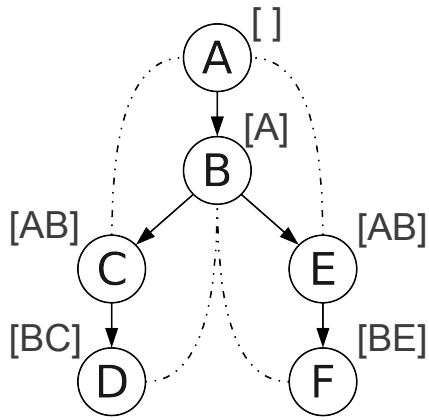
Cache table for F (independent of A)

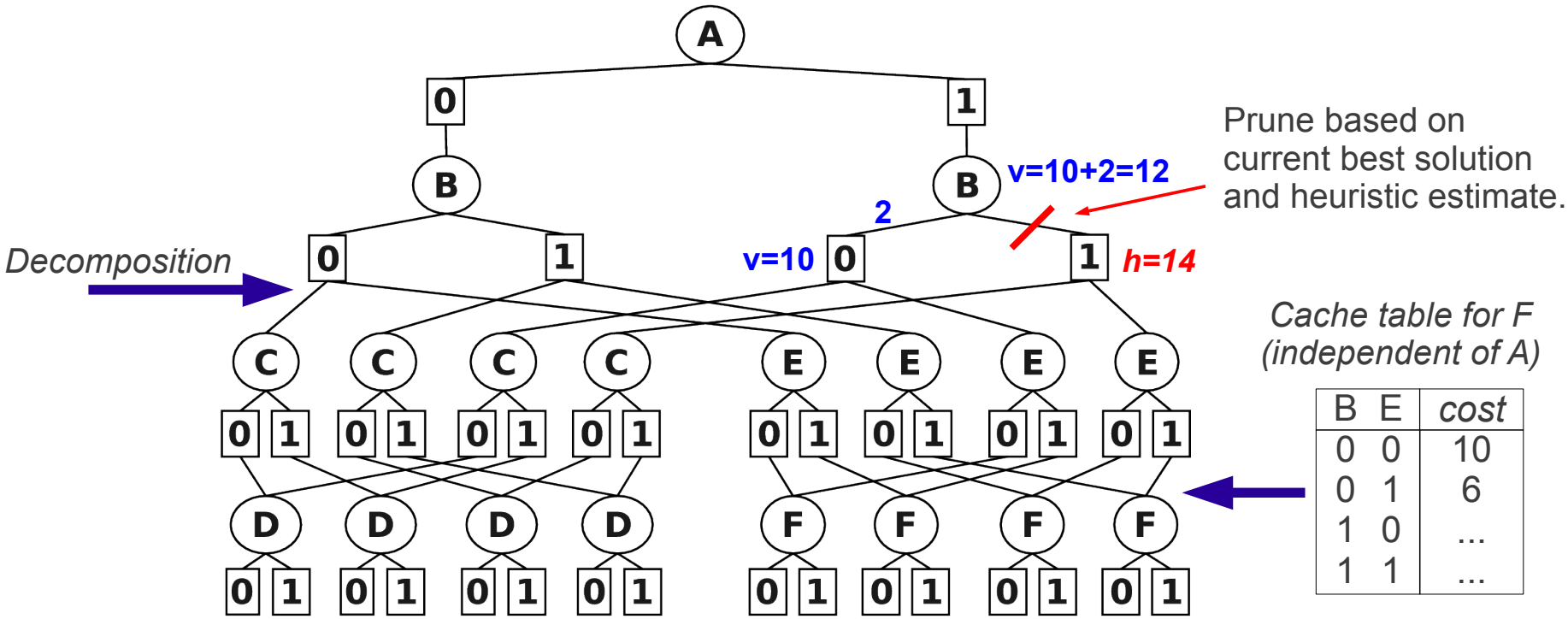
B	E	*cost*
0	0	10
0	1	6
1	0	...
1	1	...

AND/OR Branch and Bound

- Guided by pseudo tree:
  - Subproblem decomposition.
  - Merge unifiable subproblems.
- Mini-bucket heuristic.

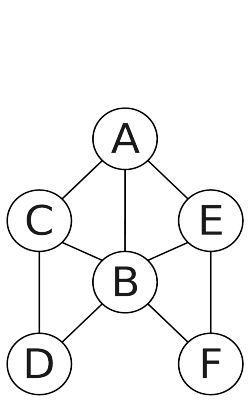


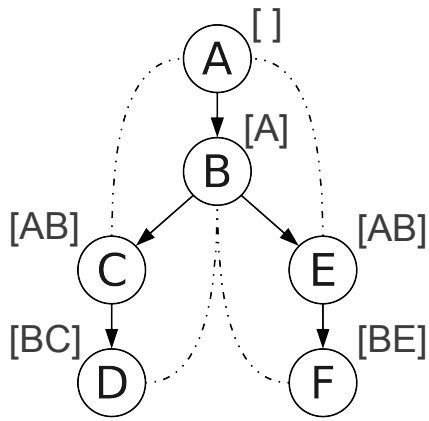


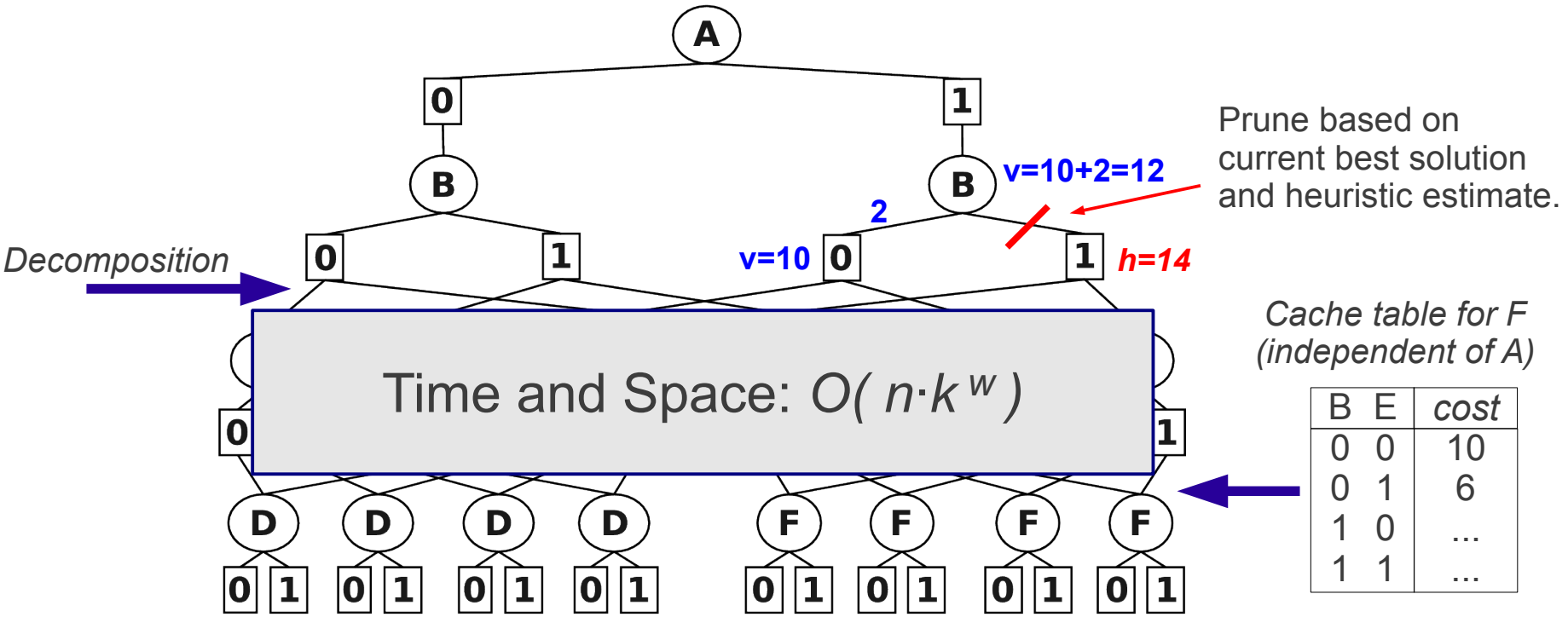


AND/OR Branch and Bound

- Guided by pseudo tree:
  - Subproblem decomposition.
  - Merge unifiable subproblems.
- Mini-bucket heuristic.





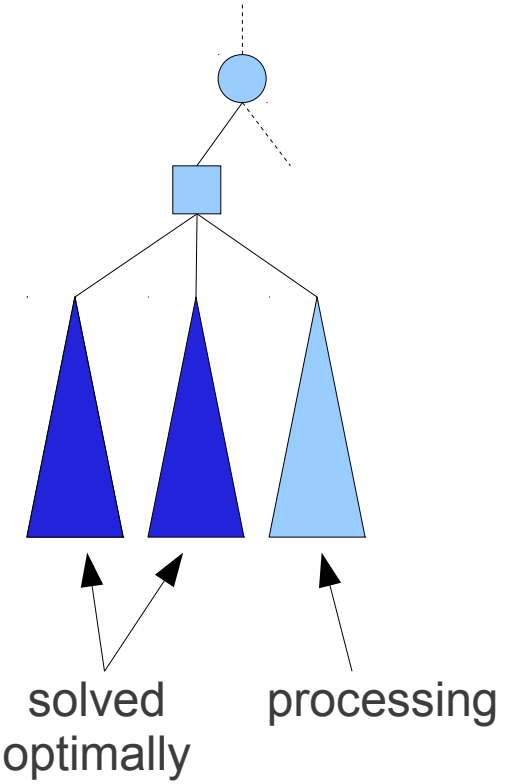


Anytime Behavior of AOBB

- Finding/proving optimal solution hard!
- Often easier:
  - Find any feasible solution quickly.
  - Improve with time, until optimum is found.
- Branch and Bound is anytime.
  - Depth-first to any solution and improve.
- Usage as approximation scheme:
  - AOBB competitive in UAI'10 evaluation.
    - But some cases: no solution within time bound.

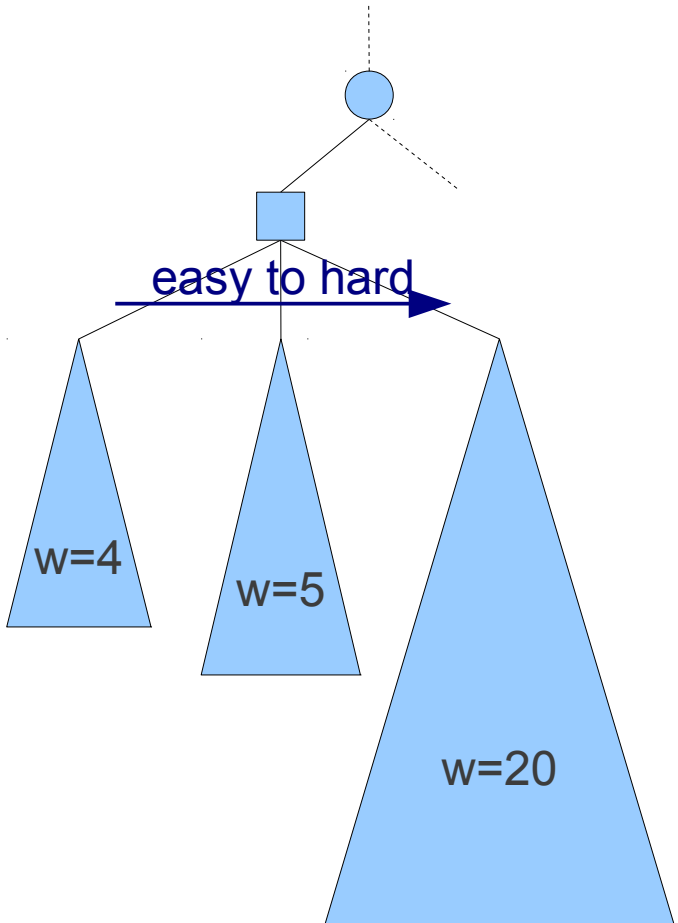
Anytime vs. Decomposition

- Depth-first traversal of AND/OR space:
  - Subproblems successively solved to optimality.
- Breaks anytime behavior.
- First overall solution:
  - Is delayed until last subproblem starts processing.
  - Contains optimal solutions to all but last subproblem.



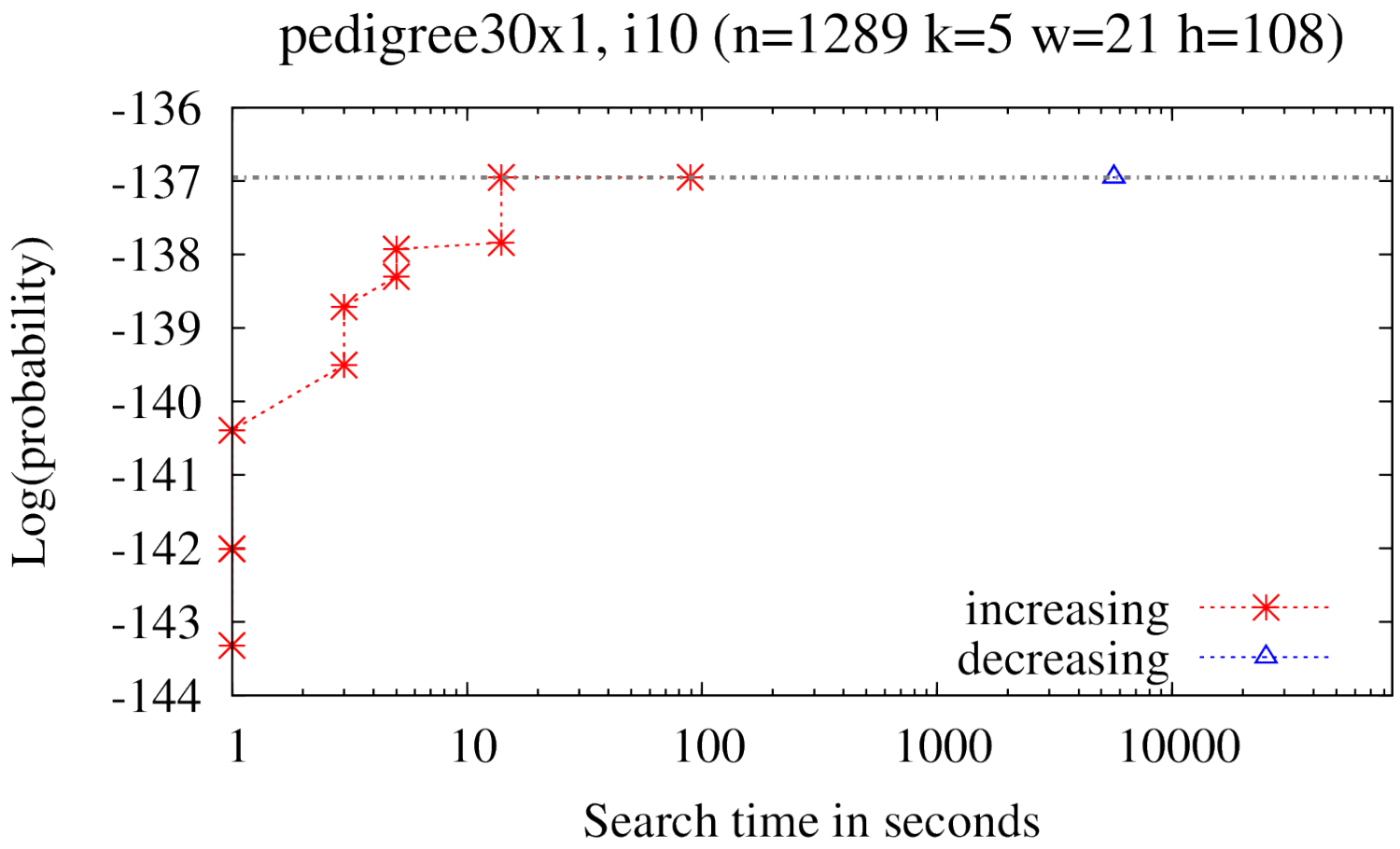
Remedy I: Subproblem Order

- One complex and several easy subproblems:
  - Solve easy ones optimally (fast).
  - Combine with anytime solutions from complex subproblem.
- Suggests ordering subproblems by increasing hardness:
  - Heuristic: subproblem induced width w .
    - AOBB has time O($n \cdot k^w$) .
  - However, fails for multiple complex subproblems.



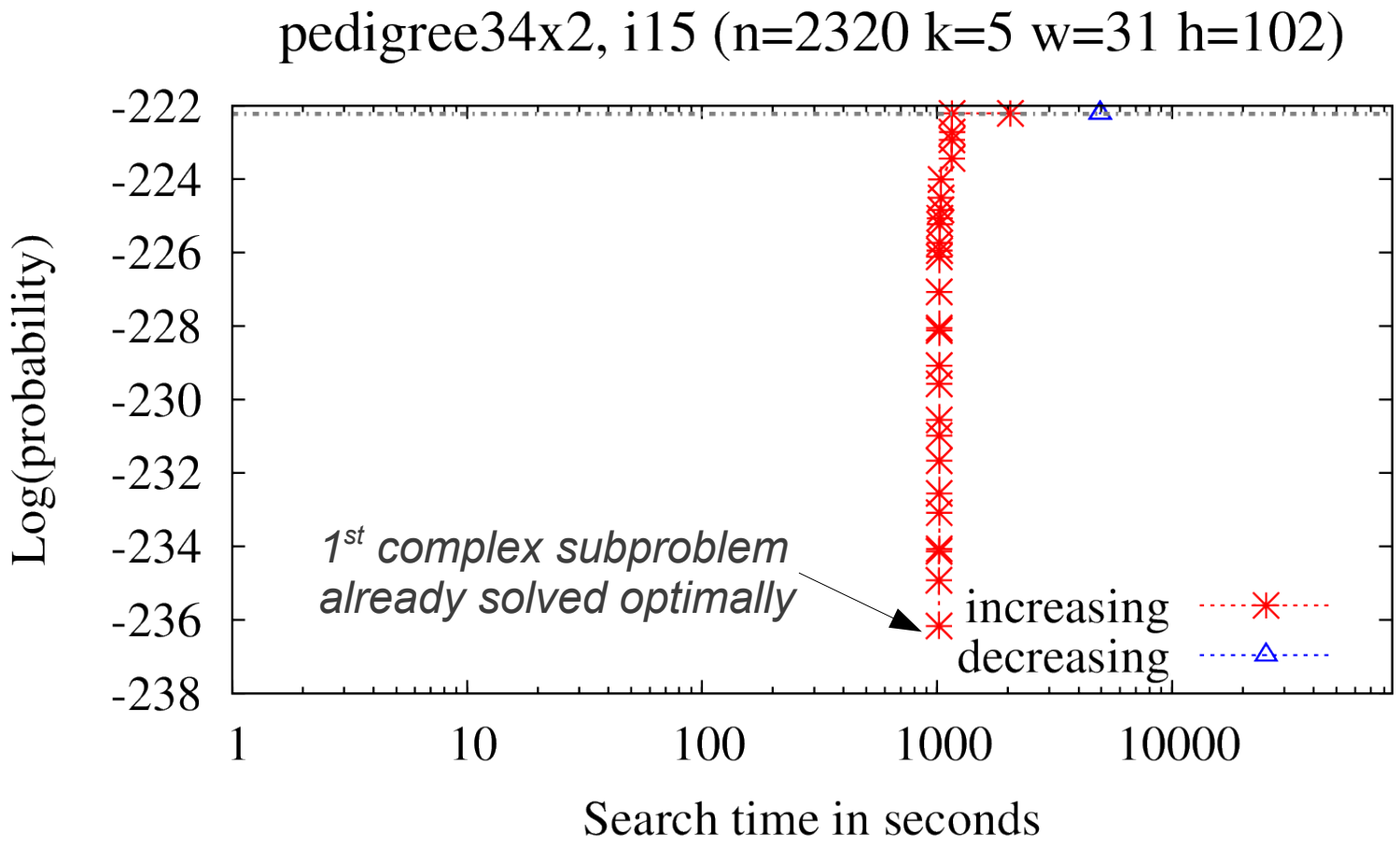
Remedy I: Subproblem Order

- **1** large and several smaller components.
  - Increasing order yields good performance.



Remedy I: Subproblem Order

- **2** large and several smaller components.
  - Even increasing order fails.

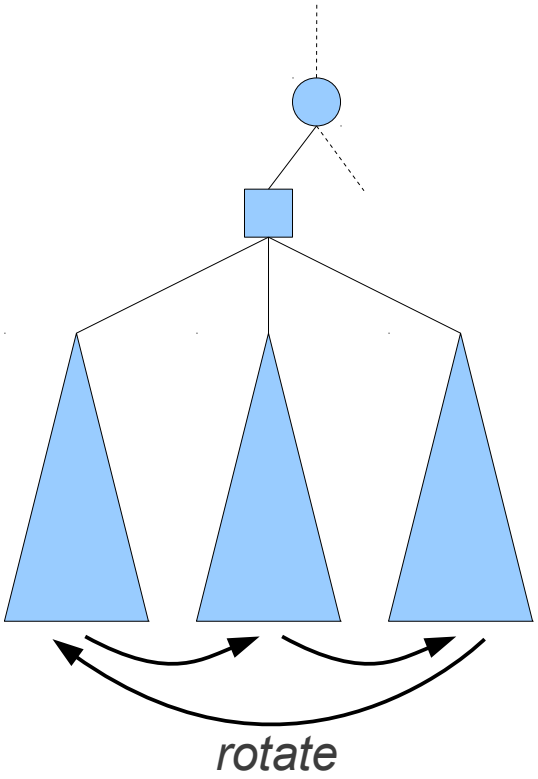


Remedy II: Subproblem Dive

- Upon decomposition:
  - Greedily find solution for each subproblem.
    - Use mini-bucket heuristic for guidance.
  - Then solve each subproblem optimally, depth-first as before.
- Relies very much on heuristic:
  - Might fail due to dead end.
- Mixed performance in experiments.

Breadth-Rotating AOBB

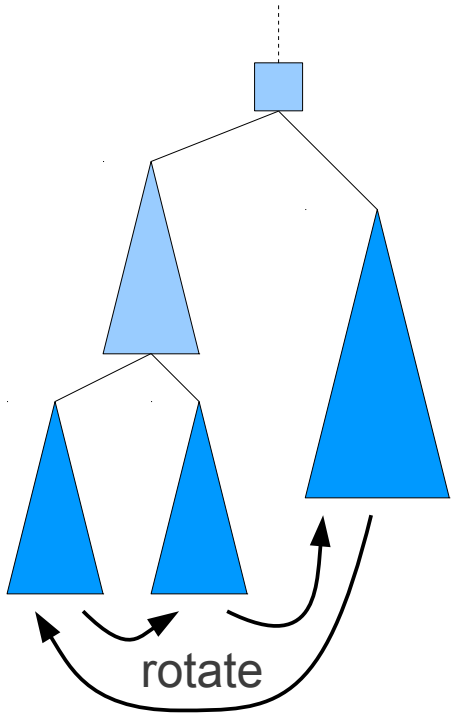
- Construct all branches of solution tree “simultaneously”:
  - Take turns in processing subproblems.
    - Limit number of operations per visit.
  - Solve each depth-first as before.



Breadth-Rotating AOBB

- High-level pseudo code:

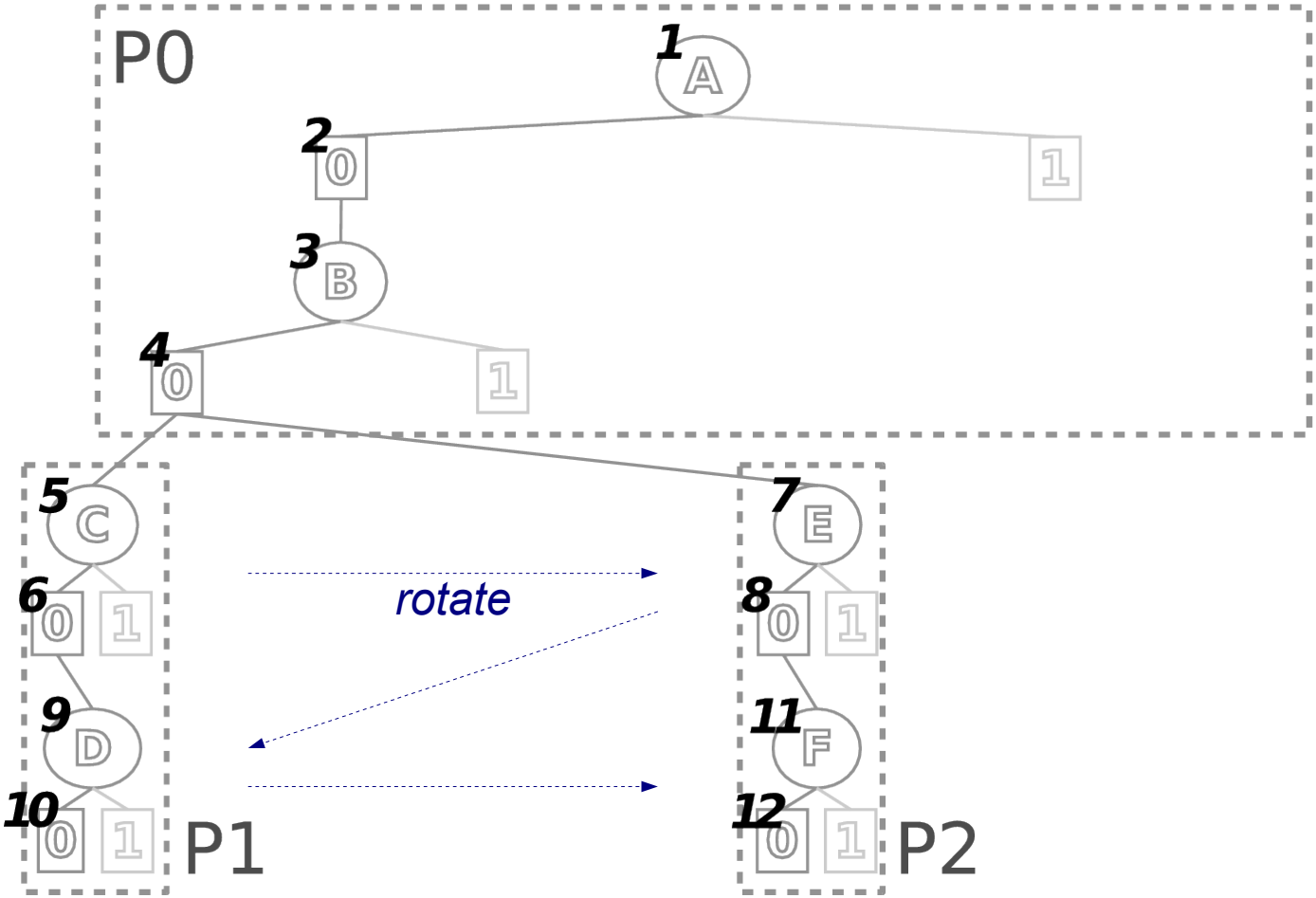
1. Move breadth-first to next open subproblem *P* .
2. Process *P* depth-first, until either:
   - *P* is solved optimally,
   - *P* decomposes into child subproblems,
   - a predefined threshold of operations is reached.



- Rotation skips subproblems with current child subproblems.

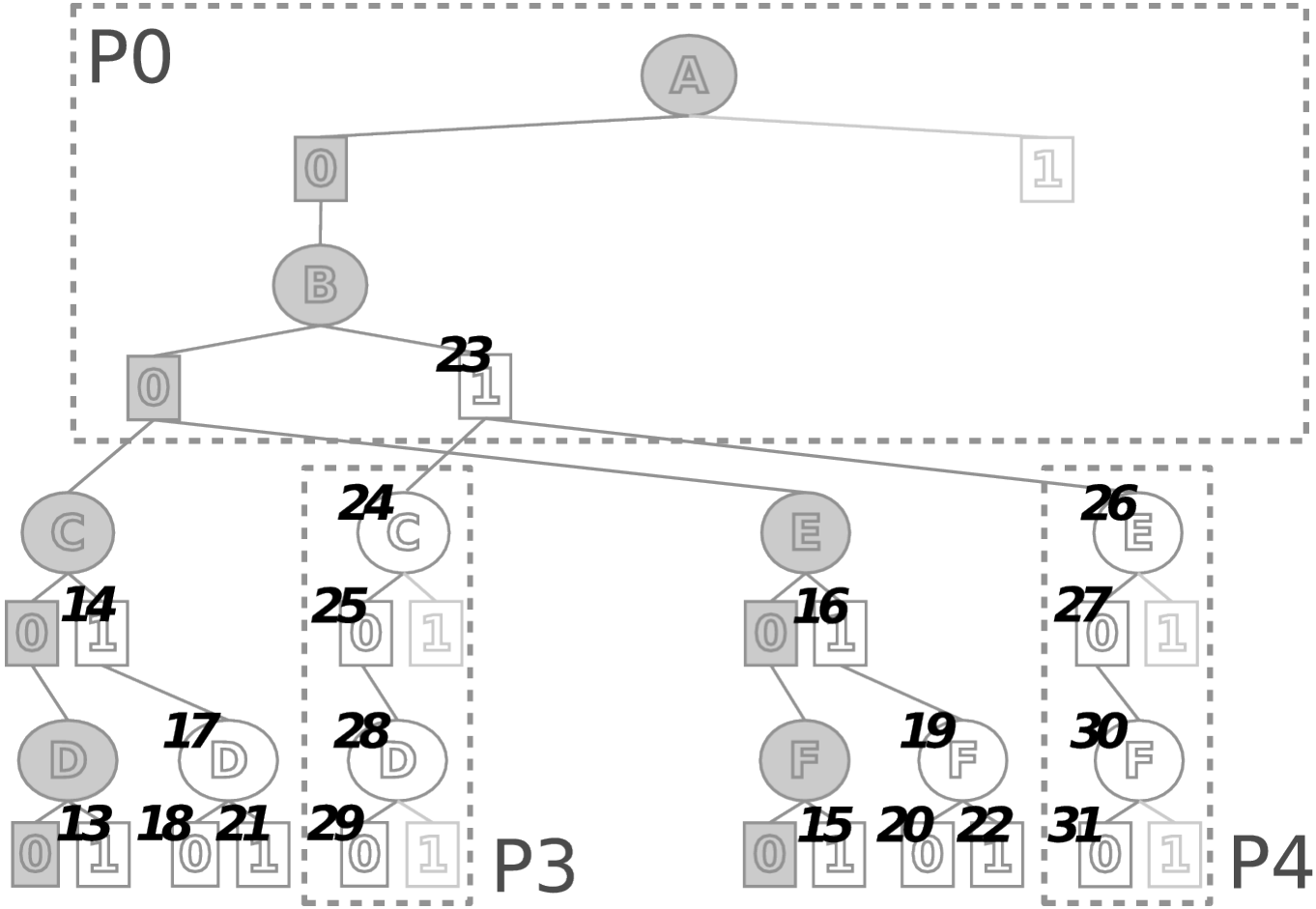
Breadth-Rotating AOBB

- Example problem with threshold $z = 2$.



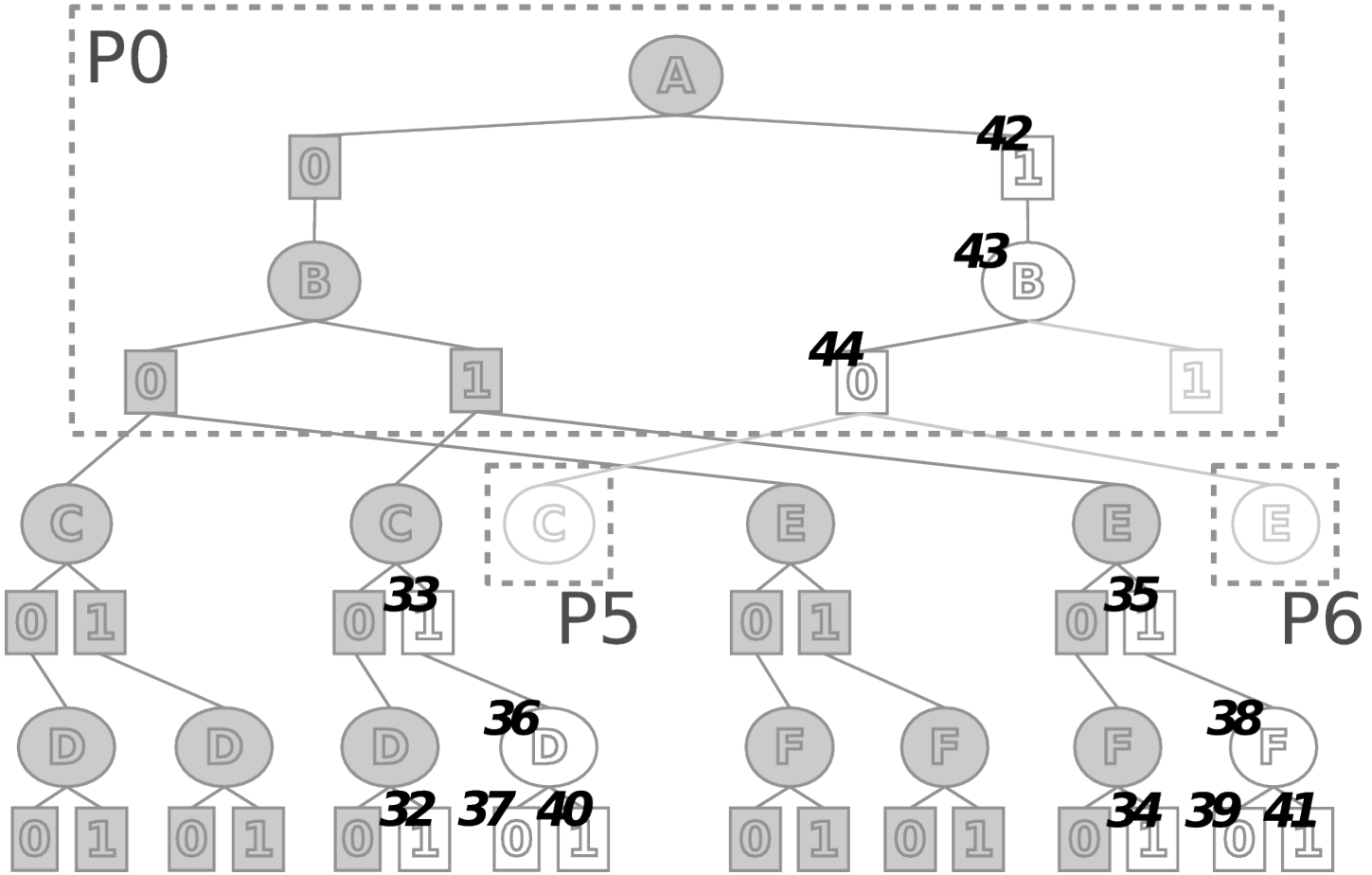
Breadth-Rotating AOBB

- Example problem with threshold $z = 2$.



Breadth-Rotating AOBB

- Example problem with threshold $z = 2$.



BRAOBB Analysis

- *Theorem:* BRAOBB maintains favorable asymptotic complexity of depth-first search.
  - Time: graph search $O(n \cdot k^w)$, tree search $O(n \cdot k^h)$.
  - Space: graph search $O(n \cdot k^w)$, tree search $O(n)$.

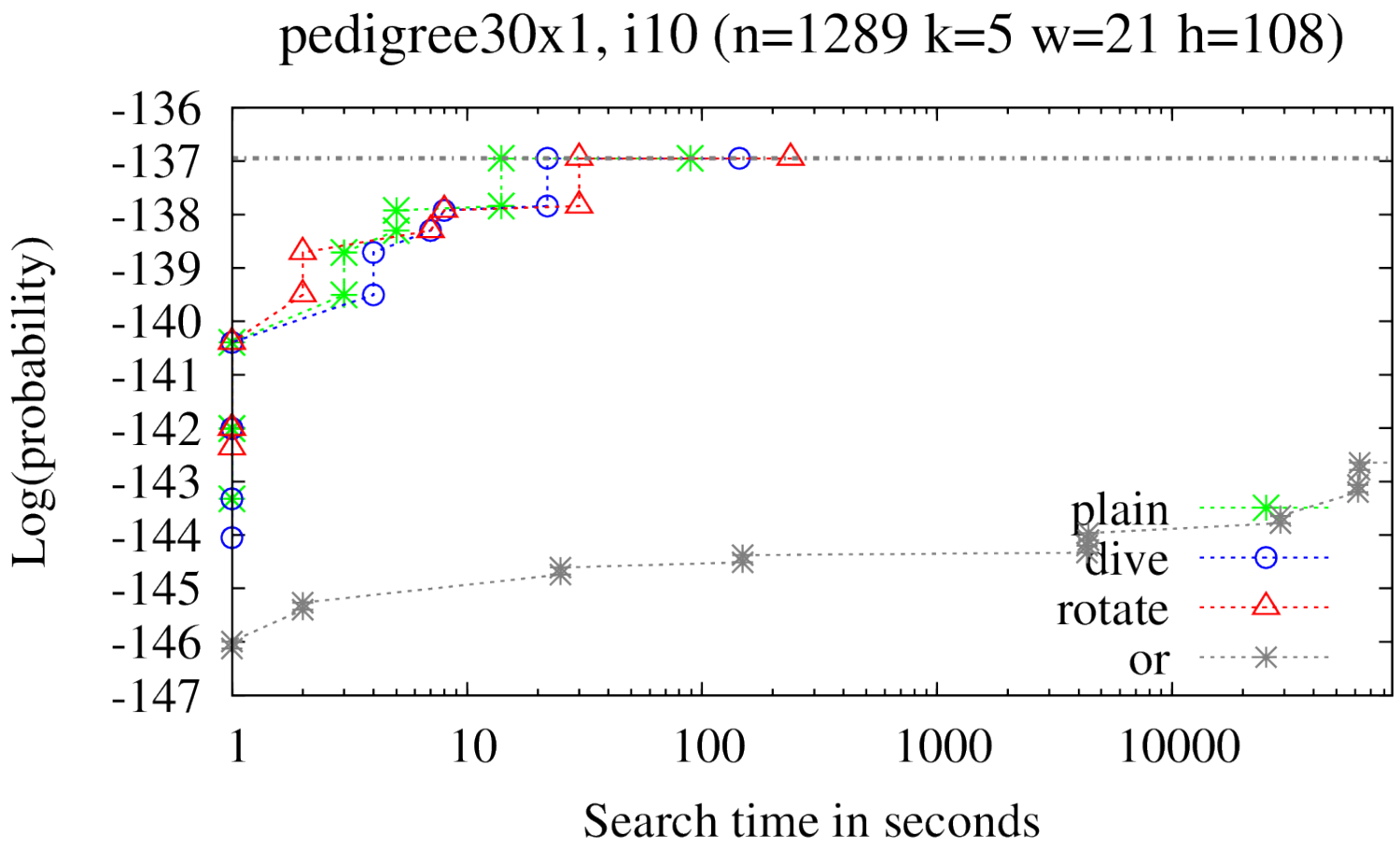
- Comparison with AOBB:
  - Anytime performance does not depend on subproblem order.
  - Overall performance (optimality proof) can increase or decrease.
    - Pruning impacted by node exploration.

Experimental Evaluation

- Run and record anytime profiles (24h timeout):
  - Plain AOBB (increasing subproblem order).
  - AOBB with greedy dive.
  - Breadth-rotating AOBB.
  - OR graph search (no decomposition).
- Enforce decomposable problems:
  - Create network copies and connect at root.
  - 57 pedigree, 150 grid, 24 mastermind.
    - Combined over 60.000 CPU hours.
- UAI'10: Two very hard instances.

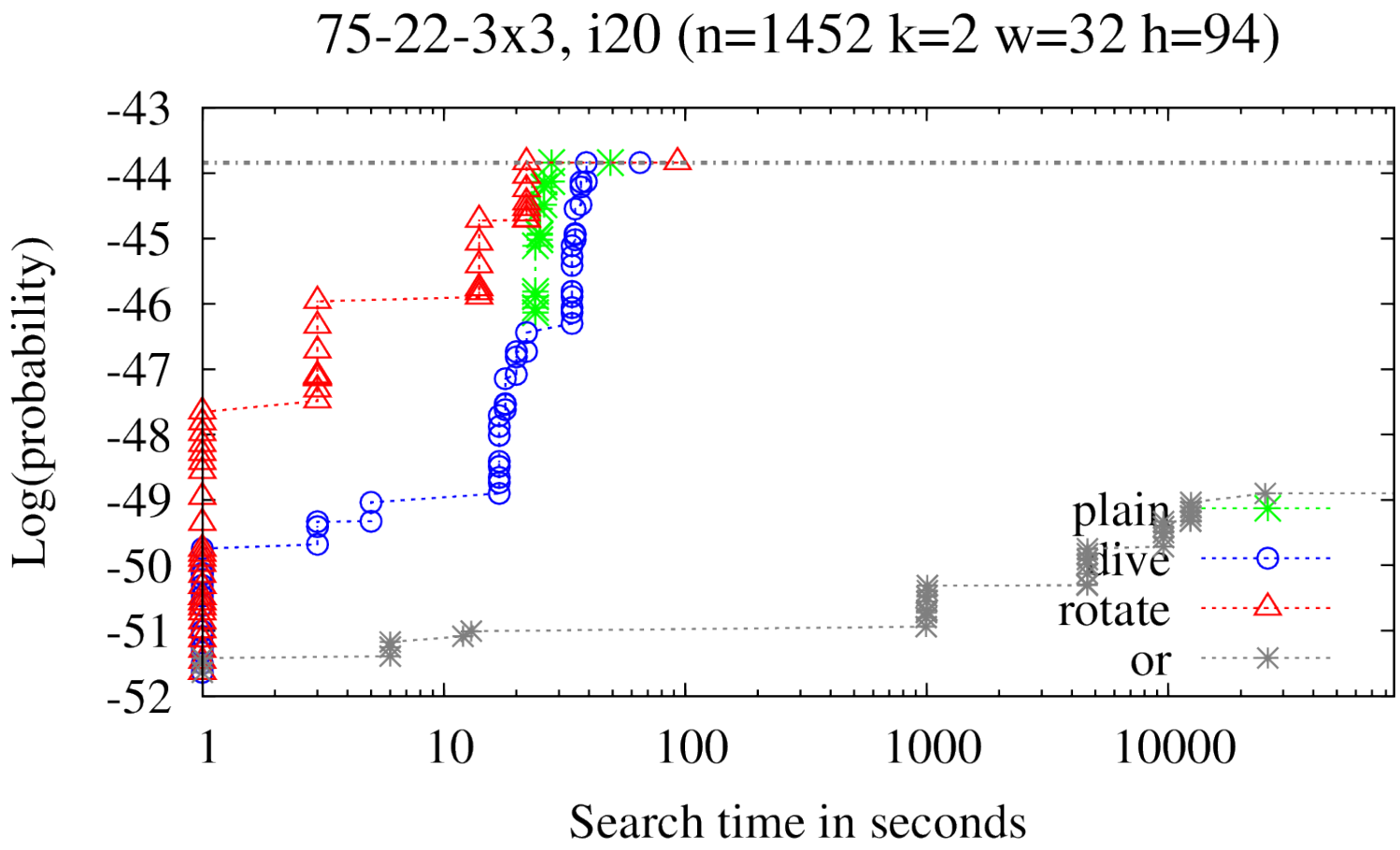
Experimental Evaluation

- "Sanity check": **1** large subproblem
  - AND/OR schemes similarly good, *OR* search slow.



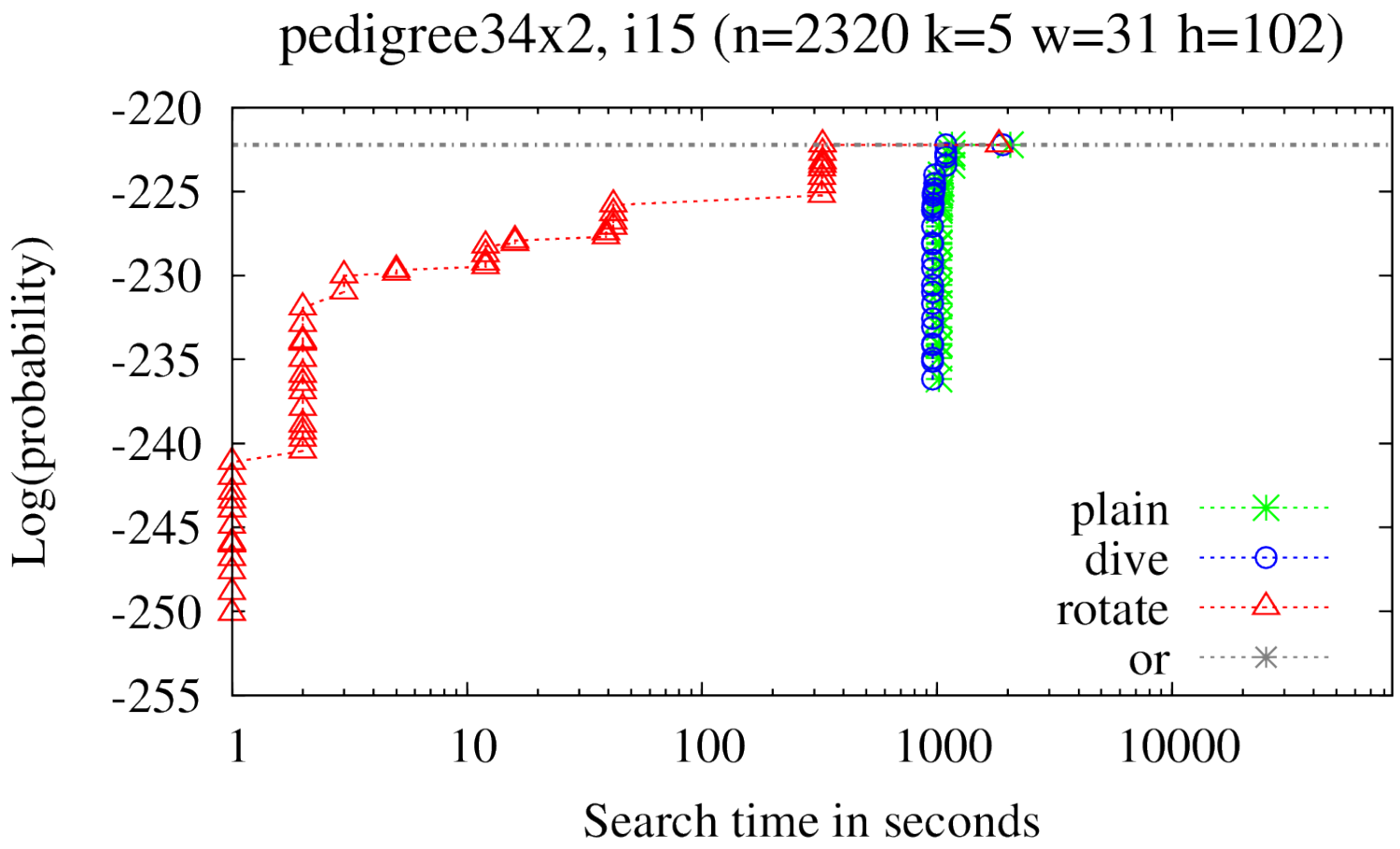
Experimental Evaluation

- Grid network with three large subproblems.
  - *rotate* outperforms *dive*, *plain* very late.



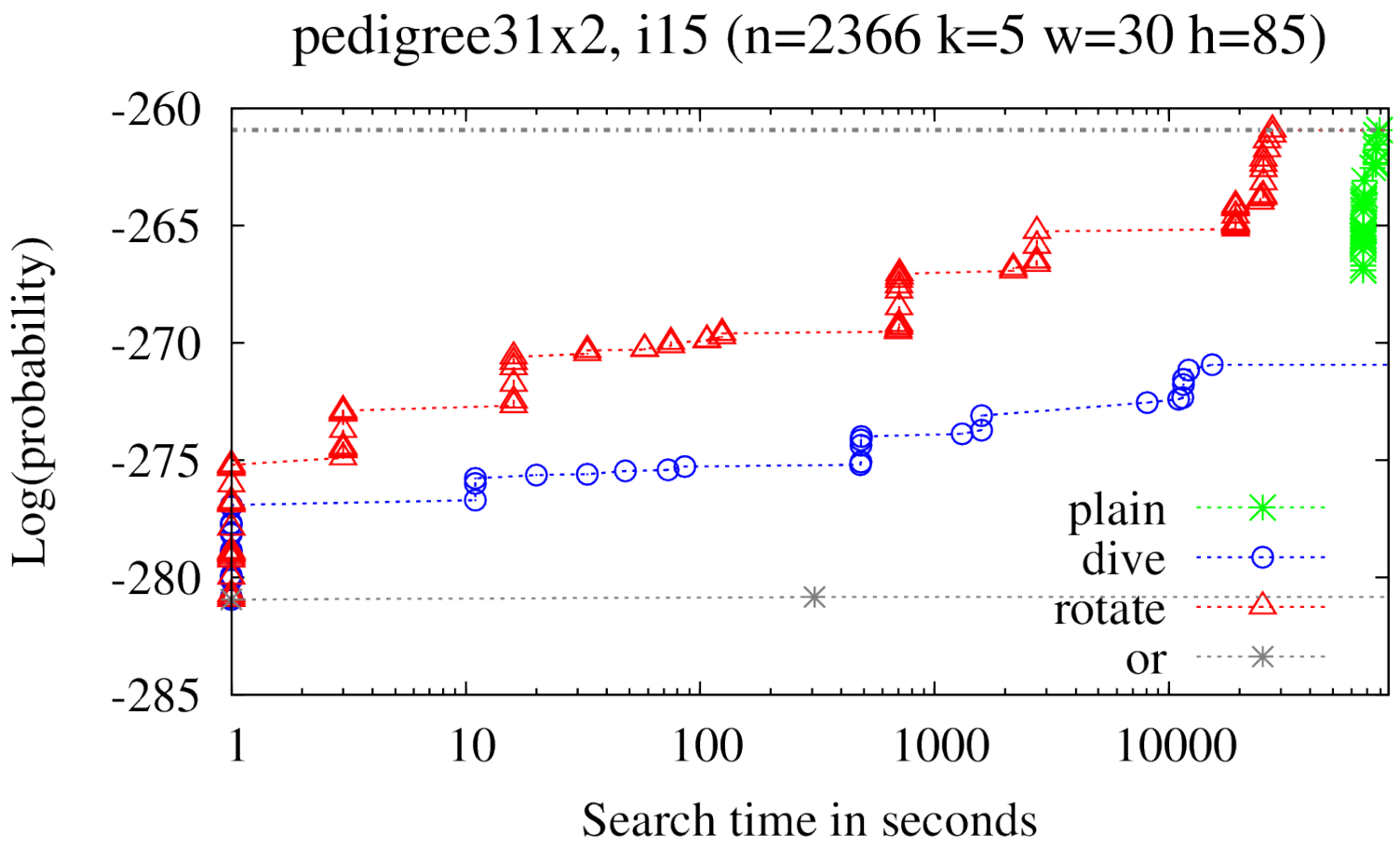
Experimental Evaluation

- Pedigree with two large subproblems.
  - *Dive* fails due to dead end, behaves like *plain*.



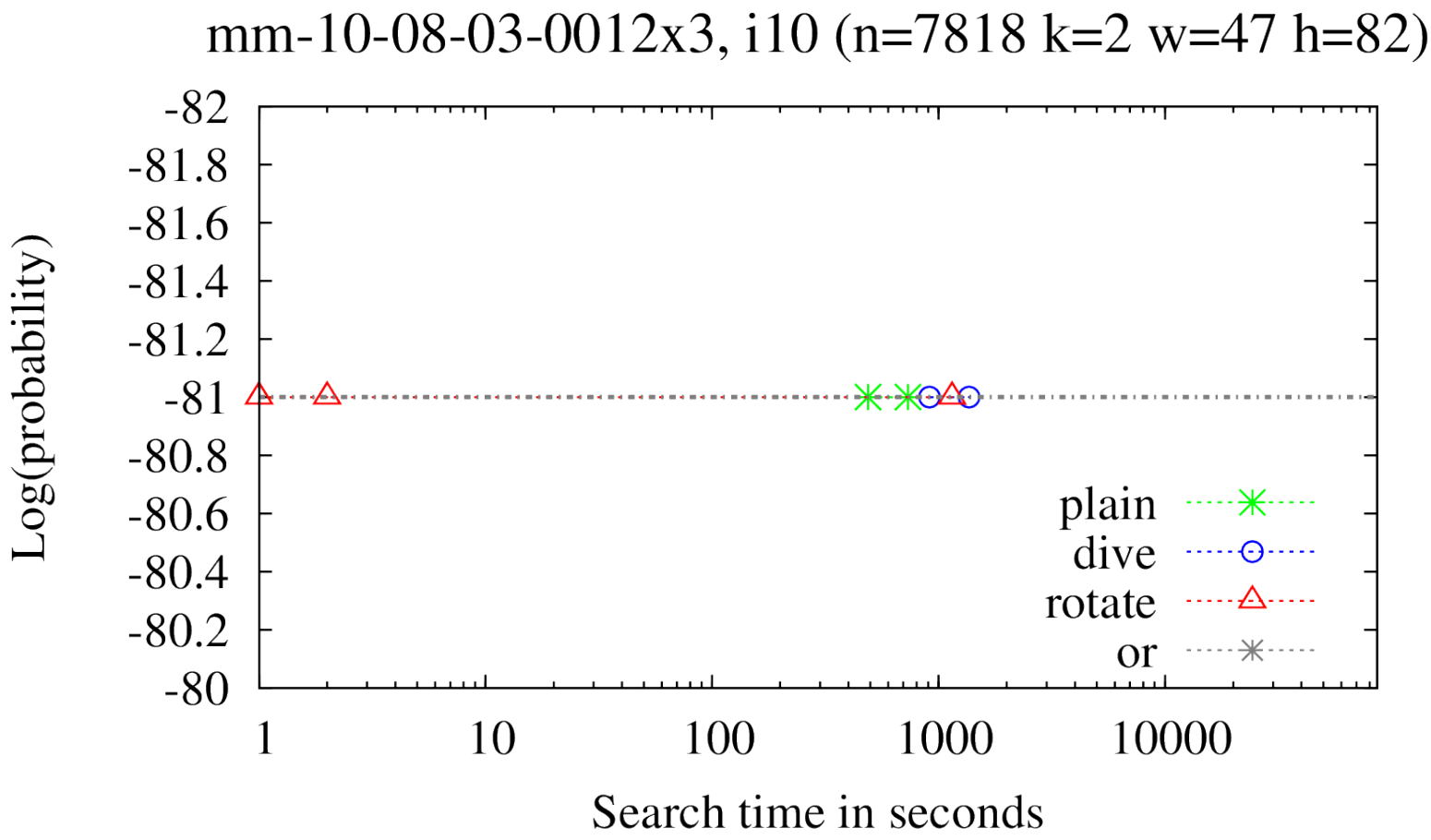
Experimental Evaluation

- Pedigree with two large subproblems.
  - *rotate* outperforms *dive*, *plain* very late.



Experimental Evaluation

- Mastermind with three complex subproblems.
  - *rotate* finds solution much sooner.



Experimental Evaluation

- Summary table, entries give #instances:
  - Any solution / optimal solution / optimality proven

	Time bound						
	1 sec	5 sec	10 sec	1 min	5 min	1 hour	24 hours
Pedigree networks (171 total)							
plain	52 / 19 / **6**	70 / 36 / **17**	75 / 42 / **24**	87 / 56 / **48**	101 / **76** / **68**	111 / 90 / **86**	129 / 117 / **108**
dive	76 / 16 / 5	86 / 29 / 13	94 / 38 / 20	105 / 53 / **48**	116 / 69 / 64	127 / 89 / **86**	135 / 114 / 105
rotate	**153** / **26** / 2	**157** / **40** / 15	**160** / **47** / **24**	**162** / **59** / **48**	**164** / 74 / 60	**165** / **98** / 84	**167** / **127** / 102
or	73 / 6 / 1	76 / 7 / 5	77 / 10 / 5	79 / 10 / 9	82 / 12 / 11	87 / 16 / 15	90 / 22 / 21
Grid networks (150 total)							
plain	38 / 10 / 0	48 / 19 / 0	58 / 32 / **4**	84 / 62 / **52**	101 / 82 / **76**	128 / 120 / **113**	**149** / 148 / **147**
dive	47 / 6 / 0	52 / 12 / 0	55 / 24 / 1	82 / 54 / 37	97 / 78 / 71	121 / 111 / 104	147 / 147 / 146
rotate	**122** / **16** / 0	**128** / **27** / 0	**129** / **35** / 1	**136** / **69** / 38	**143** / **86** / 73	**146** / **126** / 110	**149** / **149** / **147**
or	45 / 0 / 0	45 / 1 / 0	46 / 1 / 0	53 / 2 / 0	57 / 4 / 2	64 / 10 / 9	74 / 21 / 21
Mastermind networks (24 total)							
plain	8 / 8 / **1**	8 / 8 / **3**	8 / 8 / **3**	10 / 10 / 4	13 / 13 / **7**	17 / 17 / 12	**24** / **24** / **24**
dive	8 / 8 / **1**	8 / 8 / **3**	8 / 8 / **3**	11 / 11 / **5**	12 / 12 / 6	21 / 21 / **19**	**24** / **24** / **24**
rotate	**18** / **18** / **1**	**18** / **18** / **3**	**18** / **18** / **3**	**18** / **18** / 3	**21** / **21** / 4	**24** / **24** / **19**	**24** / **24** / **24**
or	0 / 0 / 0	0 / 0 / 0	0 / 0 / 0	0 / 0 / 0	0 / 0 / 0	0 / 0 / 0	0 / 0 / 0

Experimental Evaluation

- Summary table, entries give #instances:
  - Any solution / optimal solution / optimality proven

	Time bound						
	1 sec	5 sec	10 sec	1 min	5 min	1 hour	24 hours
Pedigree networks (171 total)							
plain	52 / 19 / **6**	70 / 36 / **17**	75 / 42 / **24**	87 / 56 / **48**	101 / **76** / **68**	111 / 90 / **86**	129 / 117 / **108**
dive	76 / 16 / 5	86 / 29 / 13	94 / 38 / 20	105 / 53 / **48**	116 / 69 / 64	127 / 89 / **86**	135 / 114 / 105
rotate	**153** / **26** / 2	**157** / **40** / 15	**160** / **47** / **24**	**162** / **59** / **48**	**164** / 74 / 60	**165** / **98** / 84	**167** / **127** / 102
or	73 / 6 / 1	76 / 7 / 5	77 / 10 / 5	79 / 10 / 9	82 / 12 / 11	87 / 16 / 15	90 / 22 / 21
Grid networks (150 total)							
plain	38 / 10 / 0	48 / 19 / 0	58 / 32 / **4**	84 / 62 / **52**	101 / 82 / **76**	128 / 120 / **113**	**149** / 148 / **147**
dive	47 / 6 / 0	52 / 12 / 0	55 / 24 / 1	82 / 54 / 37	97 / 78 / 71	121 / 111 / 104	147 / 147 / 146
rotate	**122** / **16** / 0	**128** / **27** / 0	**129** / **35** / 1	**136** / **69** / 38	**143** / **86** / 73	**146** / **126** / 110	**149** / **149** / **147**
or	45 / 0 / 0	45 / 1 / 0	46 / 1 / 0	53 / 2 / 0	57 / 4 / 2	64 / 10 / 9	74 / 21 / 21
Mastermind networks (24 total)							
plain	8 / 8 / **1**	8 / 8 / **3**	8 / 8 / **3**	10 / 10 / 4	13 / 13 / **7**	17 / 17 / 12	**24** / **24** / **24**
dive	8 / 8 / **1**	8 / 8 / **3**	8 / 8 / **3**	11 / 11 / **5**	12 / 12 / 6	21 / 21 / **19**	**24** / **24** / **24**
rotate	**18** / **18** / **1**	**18** / **18** / **3**	**18** / **18** / **3**	**18** / **18** / 3	**21** / **21** / 4	**24** / **24** / **19**	**24** / **24** / **24**
or	0 / 0 / 0	0 / 0 / 0	0 / 0 / 0	0 / 0 / 0	0 / 0 / 0	0 / 0 / 0	0 / 0 / 0

First solutions quickly.

Experimental Evaluation

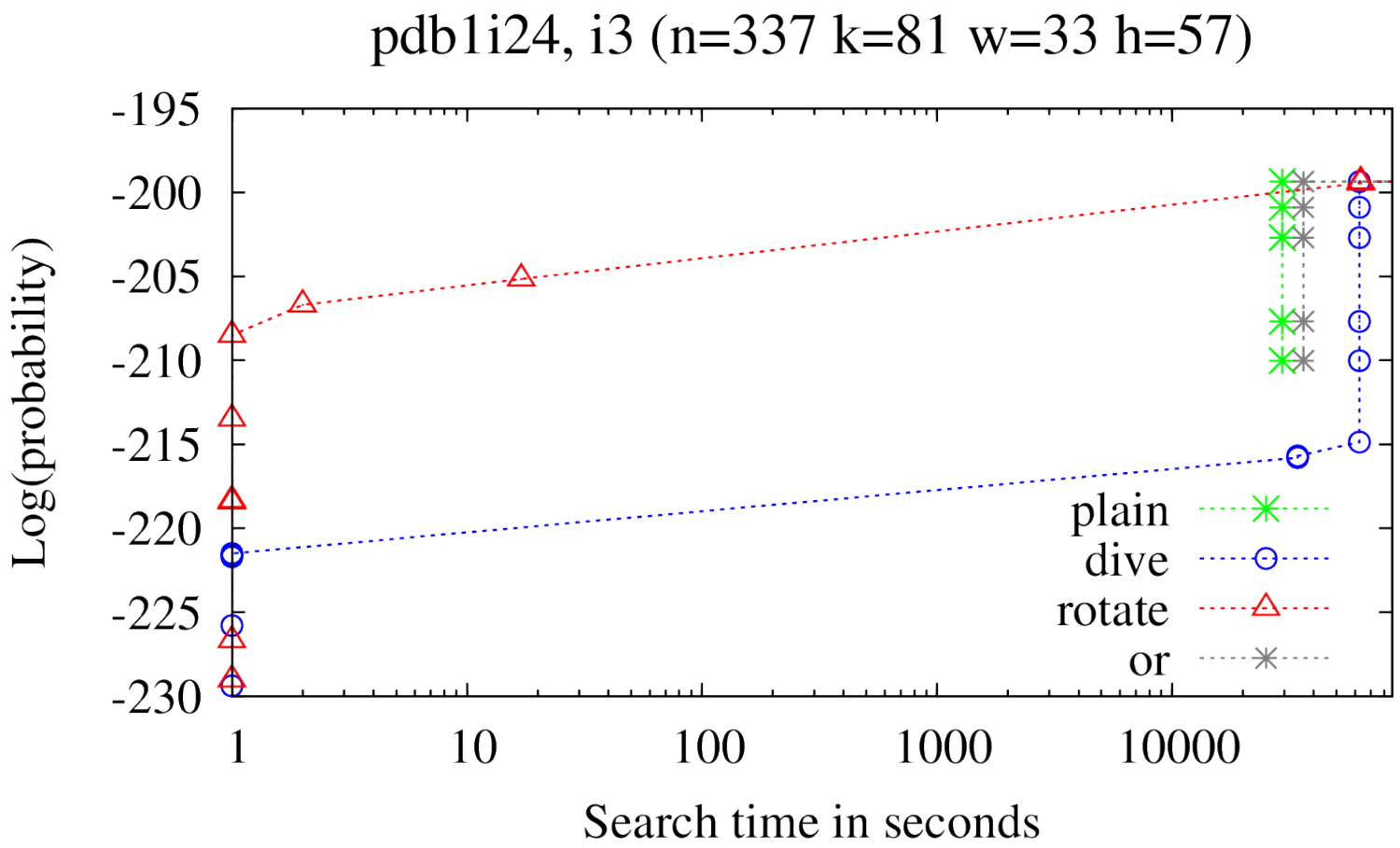
- Summary table, entries give #instances:
  - Any solution / optimal solution / optimality proven

	Time bound						
	1 sec	5 sec	10 sec	1 min	5 min	1 hour	24 hours
Pedigree networks (171 total)							
plain	52 / 19 / **6**	70 / 36 / **17**	75 / 42 / **24**	87 / 56 / **48**	101 / **76** / **68**	111 / 90 / **86**	129 / 117 / **108**
dive	76 / 16 / 5	86 / 29 / 13	94 / 38 / 20	105 / 53 / **48**	116 / 69 / 64	127 / 89 / **86**	135 / 114 / 105
rotate	**153** / **26** / 2	**157** / **40** / 15	**160** / **47** / **24**	**162** / **59** / **48**	**164** / 74 / 60	**165** / **98** / 84	**167** / **127** / 102
or	73 / 6 / 1	76 / 7 / 5	77 / 10 / 5			87 / 16 / 15	90 / 22 / 21
Grid ne							
plain	38 / 10 / 0	48 / 19 / 0	58 / 32 / **4**	84 / 62 / **52**	101 / 82 / **76**	128 / 120 / **113**	**149** / 148 / **147**
dive	47 / 6 / 0	52 / 12 / 0	55 / 24 / 1	82 / 54 / 37	97 / 78 / 71	121 / 111 / 104	147 / 147 / 146
rotate	**122** / **16** / 0	**128** / **27** / 0	**129** / **35** / 1	**136** / **69** / 38	**143** / **86** / 73	**146** / **126** / 110	**149** / **149** / **147**
or	45 / 0 / 0	45 / 1 / 0	46 / 1 / 0	53 / 2 / 0	57 / 4 / 2	64 / 10 / 9	74 / 21 / 21
Mastermind networks (24 total)							
plain	8 / 8 / **1**	8 / 8 / **3**	8 / 8 / **3**	10 / 10 / 4	13 / 13 / **7**	17 / 17 / 12	**24** / **24** / **24**
dive	8 / 8 / **1**	8 / 8 / **3**	8 / 8 / **3**	11 / 11 / **5**	12 / 12 / 6	21 / 21 / **19**	**24** / **24** / **24**
rotate	**18** / **18** / **1**	**18** / **18** / **3**	**18** / **18** / **3**	**18** / **18** / 3	**21** / **21** / 4	**24** / **24** / **19**	**24** / **24** / **24**
or	0 / 0 / 0	0 / 0 / 0	0 / 0 / 0	0 / 0 / 0	0 / 0 / 0	0 / 0 / 0	0 / 0 / 0

Faster to optimality,
a bit slower to prove.

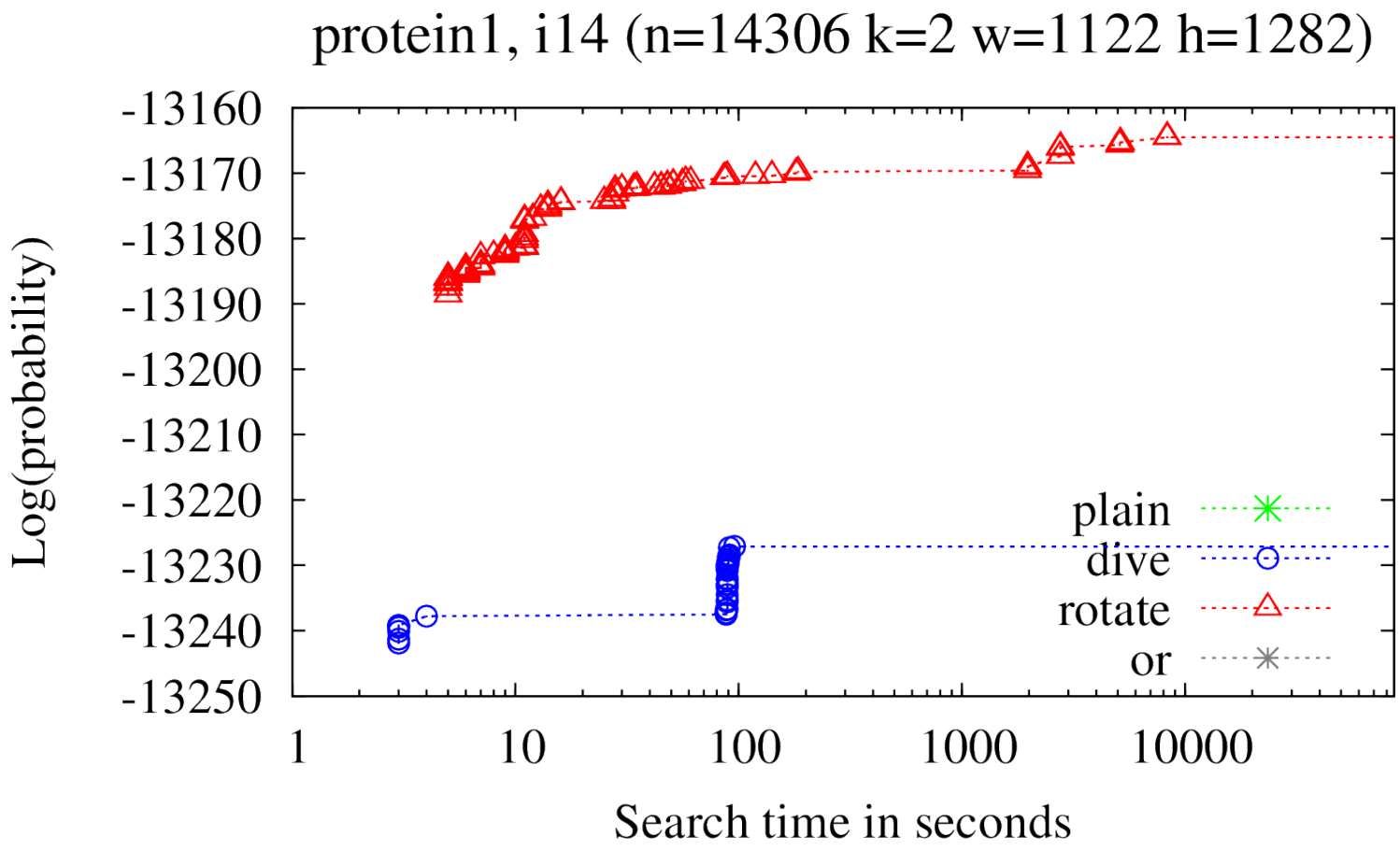
Experimental Evaluation

- UAI'10: protein sidechain prediction.
  - Very hard due to large domains.



Experimental Evaluation

- UAI'10: protein folding.
  - Very complex, induced width >1000.



Experiments Summary

- Plain AOBB:
  - Fails for more than one complex subproblem.
- AOBB with greedy dive:
  - Quick initial solution possible, if heuristic allows.
  - Slow to improve afterwards.
- Breadth-rotating AOBB:
  - Consistently good anytime performance.
    - Immediate initial solution on 293/345 instances.
    - Subsequent rapid improvements.
  - Useful as approximation scheme.

Conclusion

- Anytime behavior of depth-first BaB is compromised over AND/OR search spaces.
  - Suitable subproblem ordering and other remedies only partially viable.
- Introduced Breadth-rotating AOBB:
  - Rotate over subproblems, breadth-first.
  - Each explored depth-first.
  - Maintains asymptotic complexity.
- Greatly improved anytime performance.

Thank you!

Questions?

Impact of rotation threshold *z*

- BRAOBB: little change with different *z* .
  - Other rotation criteria trigger first.