Anytime Best+Depth-First Search for Bounding Marginal MAP

Radu Marinescu
IBM Research - Ireland

Junkyu Lee, Alex Ihler and Rina Dechter
University of California, Irvine

AAAI 2017 Technical Session: RU: Reasoning under Uncertainty
Feb. 8th. 2017 10:00 am – 11:00 am Oral Presentation Paper 2066

Motivation and Contribution

- Marginal MAP Inference
  - Probabilistic inference query
    - Optimal partial configuration after marginalizing hidden/latent variables in a probability distribution
  - Complexity: NP^{pp} complete
  - Often it is the right task on various applications
    - Probabilistic conformant planning [Lee, Marinescu, Dechter, 2015]
    - Natural language processing task [Bird, Klein, Loper, 2009]
    - Image completion task [Xue, Li, Ermon, Gomes, Selman, 2016]

- Contributions
  - Anytime hybrid (best+depth-first) search for MMAP
  - Improvement of anytime performance for finding upper and lower bounds
    - Upper-bound: estimate of optimal solution from a partial solution
    - Lower-bound: sub-optimal solution

Outline

- Background
  - Graphical model
  - AND/OR search space & WMB heuristic
  - Previous MMAP search algorithms
- Best+Depth-First search for MMAP
  - LAOBF (Best-First AND/OR Search with Depth-First Lookaheads)
  - AAOBF (Alternating Best-First and Depth-First AND/OR search)
  - LnDFS (Learning Depth-First AND/OR search)
- Experiments
- Conclusion

Background – graphical model

- Graphical model $\mathcal{M} = \langle \mathbf{X}, \mathbf{D}, \mathbf{F} \rangle$
  - variables $\mathbf{X} = \{X_1, \ldots, X_n\}$
  - domains $\mathbf{D} = \{D_1, \ldots, D_n\}$
  - functions $\mathbf{F} = \{\psi_1, \ldots, \psi_r\}$

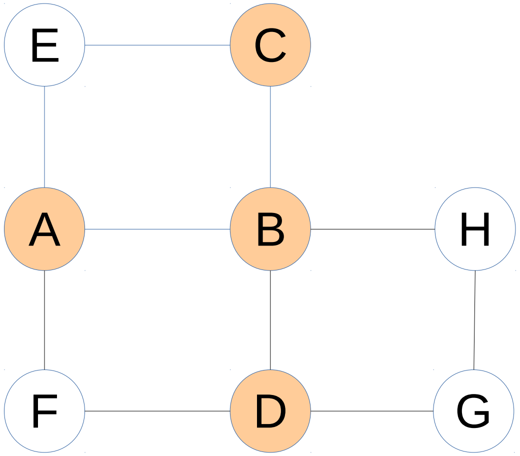
$$P(\mathbf{x}) = \frac{1}{Z} \prod_{\alpha \in F} \psi_\alpha(\mathbf{x}_\alpha)$$

- Marginal Map task

$$\mathrm{x}^*_M = \mathrm{argmax}_{\mathbf{x}_M} \sum\nolimits_{\mathbf{x}_s} \prod\nolimits_\alpha \psi_\alpha(\mathbf{x}_\alpha)$$

  - $\mathbf{x} = \mathbf{x}_M \cup \mathbf{x}_S$
  - Max and sum not commute

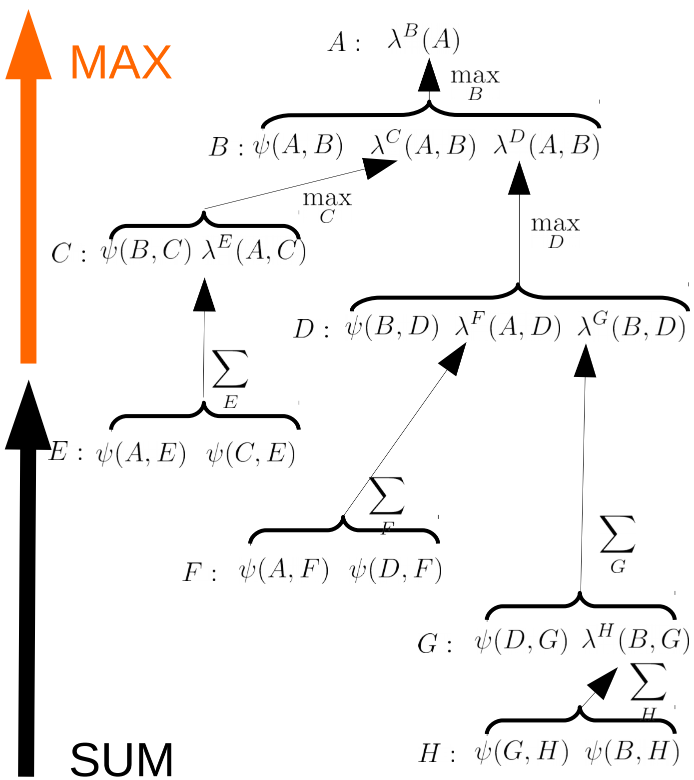
- Primal graph
  - nodes are variables
  - two nodes are connected if they appear in the same function



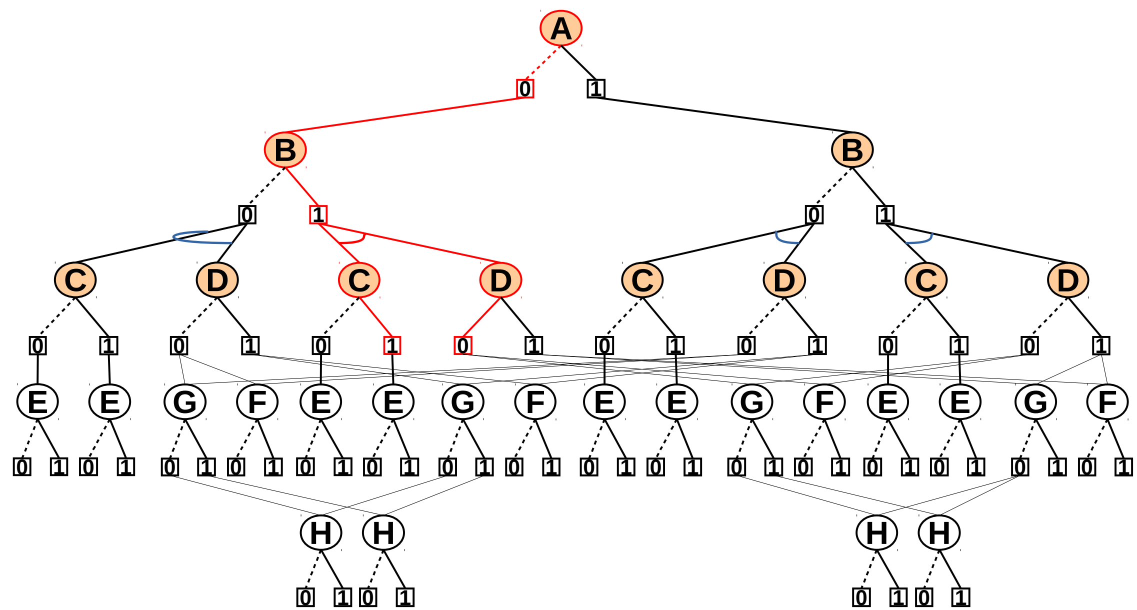
$$\mathbf{X}_M = \{A, B, C, D\}$$
$$\mathbf{X}_S = \{E, F, G, H\}$$

Background – AND/OR search space

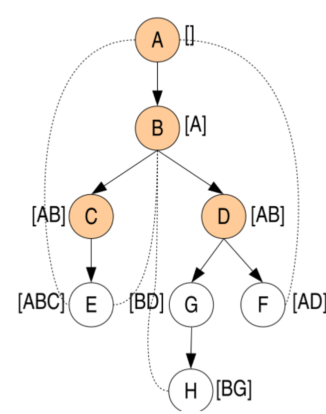
- Bucket elimination [Dechter, 1999]



- AND/OR search graph [Mateescu, Dechter, 2007]

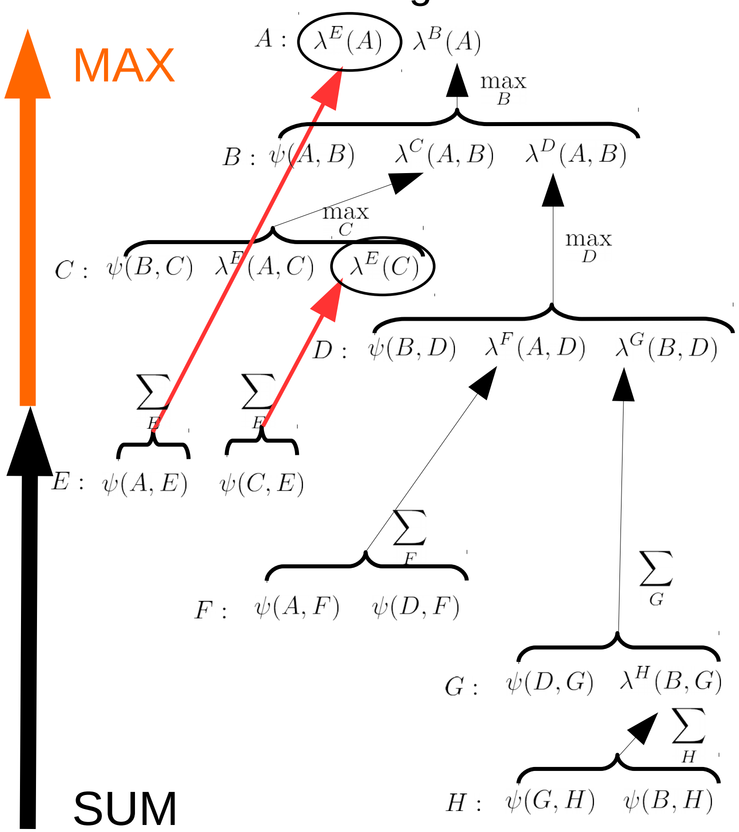


- Pseudo tree [Freuder, Quinn, 1985]



Background - WMB heuristics

- Mini-bucket elimination [Dechter, Rish 2001]
  - "i-bound", limit on the number of variables in a single mini-bucket



  - Mini-bucket upper bound

$$\sum_E [\psi(A,E)\psi(C,E)] \le [\sum_E \psi(A,E)][\sum_E \psi(C,E)]$$

- Weighted Mini-bucket [Liu, Ihler, 2012]
  - Holder's inequality

$$\sum^{w} [\prod_r (\psi_r)] \le \prod_r [\sum_x^{w_r}]$$

$$\sum_x^{w} f(x) \triangleq [\sum_x f(x)^{\frac{1}{w}}]^w \qquad w = \sum_r w_r$$

- WMB Moment Matching [Liu, Ihler, 2011] [Marinescu,Ihler,Dechter, 2014]
  - MAP variables

$$\psi_{kr} = \psi_{kr}\left(\frac{\mu}{\mu_r}\right)$$

$$\mu_r = \max_{\mathbf{Y}_r} \psi_{kr}; \; \mu = \left(\prod_r \mu_r\right)^{1/R}$$

  - SUM variables

$$\psi_{kr} = \psi_{kr}\left(\frac{\mu}{\mu_r}\right)^{w_{kr}}$$

$$\mu_r = \sum_{\mathbf{Y}_r} (\psi_{kr})^{1/w_{kr}}; \; \mu = \prod_r (\mu_r)^{w_{kr}}$$

Previous MMAP search algorithms

2003

Park, Darwiche
Depth-First BnB
Join-tree upper bound
(relaxed variable ordering)

- depth-first search
- dynamic heuristic

2009

Yuan, Hansen
Depth-First BnB
Incremental
Join-tree upper bound

- static heuristic

2014

Marinescu, Dechter, Ihler
Depth-First BnB
AND/OR Search
WMB Heuristic

- compact AND/OR search space
- more accurate WMB heuristics

2015

Marinescu, Dechter, Ihler
Best-First/Recursive BF
AND/OR Search
WMB heuristic

- BF avoids solving summation problems
- very memory intensive
- no anytime, return optimal solution or no solution

2016

Lee,Marinescu, Dechter, Ihler
Weighted Best-First
Anytime Depth-First AND/OR
WMB heuristic

- anytime solutions
- infrequent solution updates
- still memory intensive

2017

Marinescu, Lee, Iihler, Dechter
Best+Depth-First
- high quality upper/lower bounds
- more frequent solution updates
- memory efficiency

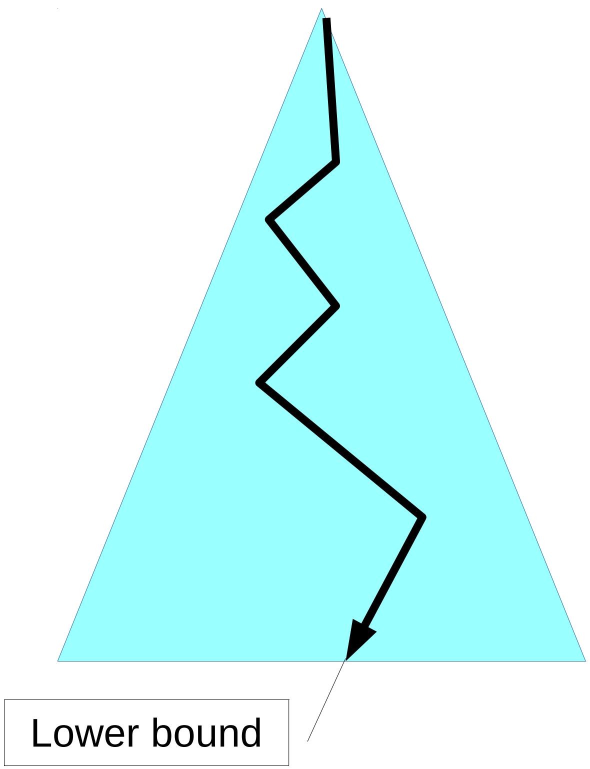
Outline

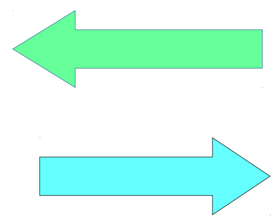
- Background
  - Graphical model
  - AND/OR search space & WMB heuristic
- Best+Depth-First search for MMAP
  - LAOBF (Best-First AND/OR Search with Depth-First Lookaheads)
  - AAOBF (Alternating Best-First and Depth-First AND/OR search)
  - LnDFS (Learning Depth-First AND/OR search)
- Experiments
- Conclusion

Best+Depth-First Search

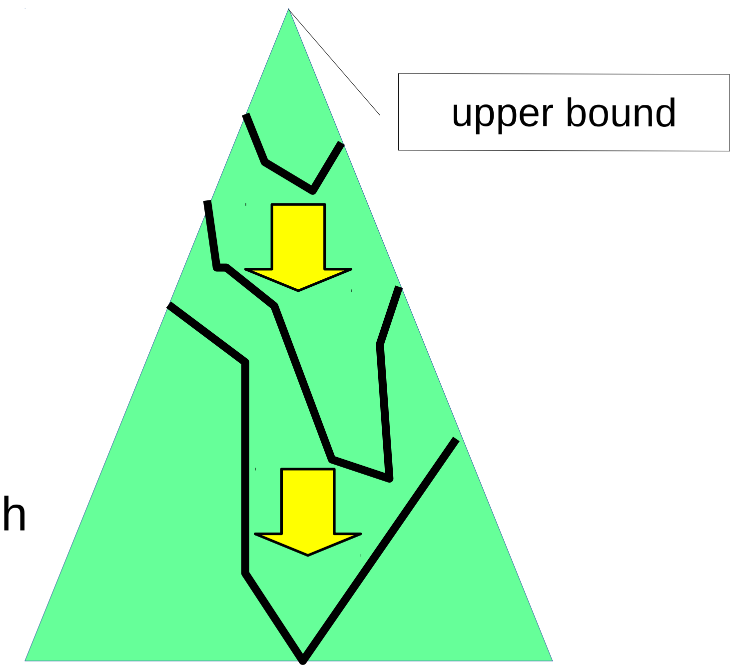
- Better guidance for depth-first dives using improved heuristics
- Frequent solution updates

Depth-First search





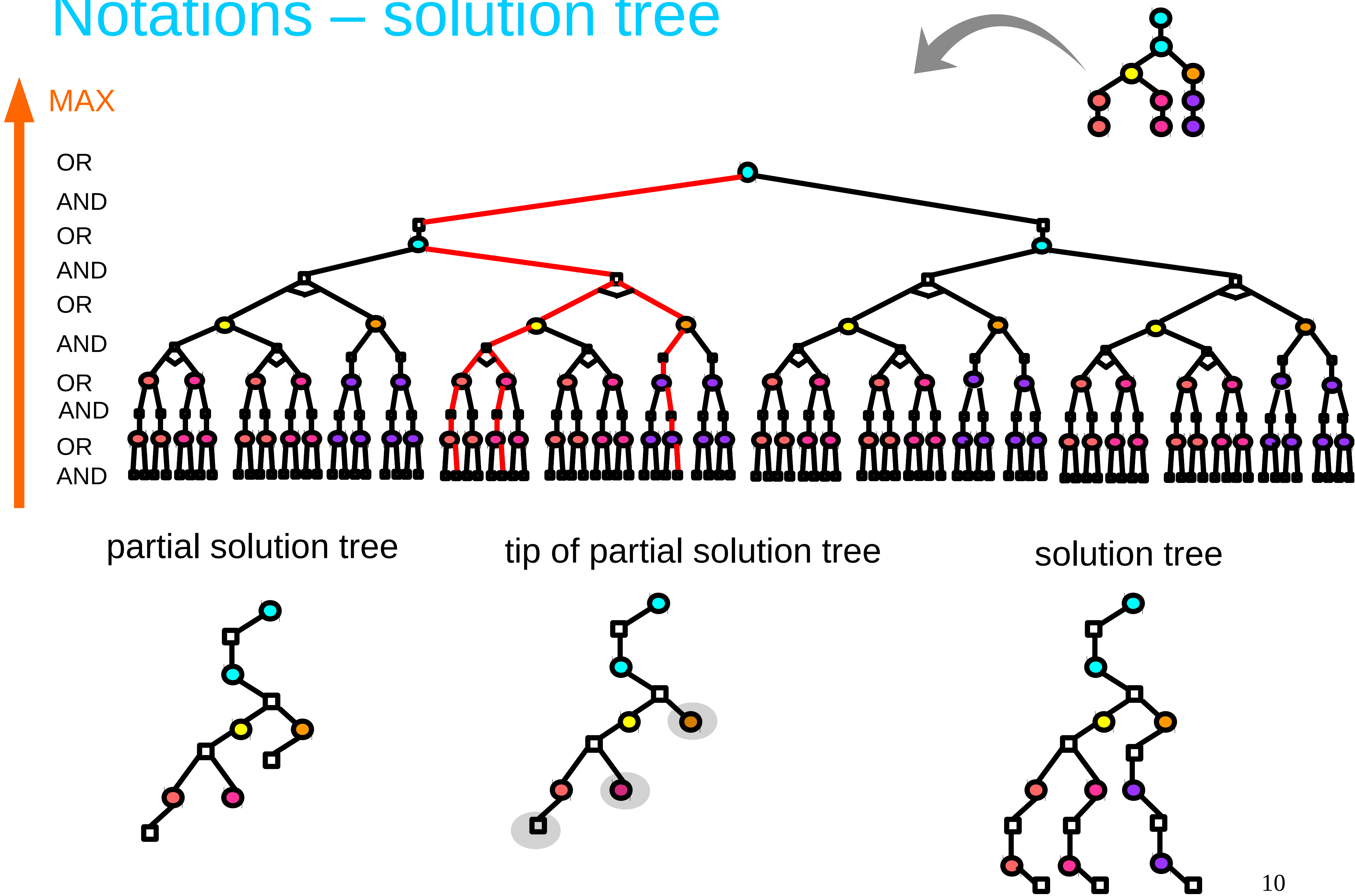
Best-First search



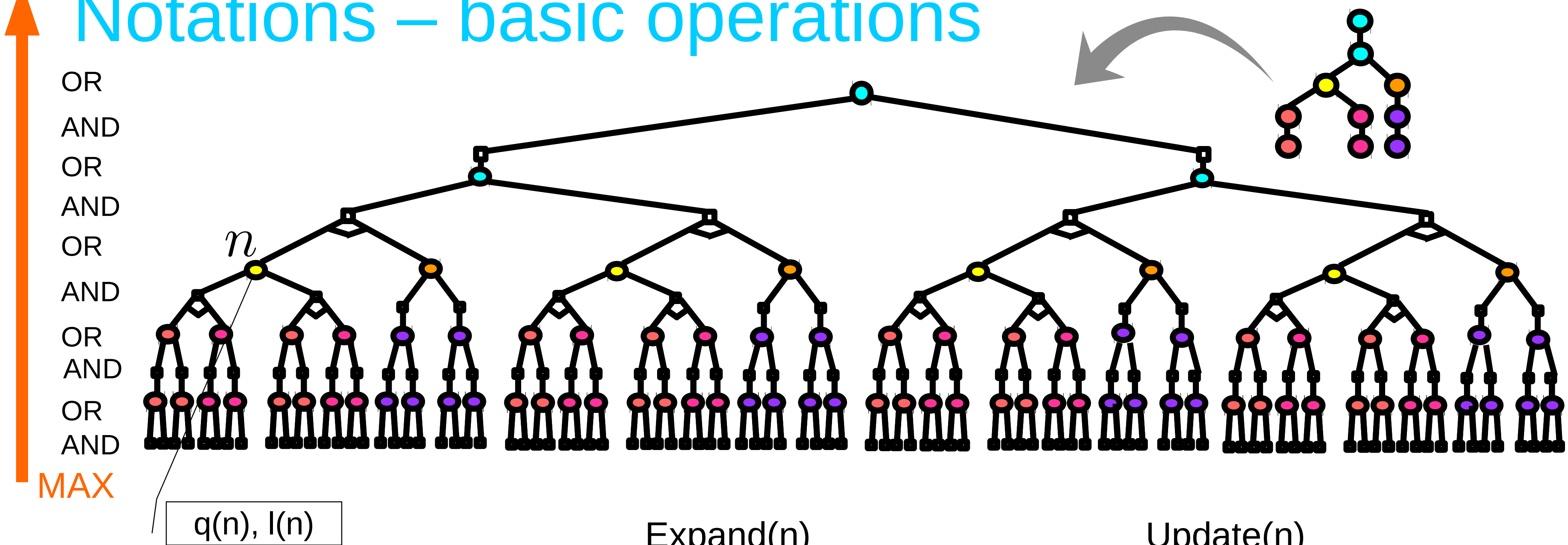
- Cutoff frontier of best-first search using improved lower bounds
- Learn accurate heuristics by depth-first lookahead

When Global UB = Global LB,
Optimal Solution Discovered

Notations – solution tree



Notations – basic operations

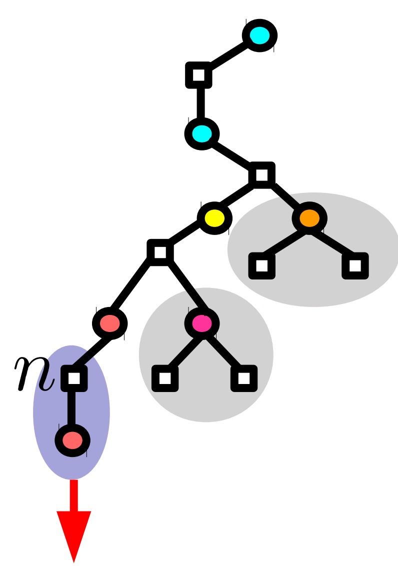


- q(n) : upper bound at n
- q(root) : global upper bound

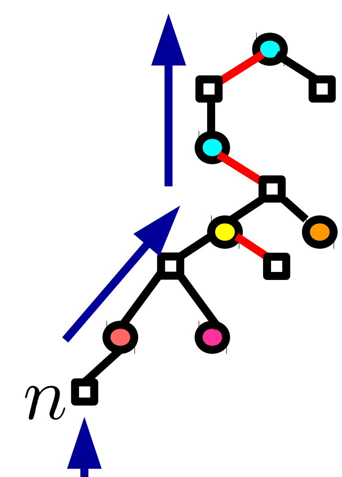
- l(n) : lower bound at n
- l(root) : global lower bound

- T_b : best partial solution tree (partial solution tree where OR nodes direct the child m with best q(m)

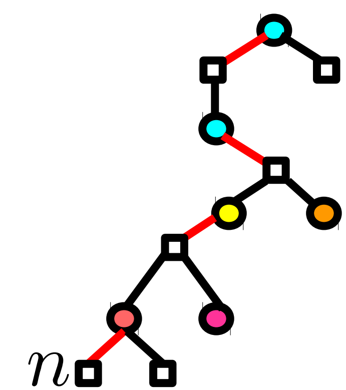
Expand(n)



Update(n)



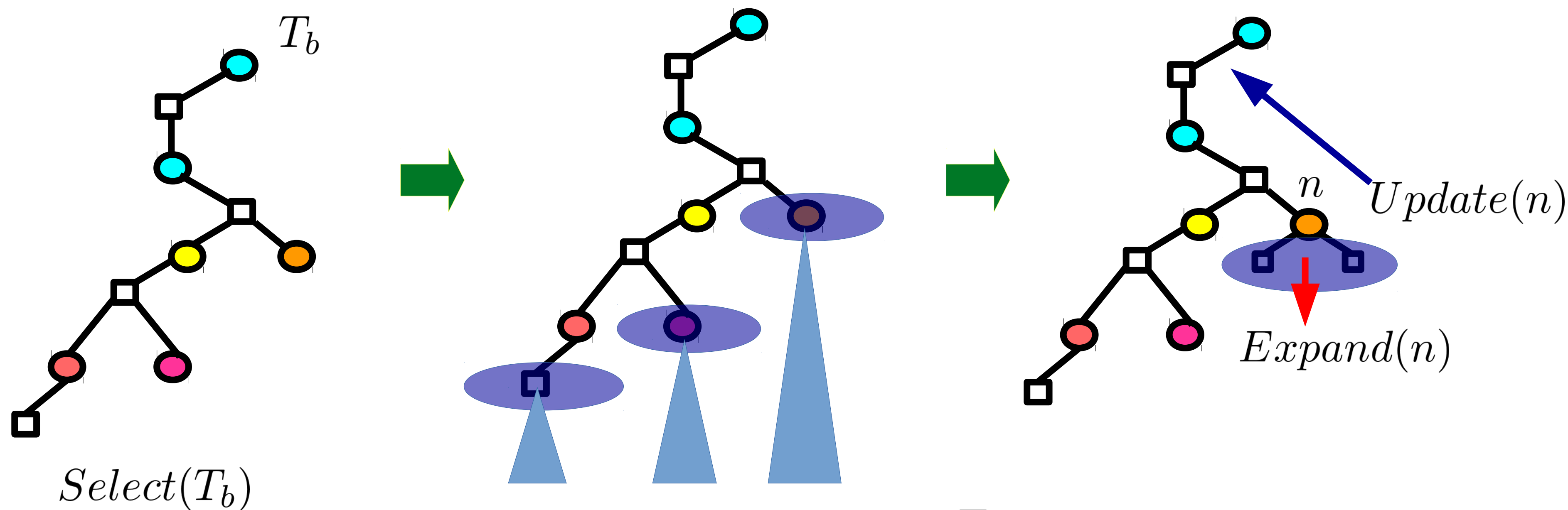
- backup q and l values



- re-direct best partial solution tree

LAOBF (best-first AND/OR search with depth-first lookaheads)

Best-first selection — Depth-first lookahead — Best-first expansion & update



- depth-first dive at the tip of T_b
- compute global lower bound
- cache summation subproblems

- Select a tip node n
- Expand and Update n

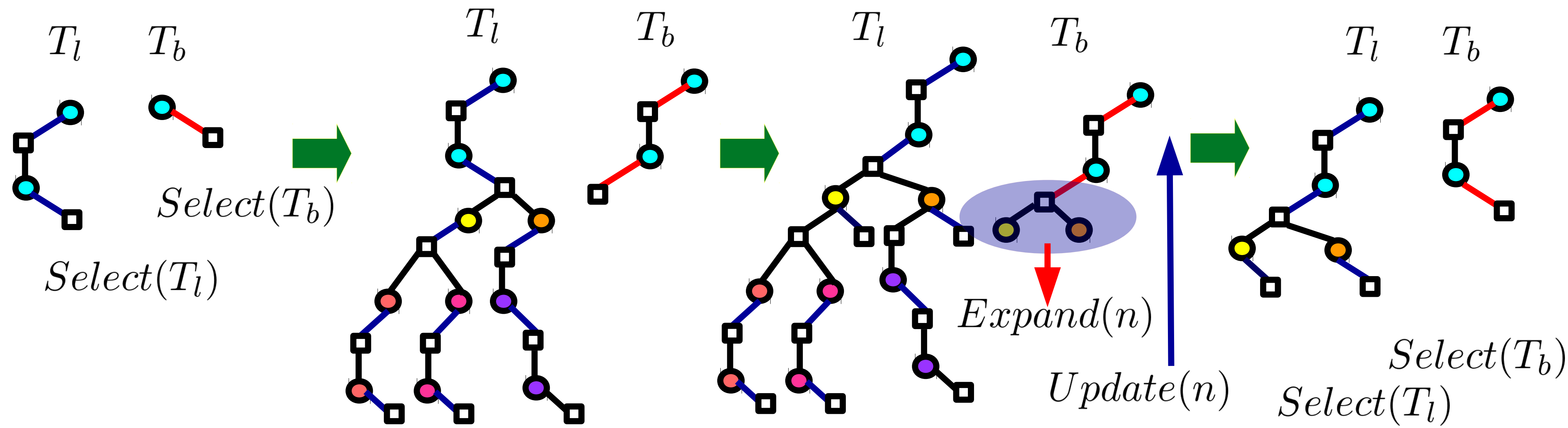
cutoff parameter: control depth-first lookahead (at every θ number of node expansions.)

AAOBF (alternating best-first with depth-first AND/OR search)

Depth-first greedy expansion T_l
Best-first re-direct T_b

Depth-first re-direct T_l
Best-first expansion & update T_b

Depth-first selection
Best-first selection



- Expand(n) and Update(n)

  $n \in T_l$

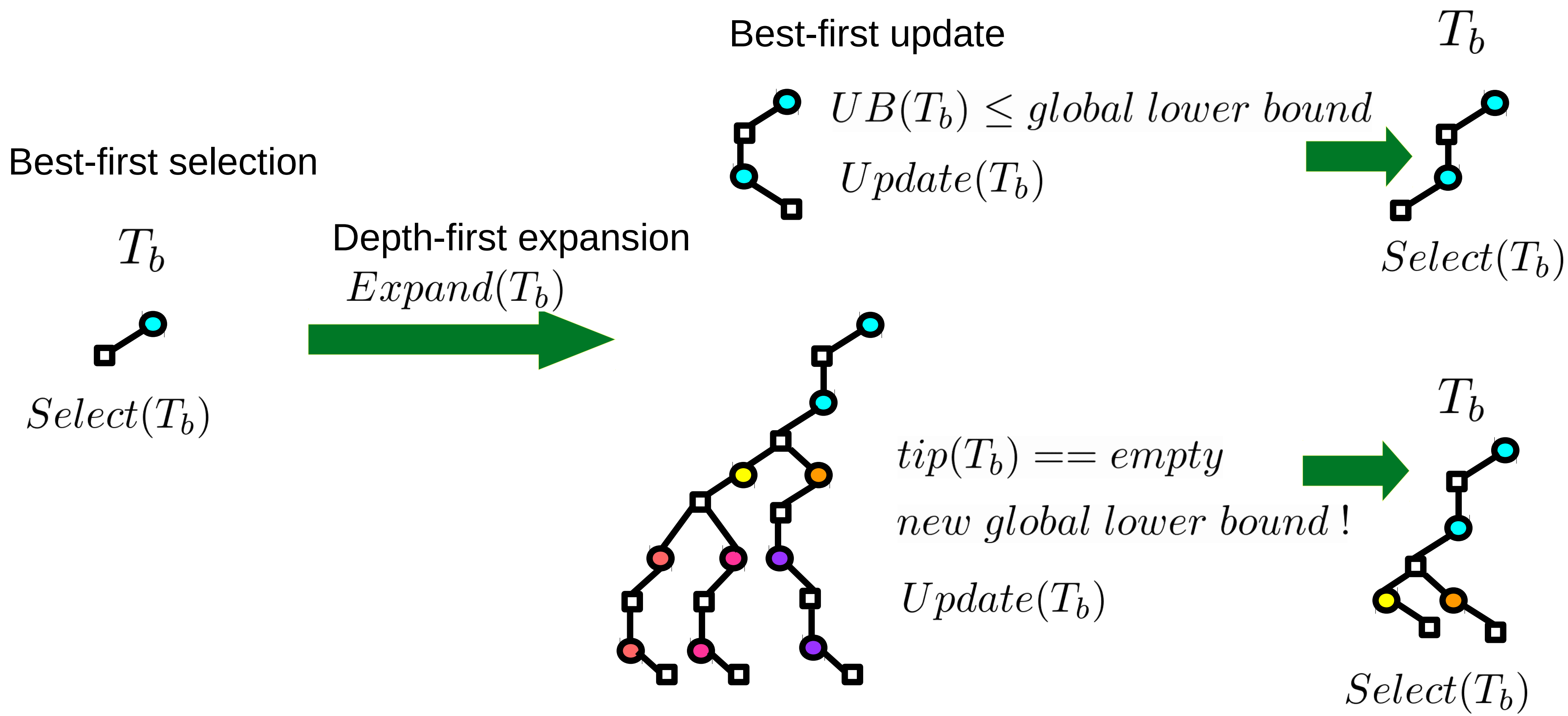
- depth-first greedy search

  $\max w(n,m)q(m)$

- redirect T_l from explicated search graph from the root with updated q and l $\min \frac{l(n)}{w(n,m)q(m)}$

- select T_b
  Expand and Update a tip node

LnDFS (learning depth-first AND/OR search)



$Expand(T_b)$ Keep expanding tips nodes of T_b

$Update(T_b)$ Update values from tip nodes of T_b

Outline

- Background
  - Graphical model
  - AND/OR search space & WMB heuristic
- Best+Depth-First search for MMAP
  - LAOBF (Best-First AND/OR Search with Depth-First Lookaheads)
  - AAOBF (Alternating Best-First and Depth-First AND/OR search)
  - LnDFS (Learning Depth-First AND/OR search)
- Experiments
- Conclusion

Experiments

- Anytime Algorithms
  - Presented Best+Depth-First Search
    - LAOBF $\theta = 1000$
    - AAOBF
    - LnDFS
  - State-of-the-art
    - Weighted Recursive Best-First AND/OR Search with Overestimation $w_{i+1} = \sqrt{w_i}$ $w_0 = 64$ [Lee, Marinescu, Ihler, Dechter, 2016]
    - Breadth Rotate AND/OR Branch and Bound [Lee, Marinescu, Ihler, Dechter, 2016]
    - Anytime Factor Set Elimination [Maua, Campos, 2012]
- Memory
  - total 24 GB
  - WMB-MM(i) i-bound: 20 or the largest within 4 GB
  - caching for AND/OR search graph max 4 GB

Experiment

- Benchmark
  - derived from UAI inference competitions for MPE query
  - randomly choose 50% of the variables as MAP variables
  - generate 4 random MMAP instances
  - Grid, Pedigree, Promedas domain

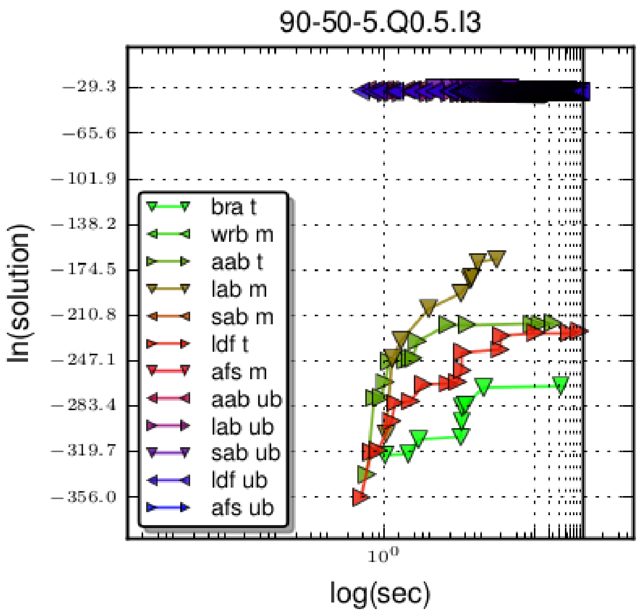
- Problem instance parameters

Domain (#. instances)	$N_{min} N_{ave} N_{max}$	$F_{min} F_{ave} F_{max}$	$K_{min}\ K_{max}$	$S_{min}\ S_{max}$	$W_{min} W_{ave} W_{max}$	$H_{min} H_{ave} H_{max}$
Grid (128)	144,649,2500	144,649,2500	2,2	3,3	25,163,814	42,189,834
Pedigree (88)	334,917,1289	334,917,1289	3,7	4,5	35,127,289	63,152,312
Promedas (100)	381,1064,1997	385,1077,2024	2,2	3,3	11,137,552	33,171,577

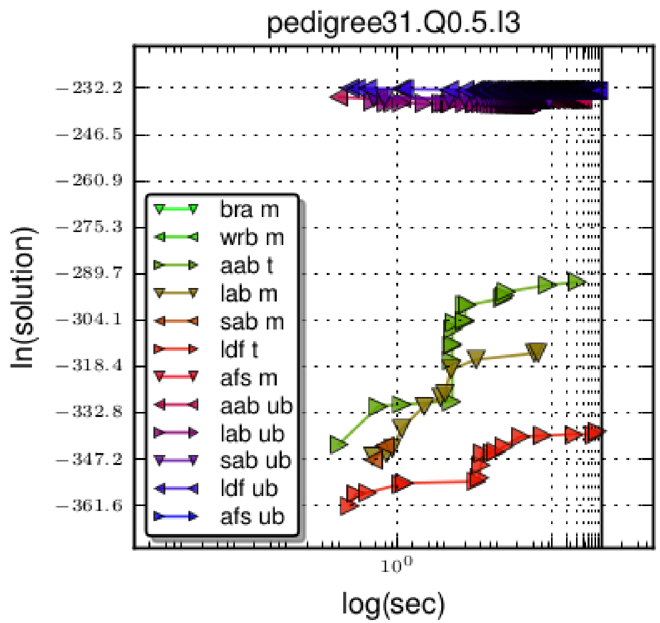
N: number of variables, F: number of functions, K: domain size, S: scope size
W: constrained induced width, H: constrained pseudo tree height

Experiment – individual instances

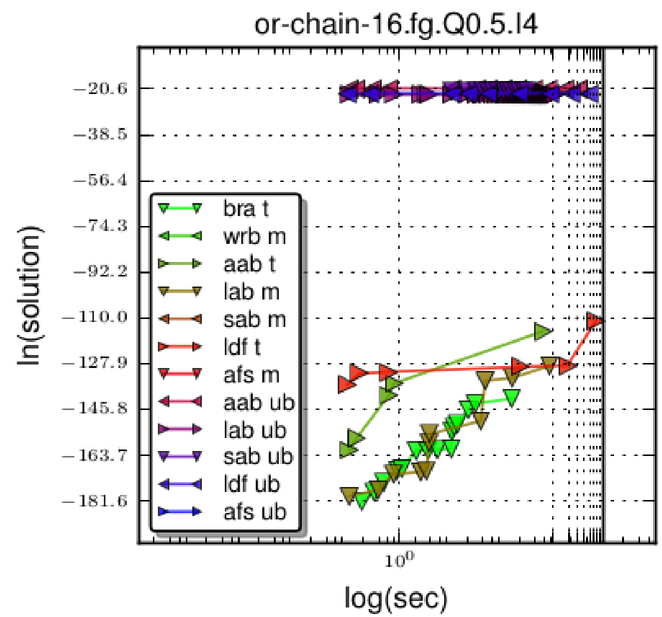
- Anytime search status for individual instances



N:2500 F:2500 K:2 S:3
W:788 H:817



N:1183 F:1183 K:5 S:5
W:272 H:290



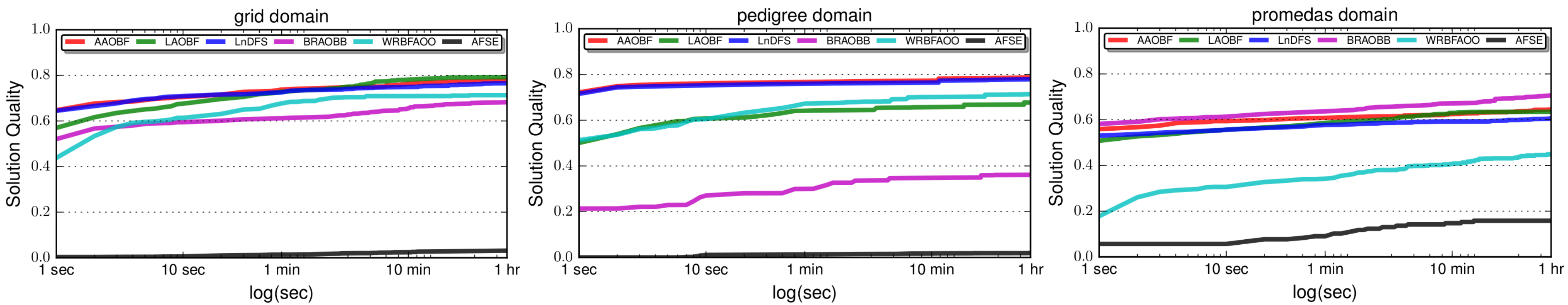
N:1675 F:1701 K:2 S:3
W:259 H:298

- search: LAOBF (lab), AAOBF (aab), LnDFS (ldt), BRAOBB (bra)
- heuristic: WMB-MM (20)
- memory: 24 GB

Other algorithms couldn't find any solution due to memory out

Experiment - average solution quality

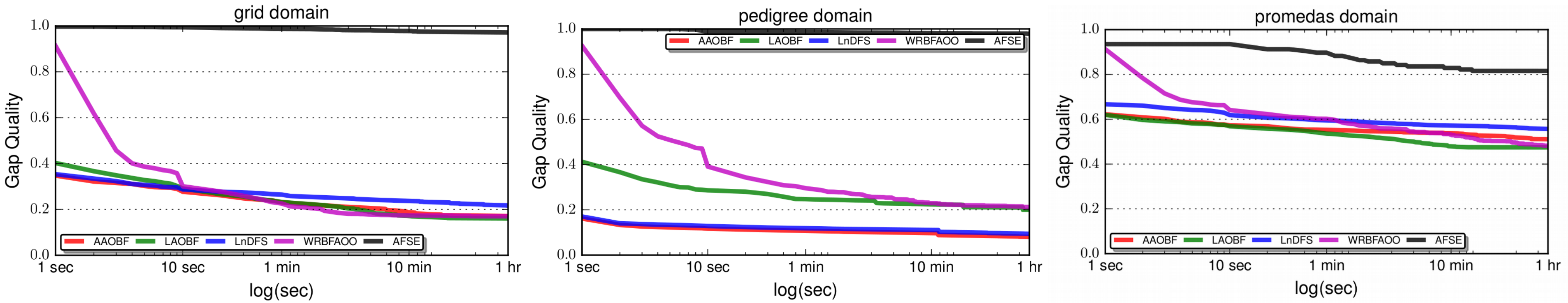
- Average solution quality $Ave[\frac{best\ solution\ found}{optimal\ solution}]$
  - anytime quality of lower bound normalized by optimal solution
  - when optimal solution is not available, used best-known solution



- Result
  - How the quality of solution improves over time
  - LAOBF, AAOBF, LnDFS
    - improved upon WRBFAOO on 3 domains
  - BRAOBB
    - best on promedas domain, second worst on pedigree domain
  - AFSE: worst performance on 3 domains

Experiment - average gap quality

- Average gap quality $Ave[\frac{upper\ bound\ -\ lower\ bound}{upper\ bound}]$
  - anytime gap (difference between upper and lower bound) normalized by upper bound (If no lower bound available, gap = 1)



- Result
  - How the gap between lower/upper bound decreases over time (gap=0 optimal)
  - LAOBF, AAOBF, LnDFS
    - All similar improvements over time, especially at shorter time bounds
    - AAOBF was overall best
  - AFSE: worst performance on 3 domains

Experiment – memory robustness

- Memory robustness

	grid				pedigree				promedas			
algorithm	M_{out}	M_{∞}	$\overline{\Delta_G}$	$\overline{T_M}$	M_{out}	M_{∞}	$\overline{\Delta_G}$	$\overline{T_M}$	M_{out}	M_{∞}	$\overline{\Delta_G}$	$\overline{T_M}$
AAOBF	22%	0%	7%	1208s	13%	0%	6%	1596s	0%	0%	-%	-s
LAOBF	48%	0%	32%	1345s	59%	5%	17%	1418s	62%	0%	77%	1012s
WRBFAOO	41%	11%	42%	645s	53%	10%	38%	707s	28%	28%	100%	749s
LnDFS	0%	0%	-%	-s	1%	0%	27%	942s	0%	0%	-%	-s
AFSE	77%	76%	99.9%	64s	83%	83%	100%	21s	74%	74%	100%	58s

  - How search algorithm effectively utilized the memory and improves gap within the memory limit
  - M_{out} % of instances terminated by memory limit
  - M_{∞} % of instances terminated by memory limit and no solution found at all
  - $\overline{\Delta_G}$ average gap computed from out of memory instances only
  - $\overline{T_M}$ average search time computed from out of memory instances

- Result
  - LnDFS is the most memory robust algorithm
  - AAOBF (LAOBF) improved memory robustness compared to WRBFAOO
  - AFSE is the worst among 5 algorithms

Conclusion

- Anytime Best+Depth-First search algorithms improved upon the state-of-the-art algorithms
  - higher quality anytime solutions
  - tighter anytime upper bounds
  - more effective use of memory

- Future work
  - New anytime search + approximate summation inference
    - variational bounds with search
    - probabilistic bounds from sampling