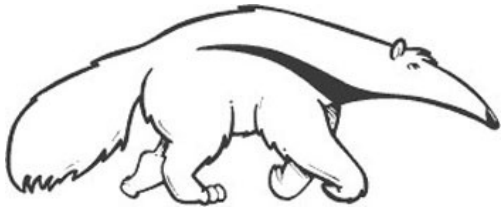
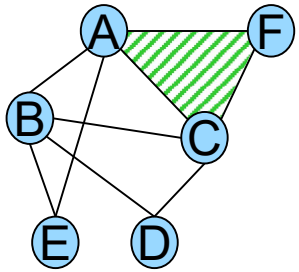
Graphical Models Meet Heuristic Search: A Personal Journey for Automating Reasoning

Rina Dechter

University of California, Irvine

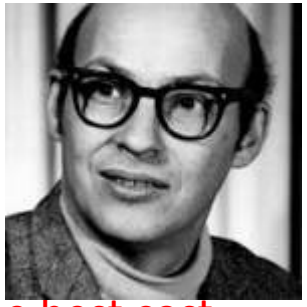




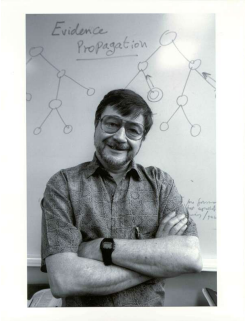
In this talk…

The Tractable Islands Paradigm

[Dechter and Pearl, JACM 1985]



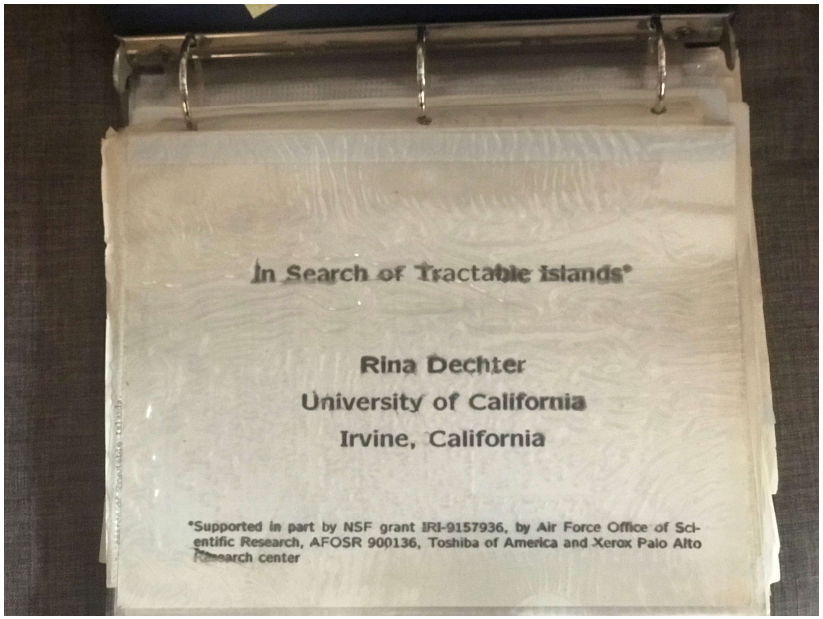
"An important component of human problem-solving expertise is the ability to use knowledge about solving easy problems to guide the solution of difficult ones." - Minsky

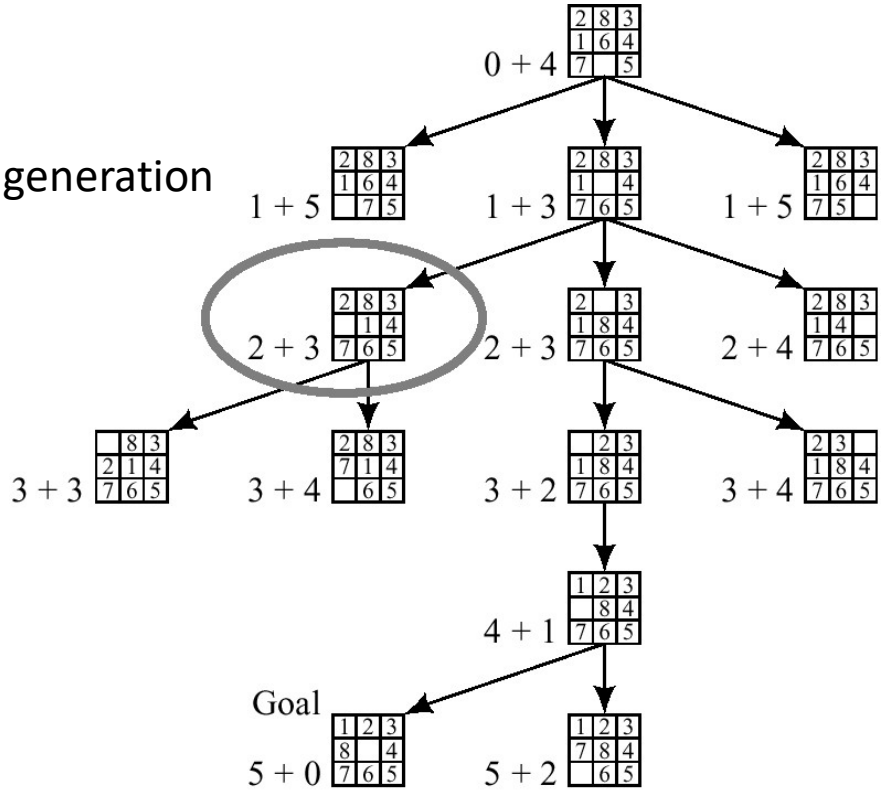


[Pearl 1983, *On the discovery and generation of certain Heuristics, AI Magazine*]

h(n):Heuristic underestimates the best cost from a node to the solution

3 steps: a) simplification, b) solution, c) advice generation

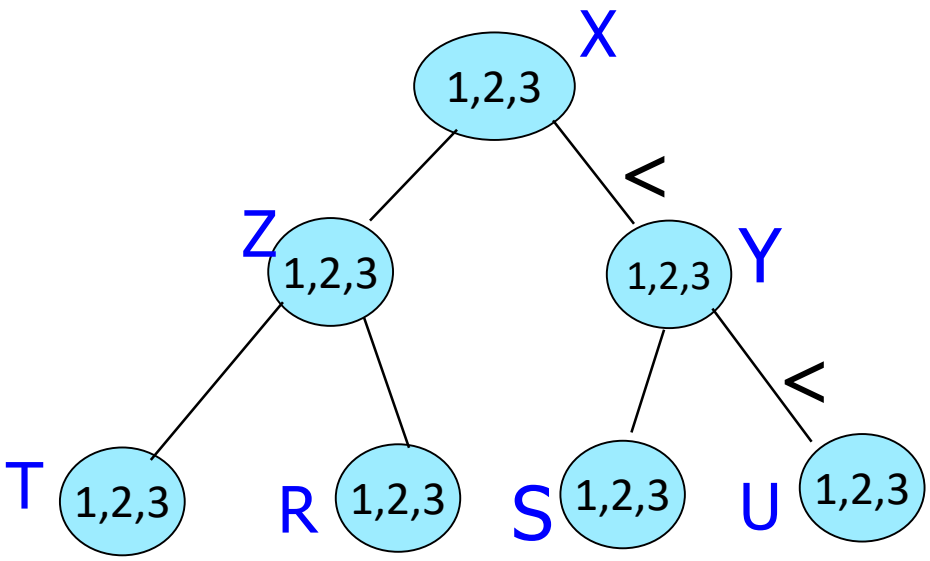




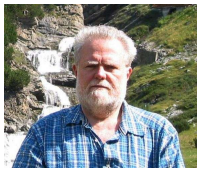
Tree-Solving is Easy

[Mackworth and Freuder. The complexity of some polynomial network consistency algorithms for constraint satisfaction problems. *AIJ*, 25, 1985.]
[Ugo Montanaru "Constraint Networks", 1977]

CSP – consistency (projection-join)




Freuder


Montanari

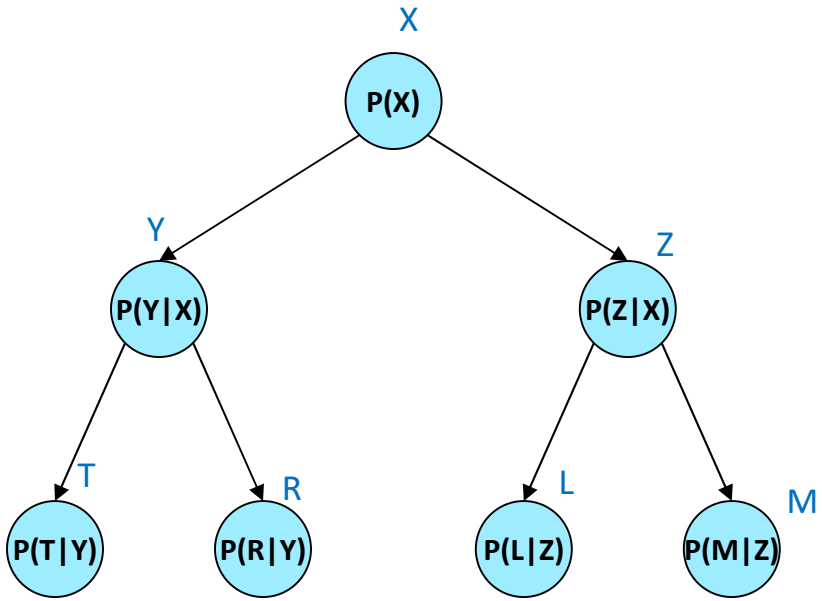

Mackworth

Trees are processed in linear time and memory

Tree-Solving is Easy

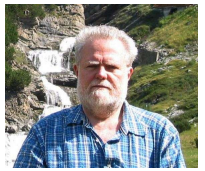
(Mackworth and Freuder. The complexity of some polynomial network consistency algorithms for constraint satisfaction problems. *AIJ*, 25, 1985.)
Ugo Montanaru "Constraint Networks", 1977

Belief updating (sum-prod)



CSP – consistency (projection-join)


Freuder


Montanari


Mackworth

MPE (max-prod)

#CSP (sum-prod)

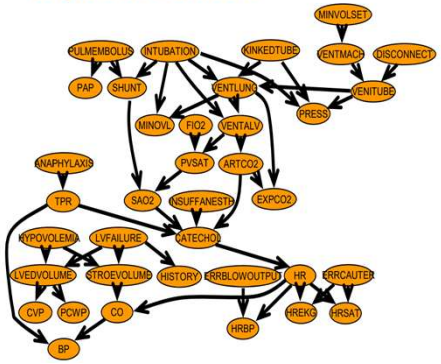
Trees are processed in linear time and memory

Graphical Models - Overview

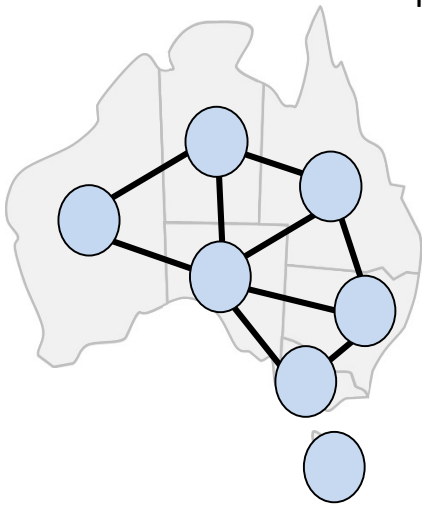
Describe structure and interdependence in a model of the world

"Alarm" network:
patient monitoring

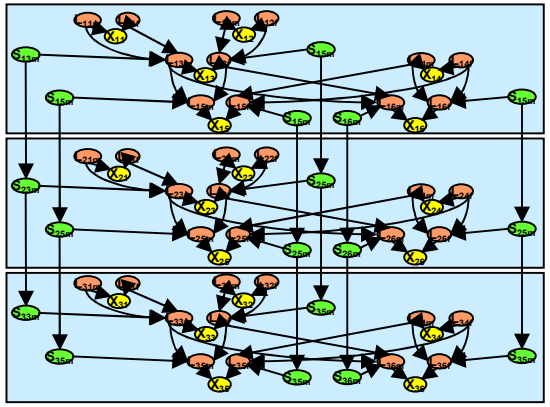
Bayesian Networks



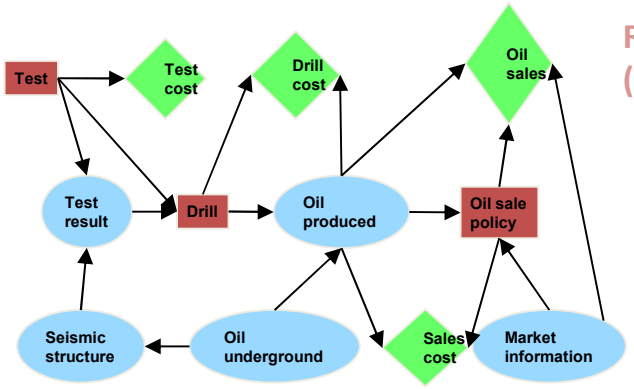
Constraint satisfaction



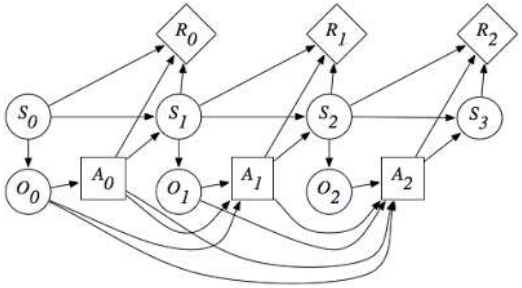
Pedigree network: genetic inheritance

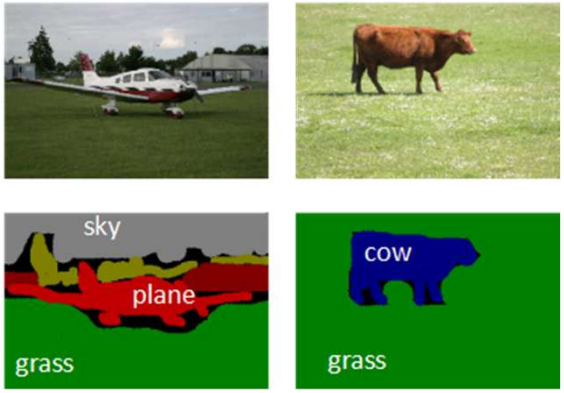


Influence Diagrams



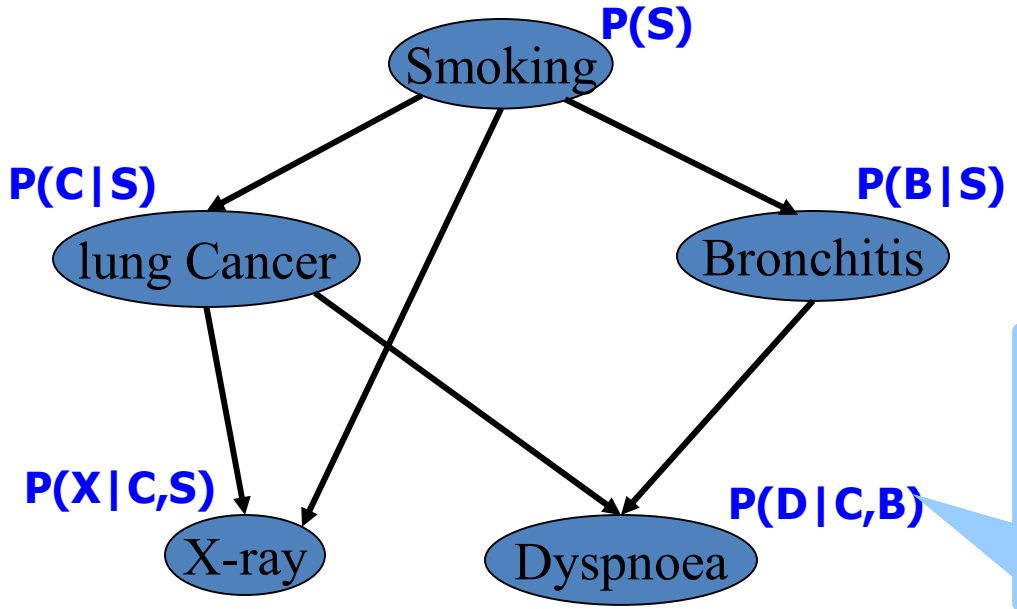
Reinforcement Learning (MDPs)
(Model-based vs. Model-free)





Bayesian Networks (Pearl 1988)

Queries: prediction, diagnosis, classification, decision making, Causal inference



CPD:

C	B	D=0	D=1
0	0	0.1	0.9
0	1	0.7	0.3
1	0	0.8	0.2
1	1	0.9	0.1

$$P(S, C, B, X, D) = P(S)\, P(C|S)\, P(B|S)\, P(X|C,S)\, P(D|C,B)$$

Conditional Independencies ➡ Efficient Representation

P (lung cancer=yes | smoking=no, dyspnoea=yes) = ?

Graphical Models

(finite, discrete domains)

A *graphical model* consists of:

$X = \{X_1, \ldots, X_n\}$ -- variables

$D = \{D_1, \ldots, D_n\}$ -- domains

$F = \{f_{\alpha_1}, \ldots, f_{\alpha_m}\}$ -- functions or "factors"

and a *combination operator* ⊗

(we'll assume discrete)

Example:

$A \in \{0, 1\}$

$B \in \{0, 1\}$

$C \in \{0, 1\}$

A	B	f(A,B)
0	0	2
0	1	4
1	0	3
1	1	1

$f_{AB}(A, B), \quad f_{BC}(B, C)$

The ***combination operator*** defines an overall function from the factors, e.g., "x" :

$$p(A, B, C) \propto f_{AB}(A, B) \times f_{BC}(B, C)$$

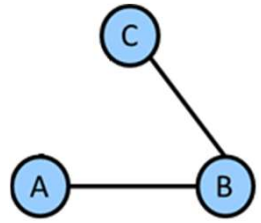
Inference: compute quantities of interest about the distribution, e.g.,

$$p(x_i) = \frac{1}{Z} \sum_{\mathbf{x} \setminus x_i} \prod_{\alpha} f_\alpha(\mathbf{x}_\alpha)$$

(marginals)

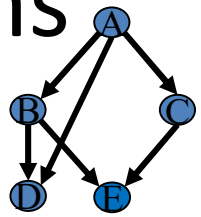
or $$Z = \sum_{\mathbf{x}} \prod_{\alpha} f_\alpha(\mathbf{x}_\alpha)$$

(partition function)



Primal graph

Graphical Models Reasoning Problems



- **Reasoning:** Algorithms for answering queries

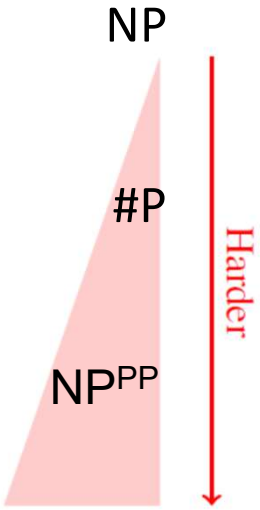
Queries: prediction, diagnosis, classification, decision making, causal effects

Causal effects

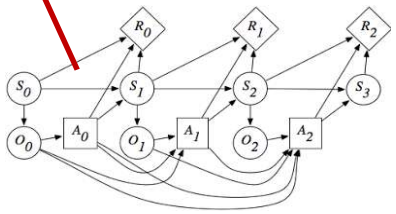
Constraint Satisfaction/Satisfiability

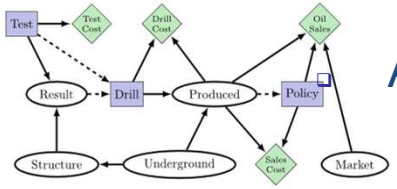
Max-Inference:	$f(x^*) = \max_x \prod_\alpha f_\alpha(x_\alpha)$
Sum-Inference:	$Z = \sum_x \prod_\alpha f_\alpha(x_\alpha)$
Mixed-Inference (MMAP):	$f_M(x_M^*) = \max_{x_M} \sum_{x_S} \prod_\alpha f_\alpha(x_\alpha)$
Mixed-Inference (MEU):	$\text{MEU} = \max_{D_1,\ldots,D_m} \sum_{X_1,\ldots X_n} (\prod_{P_i \in P} P_i) \times (\sum_{r_i \in R} r_i)$

$e^{\text{tree-width}}$



Influence diagrams & Model-based Reinforcement Learning

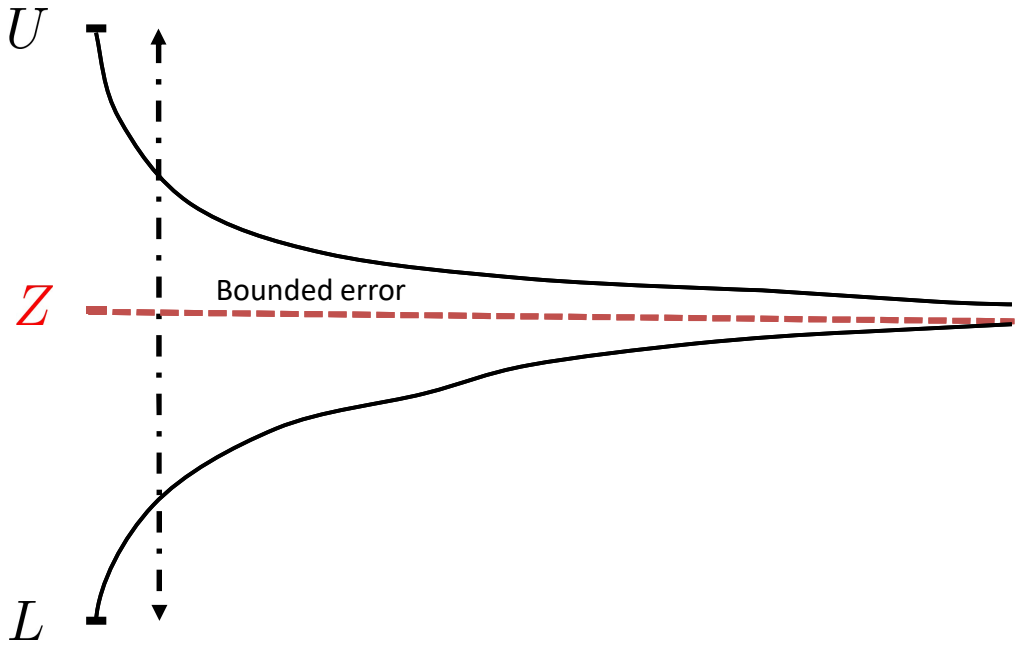




All these tasks are NP-hard

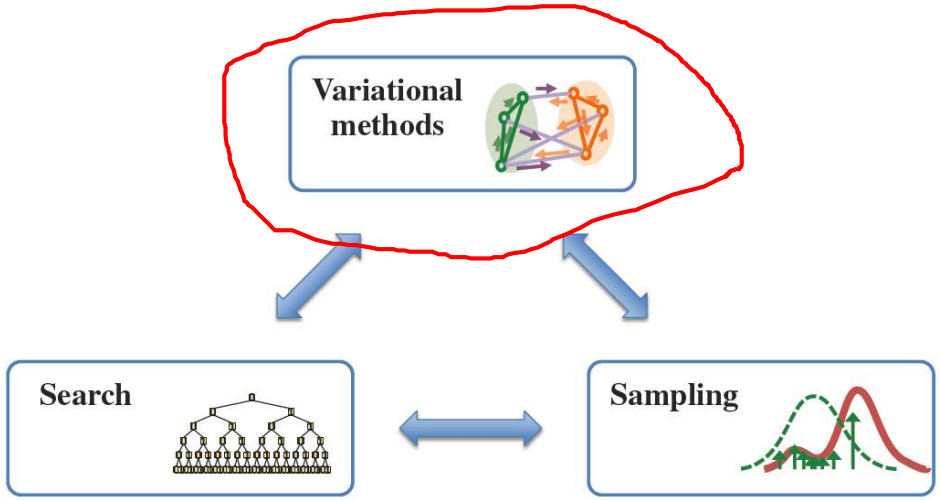
- exploit problem structure
- identify tractable cases
- Approximate
- Anytime algorithms

Goal: Anytime Algorithms



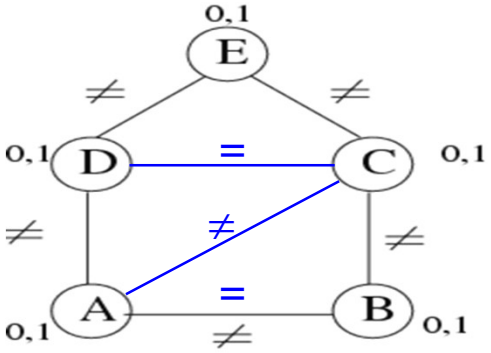
- Algorithms that improve with time and will not be hindered by memory constraints
- Error-bounds that get tighter with time
- With enough time we can get an exact solution
- We can trade computational resources for quality in an anytime fashion

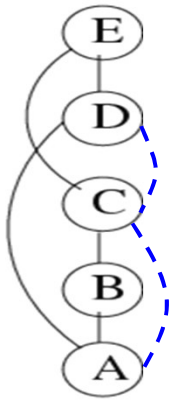
Bucket Elimination



Bucket Elimination

Adaptive Consistency [Dechter & Pearl, AIJ 1987]





Bucket E: $E \neq D,\ E \neq C$

Bucket D: $D \neq A$ → $\mathbf{D = C}$

Bucket C: $C \neq B$ → $\mathbf{A \neq C}$

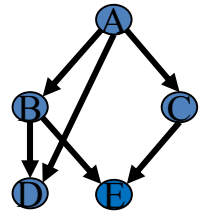
Bucket B: $B \neq A$ → $\mathbf{B = A}$

Bucket A: → **contradiction**

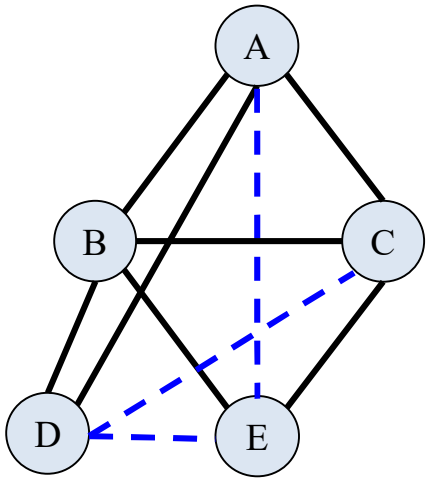
Complexity: $O(n \exp(w^*))$

w^* – induced width

Belief Updating



- p(X | Evidence) = ?



"primal" graph

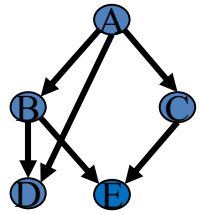
$$p(A|E=0)$$
$$\propto p(A, E=0)$$
$$= \sum_{e,d,c,b} p(A)\, p(b|A)\, p(c|A)\, p(d|b,A)\, p(e|b,c)\, \mathbb{1}[e=0]$$

$$p(A) \sum_{e} \sum_{d} \sum_{c} p(c|A)\, \mathbb{1}[e=0] \sum_{b} p(b|A)\, p(d|b,A)\, p(e|b,c)$$

$$\lambda_{B \to C}(a, d, c, e)$$

Variable Elimination

Bucket Elimination



Algorithm *BE-bel* [Dechter 1996]

$$p(A|E=0) = \alpha \sum_{e,d,c,b} p(A)\, p(b|A)\, p(c|A)\, p(d|A,b)\, p(e|b,c)\, \mathbb{1}[e=0]$$

$\sum_b \prod$ ← Elimination & combination operators

bucket B: $p(b|A)\ p(d|b,A)\ p(e|b,c)$

bucket C: $p(c|A)$ $\lambda_{B\to C}(A,d,c,e)$

bucket D: $\lambda_{C\to D}(A,d,e)$

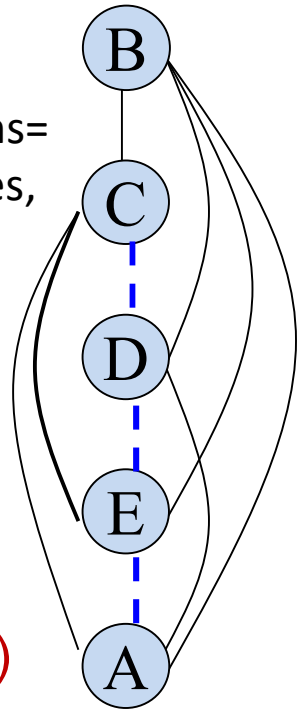
bucket E: $\mathbb{1}[E=0]\ \lambda_{D\to E}(A,e)$

bucket A: $p(A)$ $\lambda_{E\to A}(A)$

$p(E=0)$

$p(A|E=0) = p(A,E=0)\,/\,p(E=0)$

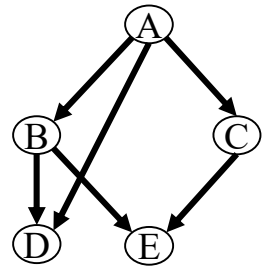
Functions= messages, tensors



W*=4
"induced width"
(max clique size)

For Map replace sum with max

Complexity of Bucket Elimination;



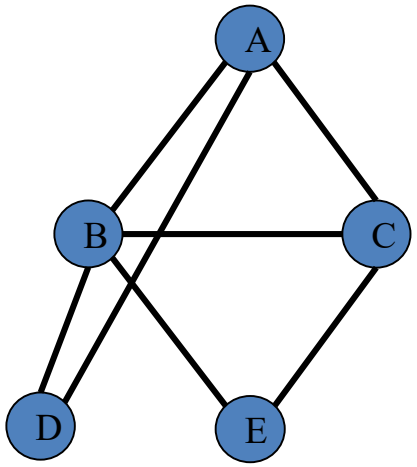
Bucket Elimination is **time** and **space**

$$O(r\, k^{w^*(d)})$$

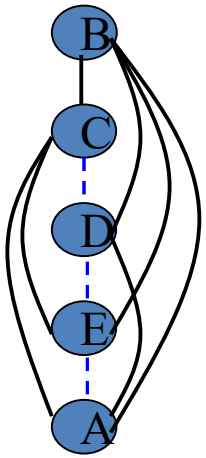
$w^*(d)$ – the induced width of graph along ordering d

r = number of functions

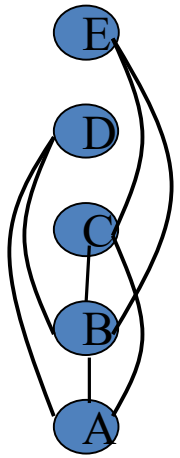
The effect of ordering:



"Moral" graph



$w^*(d_1) = 4$



$w^*(d_2) = 2$

[Kask, Lam, Won tw 2014]

Otten

Finding the smallest induced width (w*) is hard!

Gogate

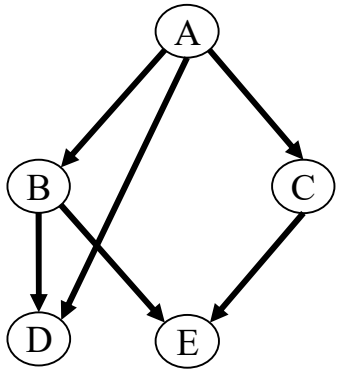
[Gogate, Dechter, UAI 2004] [Kask, Gelfand, Otten, Dechter, AAAI 2011]

Gelfand

Kask

Bucket Tree Elimination

RD2



$$p(A|E=0) = \alpha \sum_{e,d,c,b} p(A)\, p(b|A)\, p(c|A)\, p(d|A,b)\, p(e|b,c)\, \mathbb{1}[e=0]$$

E: $p(E|B,C)\mathbb{1}[E=0]$

D: $p(D|A,B)$

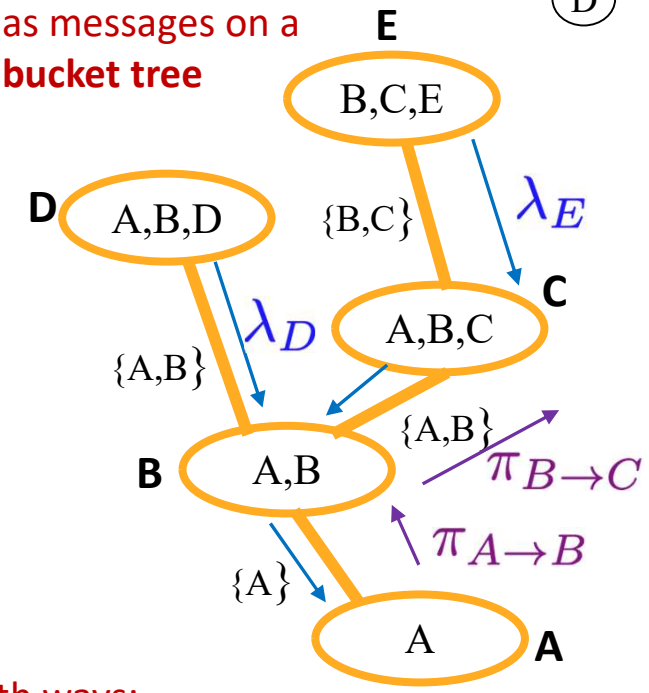
C: $p(C|A)\quad \lambda_E(B,C)$

B: $\lambda_D(A,B)\quad p(B|A)\quad \lambda_C(A,B)$

A: $p(A)\ \lambda_B(A)$

Z = p(E=0)

View elimination as messages on a **bucket tree**



Can pass messages both ways:

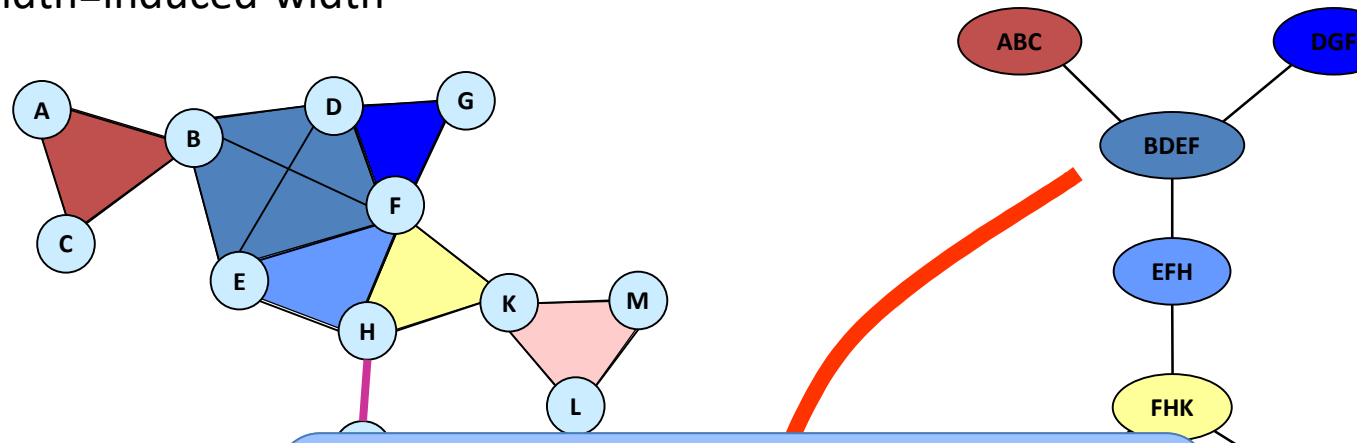
$$p(C|E=0) \propto p(C|A)\ \lambda_E(B,C)\ \pi_{B\to C}(A,B)$$

Slide 16

RD2 Rina Dechter, 7/7/2025

In summary:
The Tractable Island of Tree-Decomposition; treewidth (w)

Treewidth=induced-width



Tractable island bounded induced-width
w<30
Solved by Bucket-Elimination

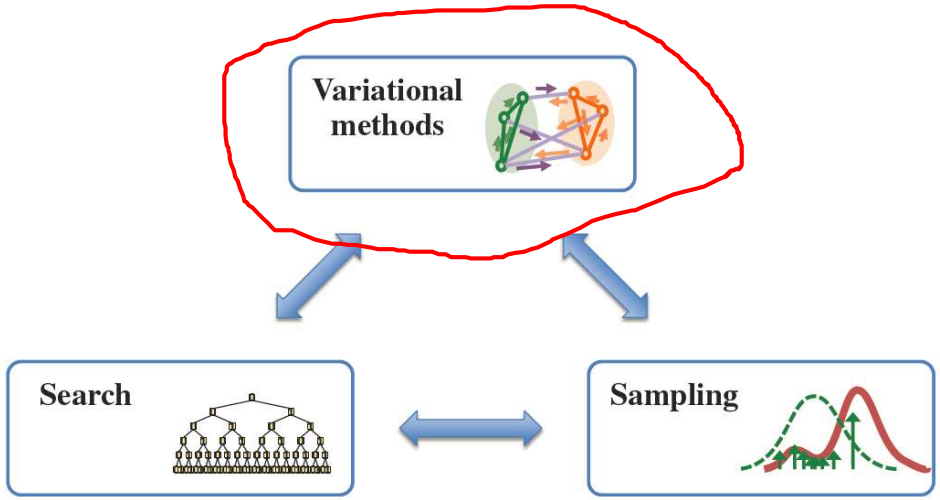
Inference (variati
algorithm:

Time: $O(k^w)$
Space: $O(k^w)$

treewidth (w)= 4 - 1 = 3
treewidth (w) = (maximum cluster size) - 1

Solved by Inference algorithms (BE):
But, memory is the bottleneck

Mini-Bucket Elimination

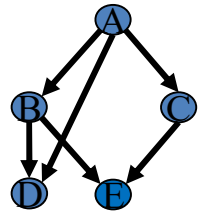


Mini-Bucket Approximation

[Dechter & Rish, JACM 2003]

Rish

For optimization (max-product)



bucket B: $f(a,b)\ f(b,c)\ f(b,d)\ f(b,e)$

bucket C: $f(c,a)\ f(c,e)\ \hat{\lambda}_{B\to C}(a,d,c,e)$

bucket D: $f(a,d)\ \hat{\lambda}_{D\to E}(a,e)$

bucket E: $\hat{\lambda}_{C\to D}(a,d,e)$

bucket A: $f(a)\ \hat{\lambda}_{E\to A}(a)$

Exact answer

mini-buckets

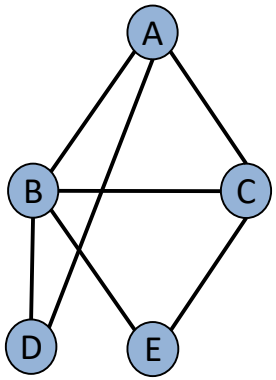
$f(a,b)\ f(b,c)$ $\quad$ $f(b,d)\ f(b,e)$

$f(c,a)\ f(c,e)\ \lambda_{B\to C}(a,c)$ $\quad$ $f(a,d)\ \lambda_{B\to D}(d,e)$

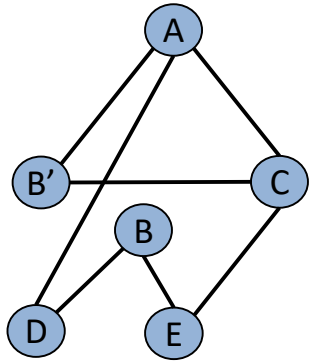
$\lambda_{C\to E}(a,e)\ \lambda_{D\to E}(a,e)$

$f(a)\ \lambda_{E\to A}(a)$

U = upper bound



W(d)=3



W(d)=2

Easier problem

Relaxing the problem into an easy one

Generating solution from Mini-Buckets

$$\mathbf{b}^* = \arg\max_b \; f(a^*, b) \cdot f(b, c^*) \cdot f(b, d^*) \cdot f(b, e^*)$$

$$\mathbf{c}^* = \arg\max_c \; f(c, a^*) \cdot f(c, e^*) \cdot \lambda_{B\to C}(a^*, c)$$

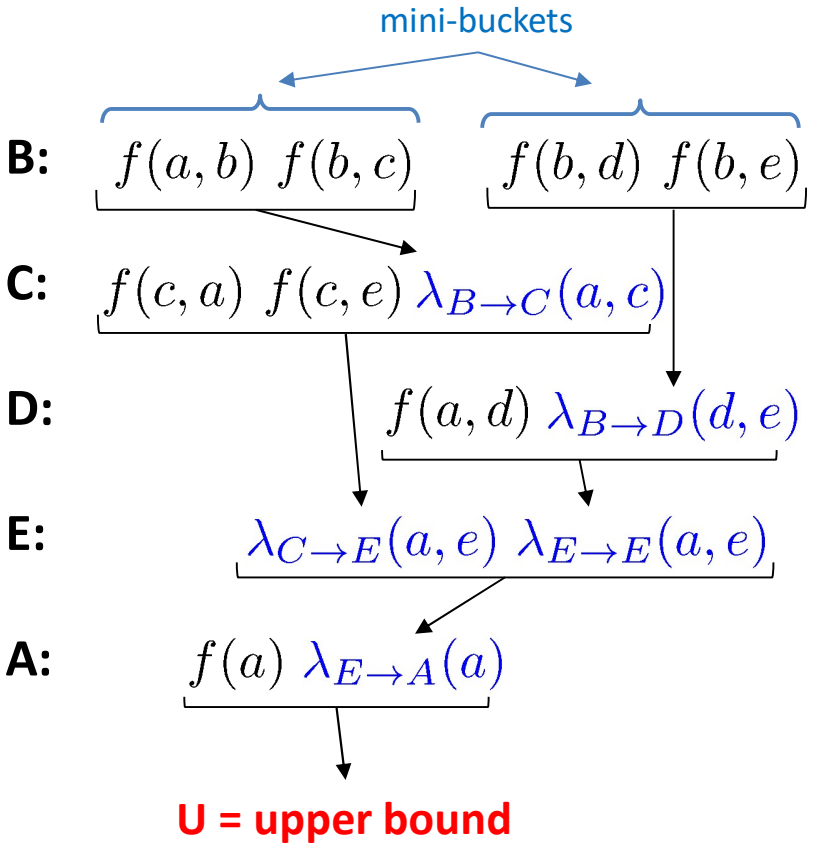
$$\mathbf{d}^* = \arg\max_d \; f(a^*, d) \cdot \lambda_{B\to D}(d, e^*)$$

$$\mathbf{e}^* = \arg\max_e \; \lambda_{C\to E}(a^*, e) \cdot \lambda_{D\to E}(a^*, e)$$

$$\mathbf{a}^* = \arg\max_a \; f(a) \cdot \lambda_{E\to A}(a)$$

Greedy configuration = lower bound

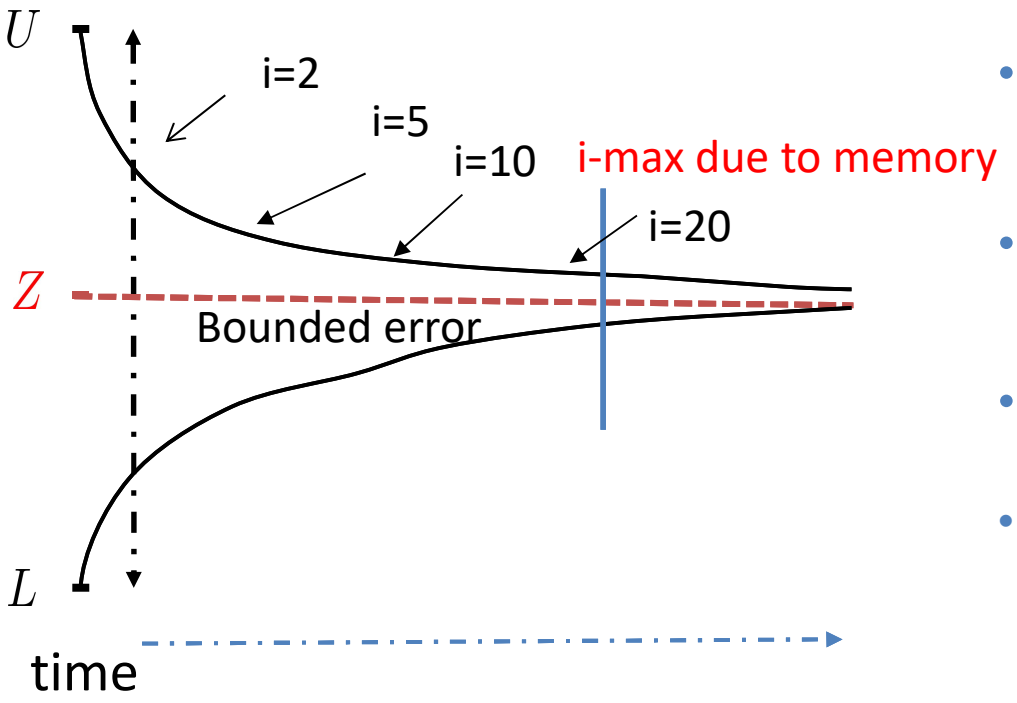
For optimization (max-product)



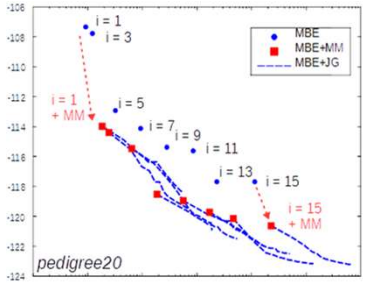
U = upper bound

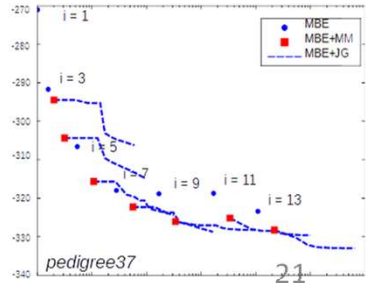
Mini-Bucket Properties

The i-bound, user control



- Yields upper or lower bounds
- **Complexity:**
  - O(r exp(i)) time and space.
- **Accuracy:** determined by Upper/Lower bound.
- **i-bound as a tuning parameter**: as i-bound increases, accuracy and complexity increase.
- i-bound selects tractable island
- **Further tightening** by variational bounds, cost-shifting methods, weighted mini-bucket [liu and Ihler, ICML, 2011],[Ihler, Flerova, Dechter & Otten ,UAI,2012]

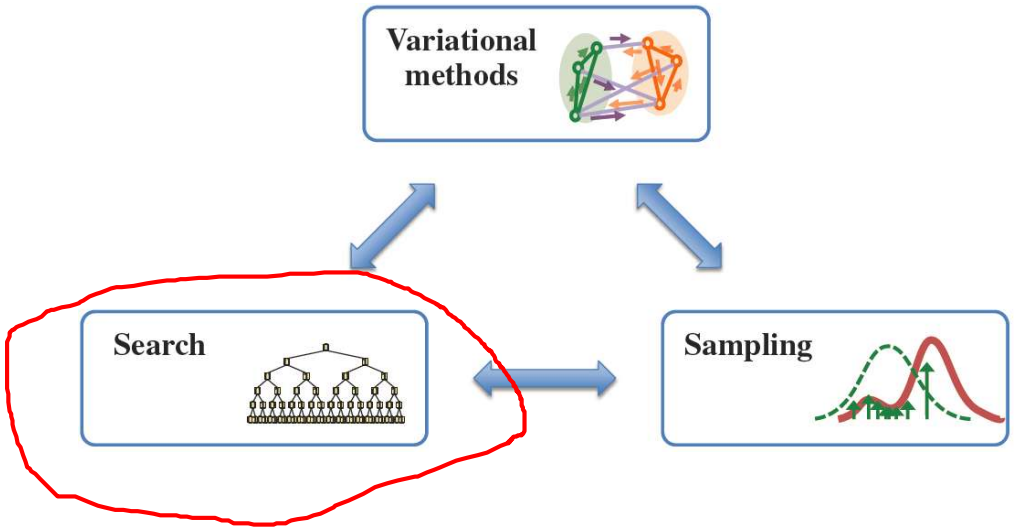




Potential use of mini-bucket:

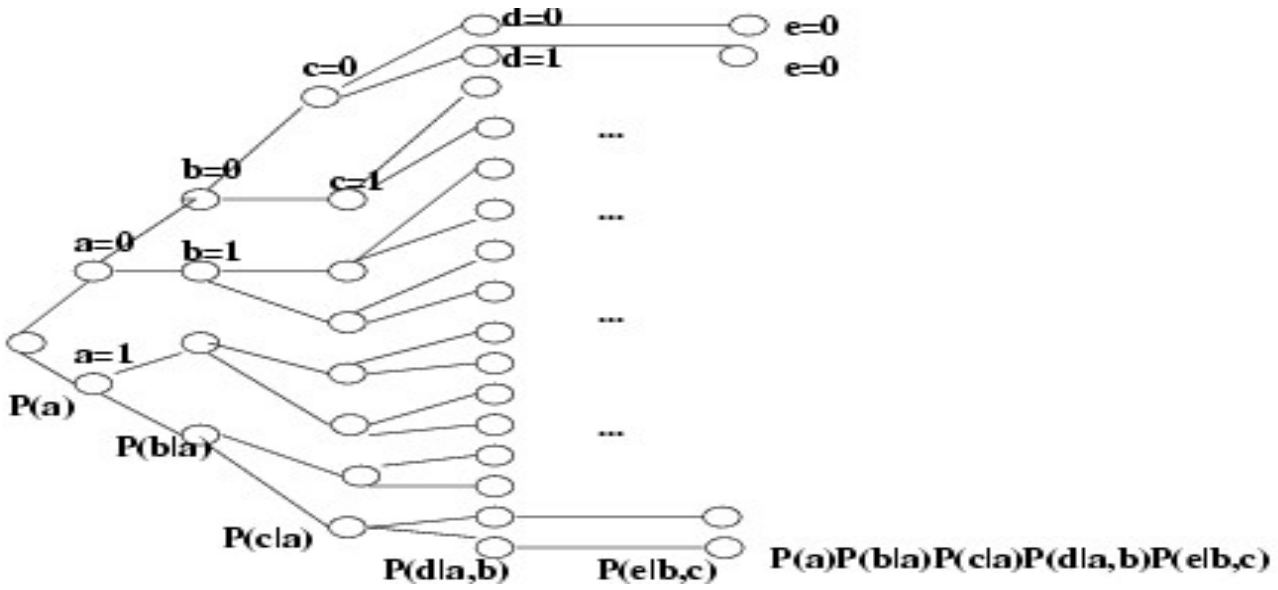
- As anytime algorithms (?)
- As heuristics in search

AND-OR Search



Traversing the Probability Tree

$$P(a, e=0) = P(a)\sum_{b} P(b \mid a)\sum_{c} P(c \mid a)\sum_{b} P(d \mid a, b)\sum_{e=0} P(e \mid b, c)$$



Brute-force Complexity: O(exp(n)), linear space
Same as counting solutions

Potential Search Spaces

[Dechter & Mateescu, AIJ 2007]

A	B	f_1
0	0	2
0	1	0
1	0	1
1	1	4

A	C	f_2
0	0	3
0	1	0
1	0	0
1	1	1

A	E	f_3
0	0	0
0	1	3
1	0	2
1	1	0

A	F	f_4
0	0	2
0	1	0
1	0	0
1	1	2

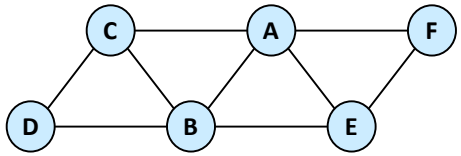
B	C	f_5
0	0	0
0	1	1
1	0	2
1	1	4

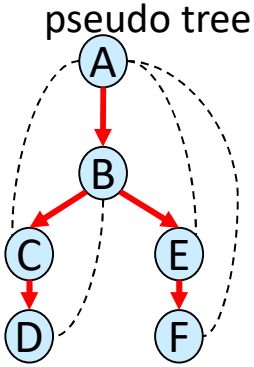
B	D	f_6
0	0	4
0	1	2
1	0	1
1	1	0

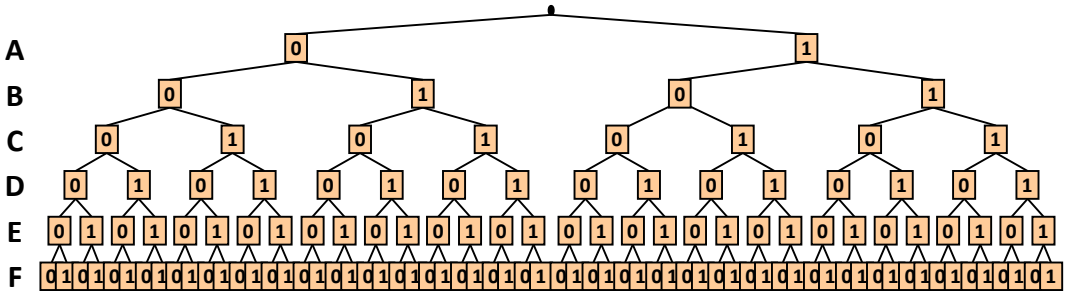
B	E	f_7
0	0	3
0	1	2
1	0	1
1	1	0

C	D	f_8
0	0	1
0	1	4
1	0	0
1	1	0

E	F	f_9
0	0	1
0	1	0
1	0	0
1	1	2



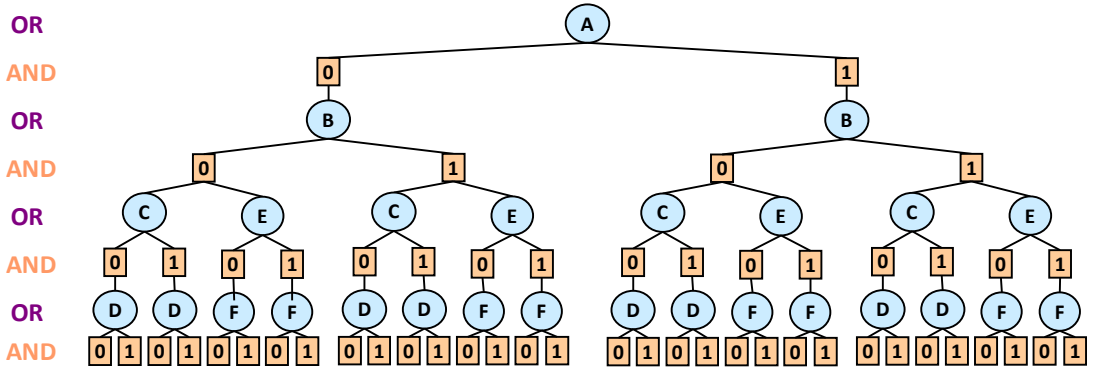




Full OR search tree

126 nodes

Size $O(k^n)$



Full AND/OR search tree

54 AND nodes

Size $O(n\ k^h)$

Still requires only linear memory

k = domain size

h = height of pseudo-tree

n = number of variables

w*= treewidth

Potential Search Spaces

A	B	f_1
0	0	2
0	1	0
1	0	1
1	1	4

A	C	f_2
0	0	3
0	1	0
1	0	0
1	1	1

A	E	f_3
0	0	0
0	1	3
1	0	2
1	1	0

A	F	f_4
0	0	2
0	1	0
1	0	0
1	1	2

B	C	f_5
0	0	0
0	1	1
1	0	2
1	1	4

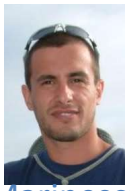
B	D	f_6
0	0	4
0	1	2
1	0	1
1	1	0

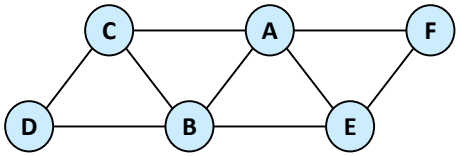
B	E	f_7
0	0	3
0	1	2
1	0	1
1	1	0

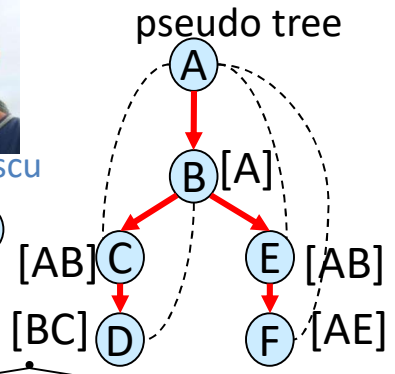
C	D	f_8
0	0	1
0	1	4
1	0	0
1	1	0

E	F	f_9
0	0	1
0	1	0
1	0	0
1	1	2

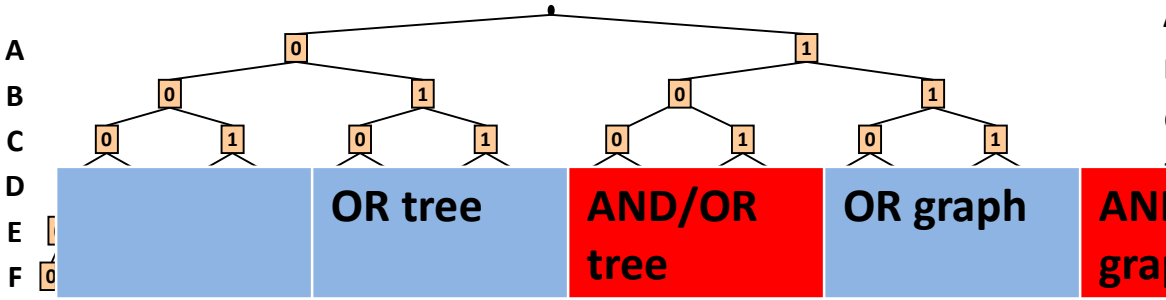

Mateescu


Marinescu

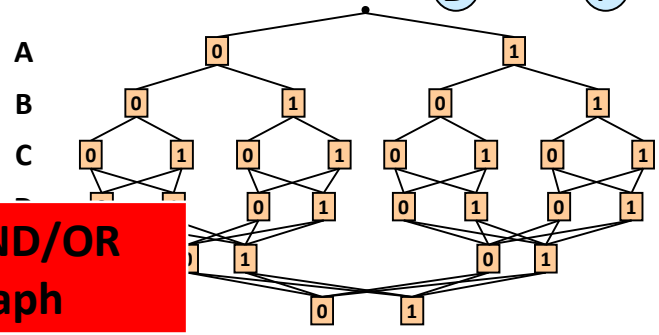




[Dechter & Mateescu, 2007]

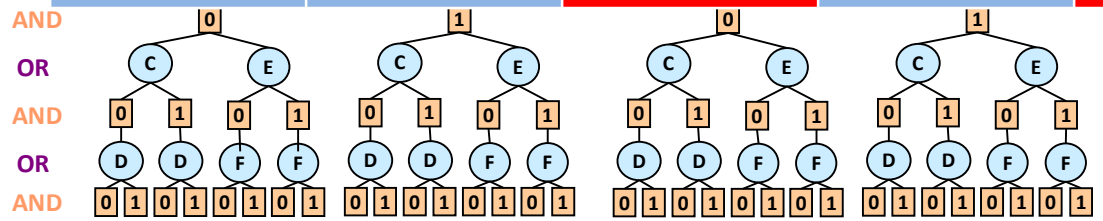


	OR tree	AND/OR tree	OR graph	AND/OR graph
Size/time	$O(k^n)$	O(nk^h)	O(n k^{pw*})	O(n k^{w*})
memory	O(n)	O(n)	O(n k^{pw*})	O(n k^{w*})



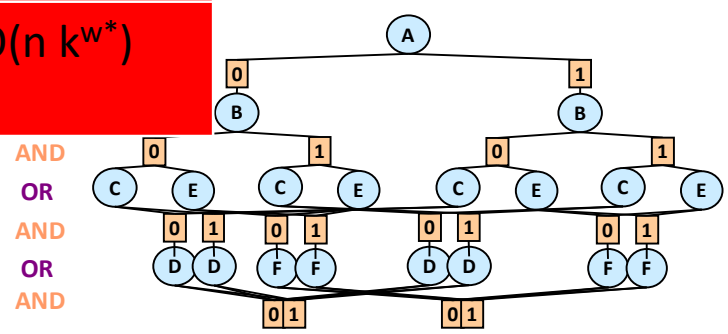
ntext minimal OR search graph

28 nodes



Full AND/OR search tree

54 AND nodes



Context minimal AND/OR search graph

18 AND nodes

Any query is best computed
Over the c-minimal AO search space

Searching AND/OR Tree (DFS or BFS)

(Probability of evidence)

$P(E \mid A,B)$

A	B	E=0	E=1
0	0	.4	.6
0	1	.5	.5
1	0	.7	.3
1	1	.2	.8

Evidence: E=0

$P(B \mid A)$

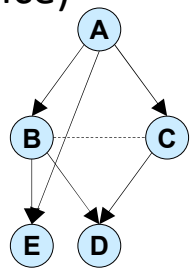
A	B=0	B=1
0	.4	.6
1	.1	.9

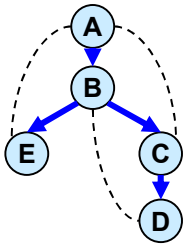
$P(C \mid A)$

A	C=0	C=1
0	.2	.8
1	.7	.3

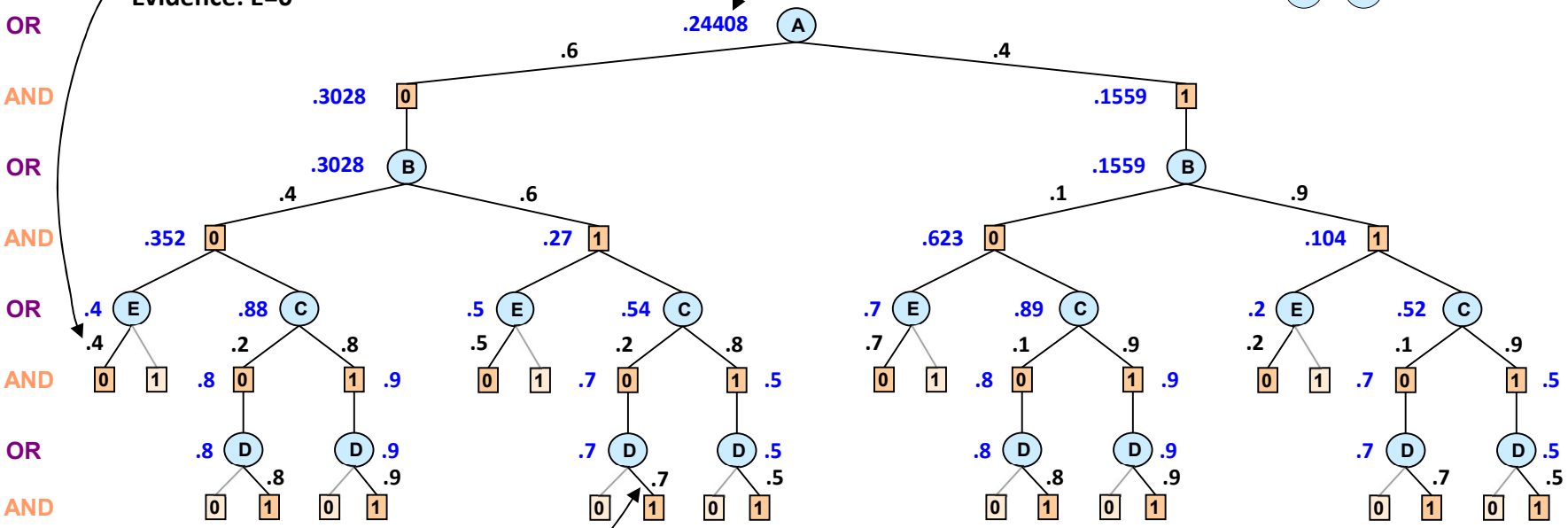
$P(A)$

A	P(A)
0	.6
1	.4





Result: P(D=1,E=0)



$P(D \mid B,C)$

B	C	D=0	D=1
0	0	.2	.8
0	1	.1	.9
1	0	.3	.7
1	1	.5	.5

Evidence: D=1

OR node: Marginalization operator (summation)

AND node: Combination operator (product)

Value of node = updated belief for sub-problem below

Searching AND/OR **Graph** (DFS or BFS)

Probability of evidence)

$P(E \mid A,B)$

A	B	E=0	E=1
0	0	.4	.6
0	1	.5	.5
1	0	.7	.3
1	1	.2	.8

Evidence: E=0

$P(B \mid A)$

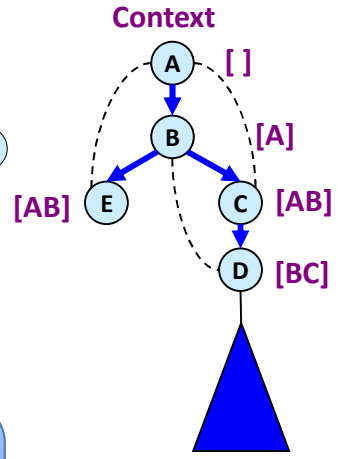
A	B=0	B=1
0	.4	.6
1	.1	.9

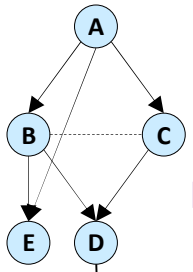
$P(C \mid A)$

A	C=0	C=1
0	.2	.8
1	.7	.3

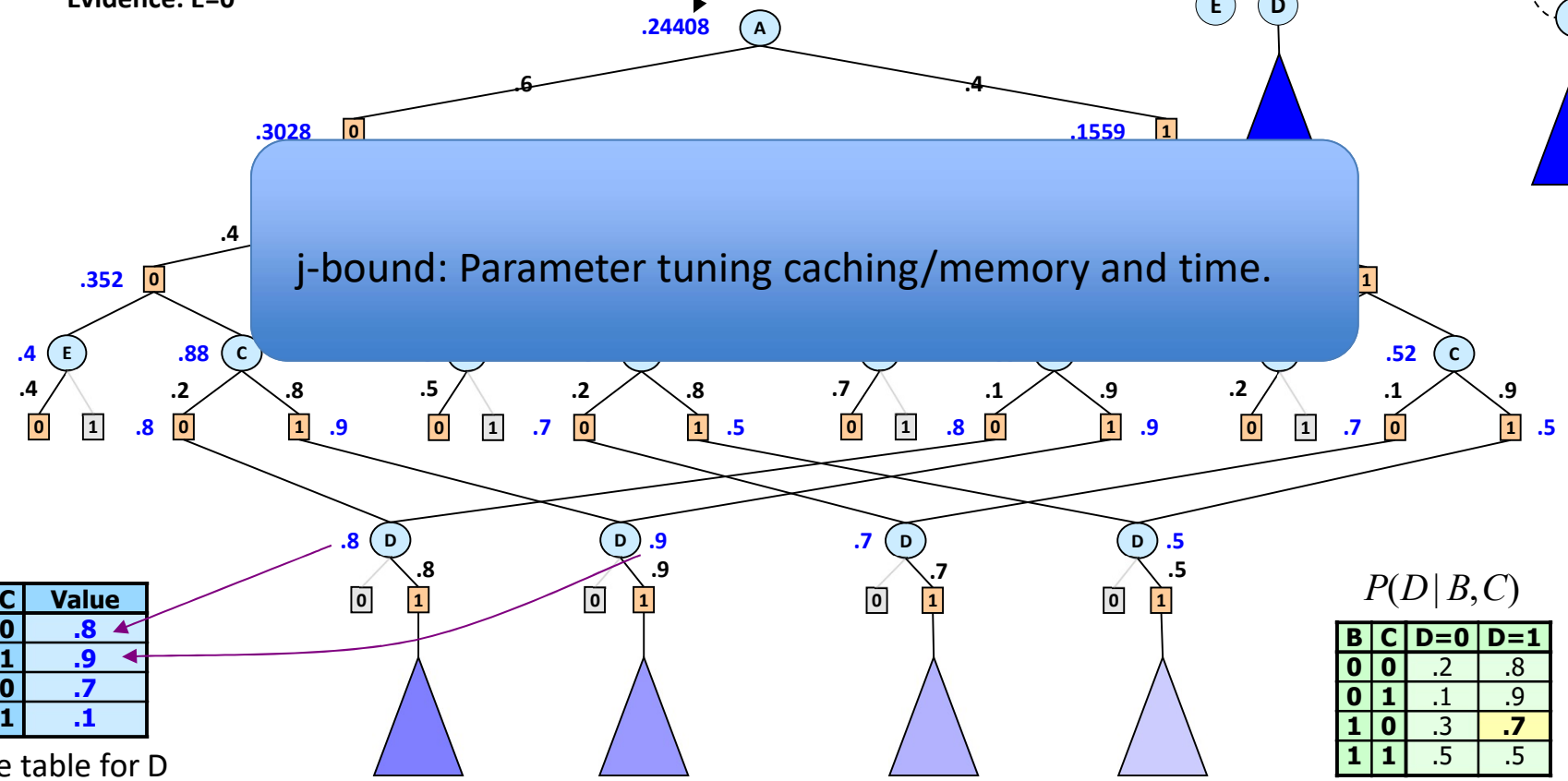
$P(A)$

A	P(A)
0	.6
1	.4





Result: P(D=1,E=0)



j-bound: Parameter tuning caching/memory and time.

B	C	Value
0	0	.8
0	1	.9
1	0	.7
1	1	.1

Cache table for D

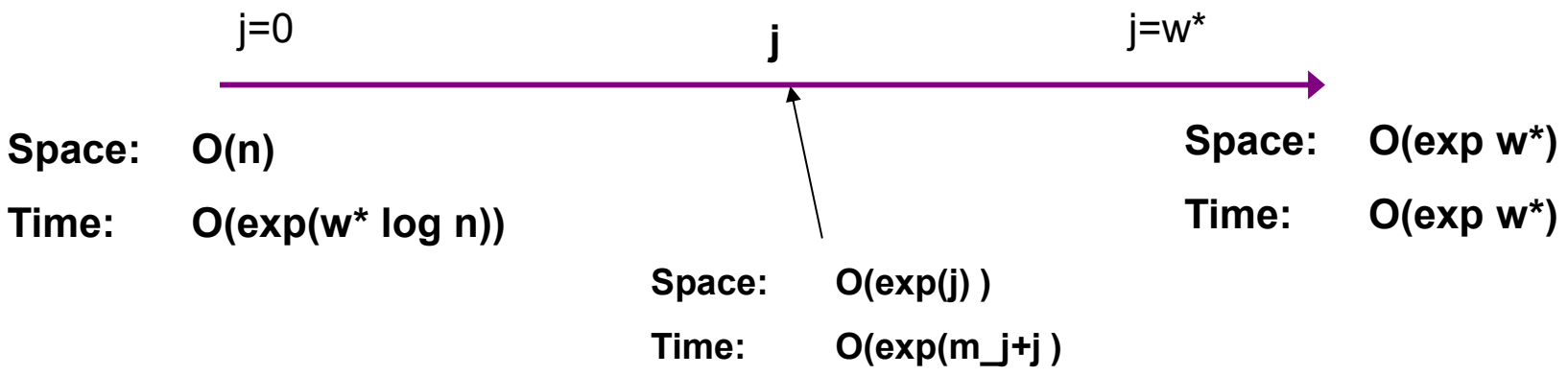
$P(D \mid B,C)$

B	C	D=0	D=1
0	0	.2	.8
0	1	.1	.9
1	0	.3	.7
1	1	.5	.5

Evidence: D=1

Searching AND/OR Graphs

- AO(j): searches depth-first, cache j-context
  - j = the max size of a cache table (i.e. number of variables in a context)



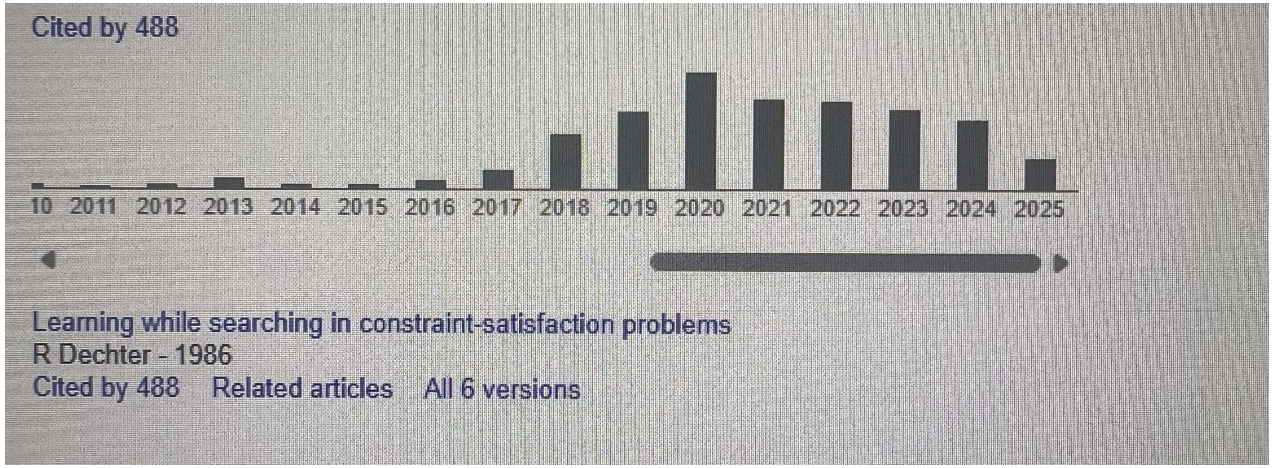
J-bound tunes time vs memory, smaller search spaces require more memory and less time.

Rina Dechter —Deep Learning Pioneer

[Florencia, Grattarola, Medium, 2018] ☺

- [Rina Dechter "Learning while searching in constraint-satisfaction problems", AAAI 1986.]

- **Deep learning** as a Conflict-Directed Learning. (CDL)
- The idea behind CDCL (Conflict directed Clause Learning) for Boolean SAT solving



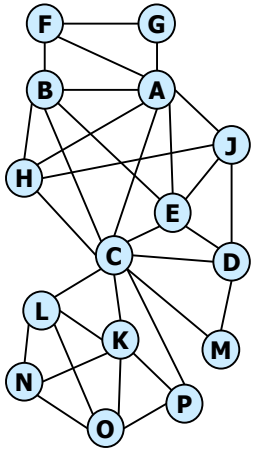
Meiri 

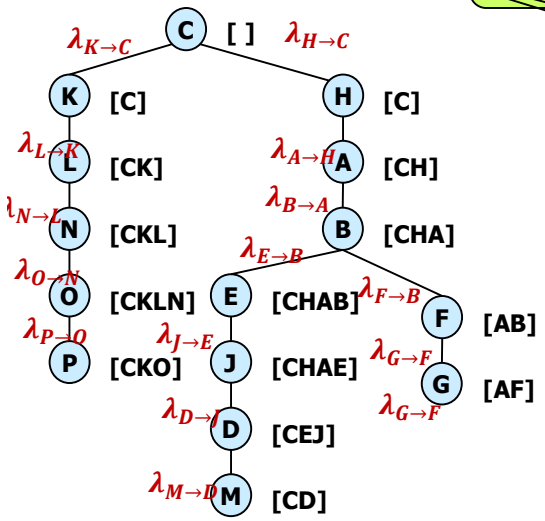
Ben-Eliyahu 

Frost 

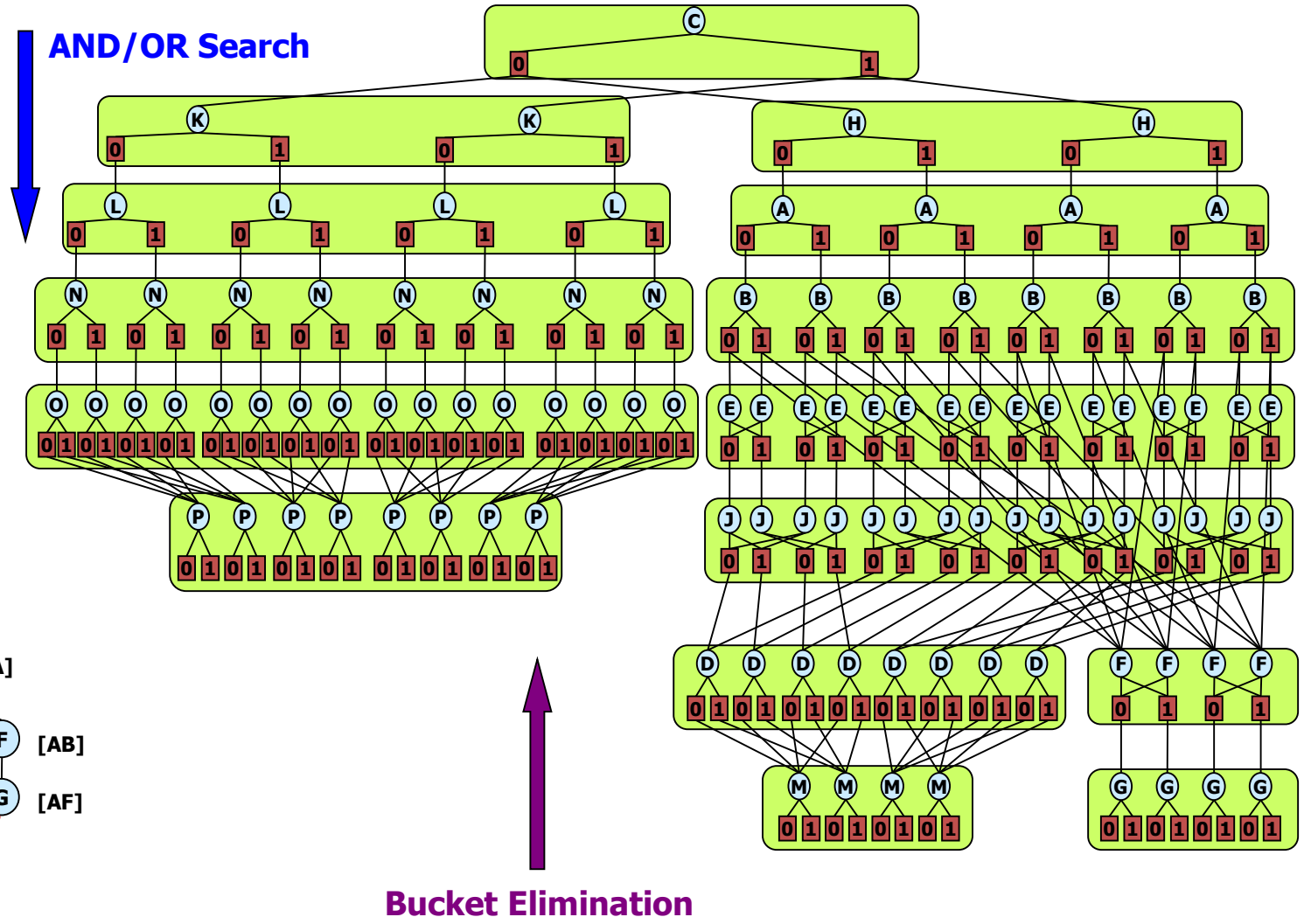
Schwalb 

AND/OR Search and Bucket Elimination

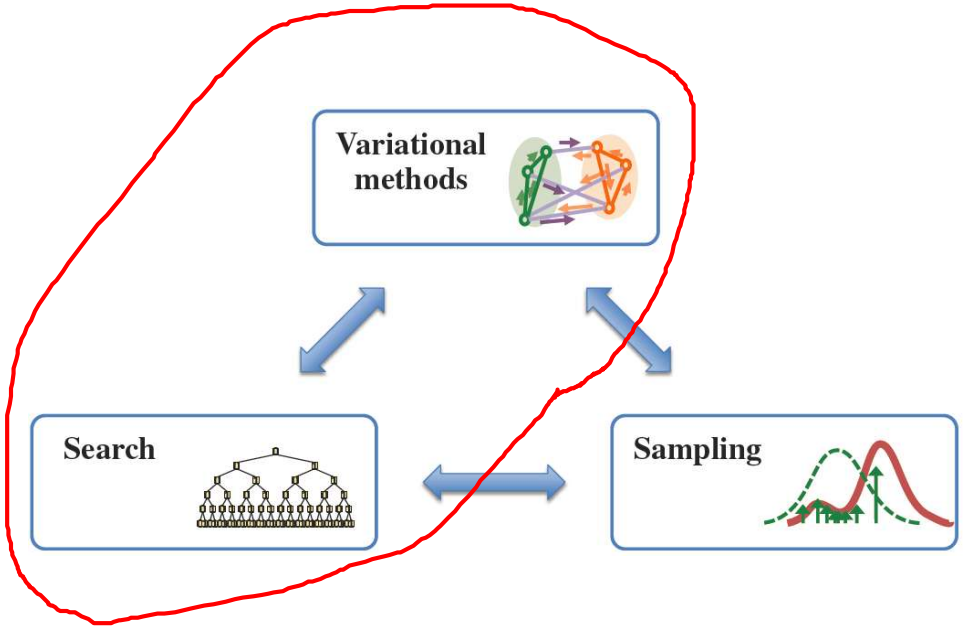




(C K H A B E J L N O D P M F G)



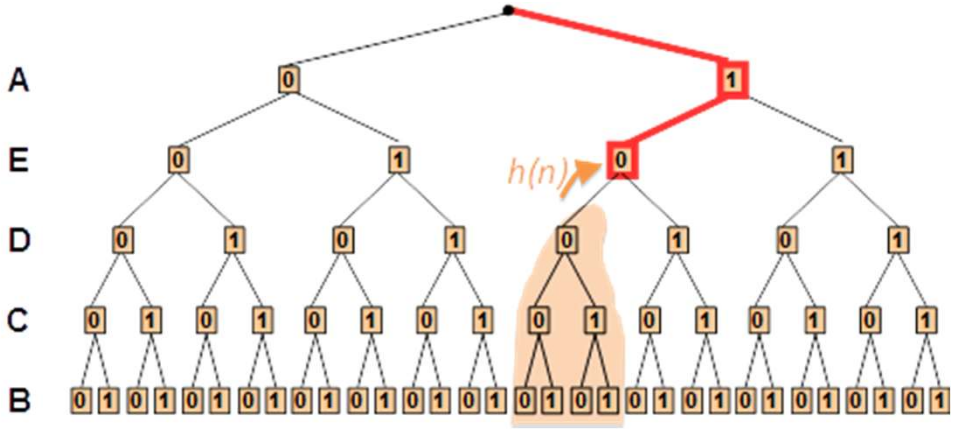
AND-OR Search **Heuristic** With Mini-Bucket

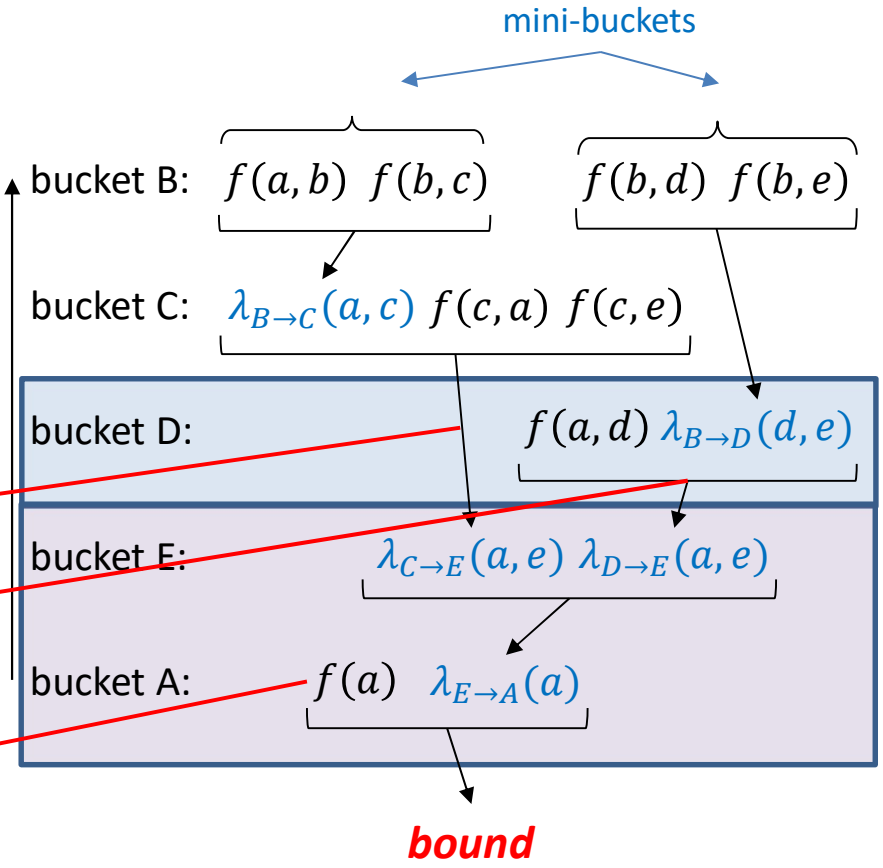


Search Guided by Mini-Bucket Heuristics

[Kask, Dechter, AIJ 2001]

Given a partial assignment, $[\hat{a} = 1, \hat{e} = 0]$
(weighted) mini-bucket gives an admissible heuristic:





"cost to go":

$$\tilde{h}(\hat{a},\hat{e},D) = \lambda_{C\to E}(\hat{a},\hat{e}) + f(\hat{a},D) + \lambda_{B\to D}(D,\hat{e})$$

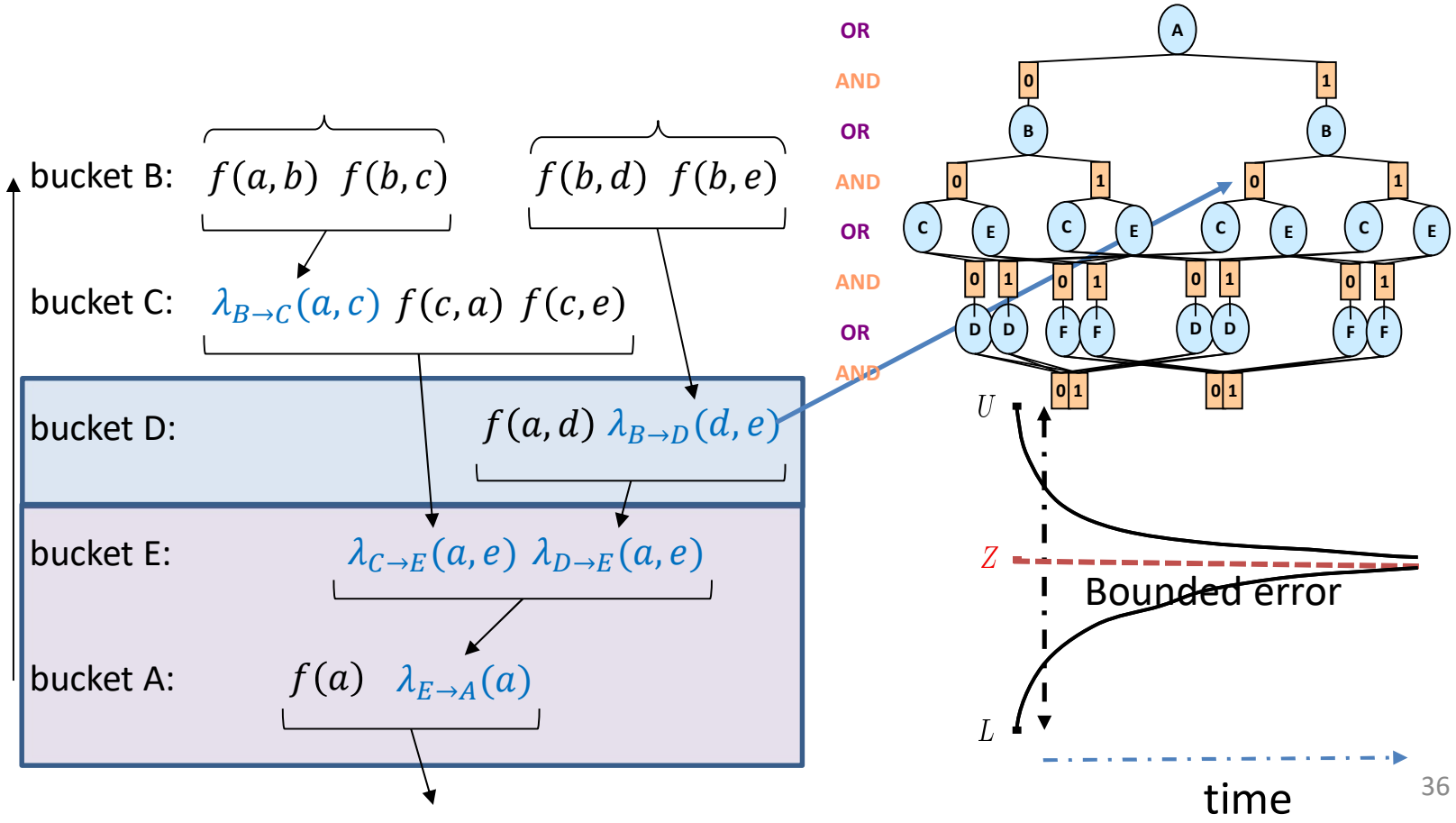
"cost so far":

$$g(\hat{a},\hat{e},D) = f(A=\hat{a})$$

For MAP, marginal map and partition function

Search Guided by Mini-Bucket Heuristics

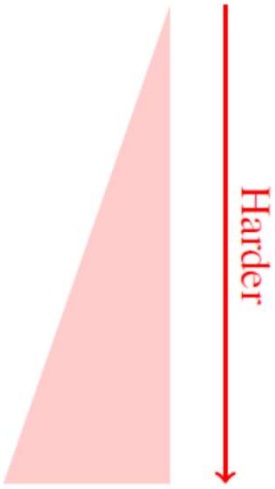
- Desiderata, meaningful confidence interval, responsive, complete
- Winning frameworks: search, (or sampling) guided by heuristics generated via tractable islands.



Search Guided by Mini-Bucket Heuristics

- 2007-2020 we developed Advance AND/OR Heuristic search for
  - Map & m-best
  - Marginal Map,
  - Partition function
  - MEU.
- Algorithms tuned by **i-bound** (mini-bucket), **j-bound** (search space) with caching,
- Illustrated anytime performance

Max-Inference:	$f(x^*) = \max_x \prod_\alpha f_\alpha(x_\alpha)$
Sum-Inference:	$Z = \sum_x \prod_\alpha f_\alpha(x_\alpha)$
Mixed-Inference (MMAP):	$f_M(x_M^*) = \max_{x_M} \sum_{x_S} \prod_\alpha f_\alpha(x_\alpha)$
Mixed-Inference (MEU):	$\text{MEU} = \max_{D_1,\ldots,D_m} \sum_{X_1,\ldots X_n} (\prod_{P_i \in P} P_i) \times (\sum_{r_i \in R} r_i)$



Anytime Bounding of Marginal MAP

[UAI'14, IJCAI'15, AAAI'16, AAAI'17, (Marinescu, Lee, Ihler, Dechter)]


Marinescu


Lee


Ihler

Otten

[Otten & Dechter, *AI Communications, 2012*.]

[Flerova, Marinescu & Dechter, m best, JAIR, 2015,

"Weighted heuristic search" *Math&AI, 2016*

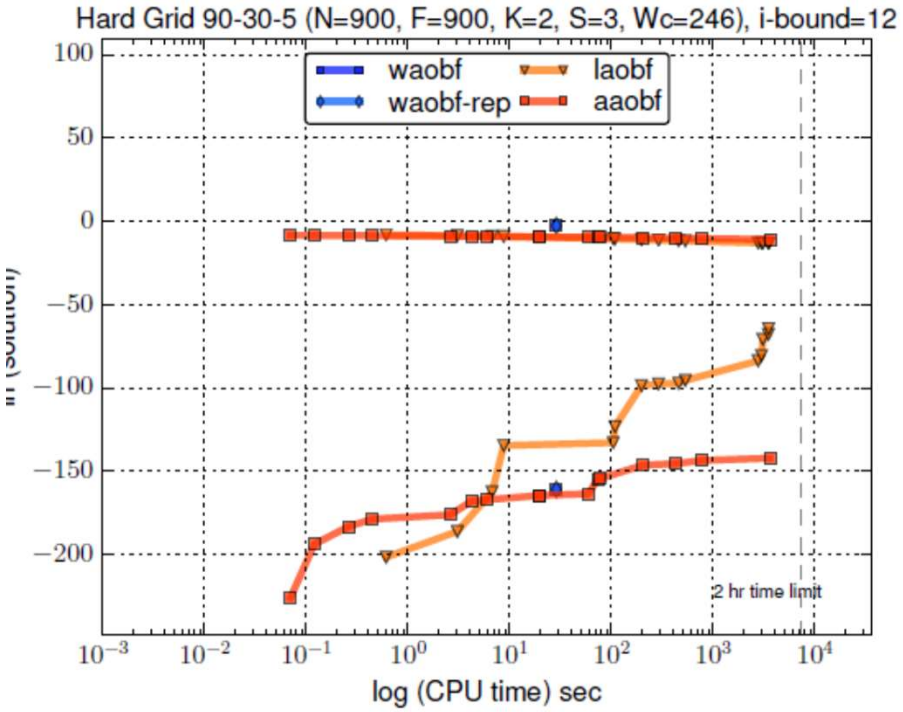
Flerova

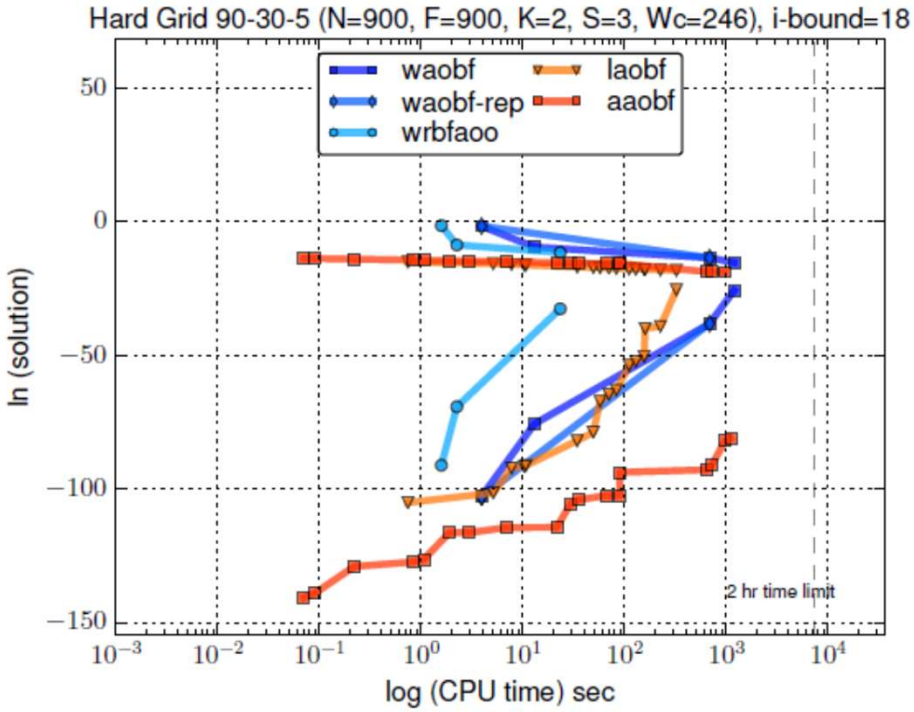
[Lam,Kask,Larrosa &Dechter. Residual-Guided Look-Ahead, JAIR 2015]

Lam

i-bound = 12

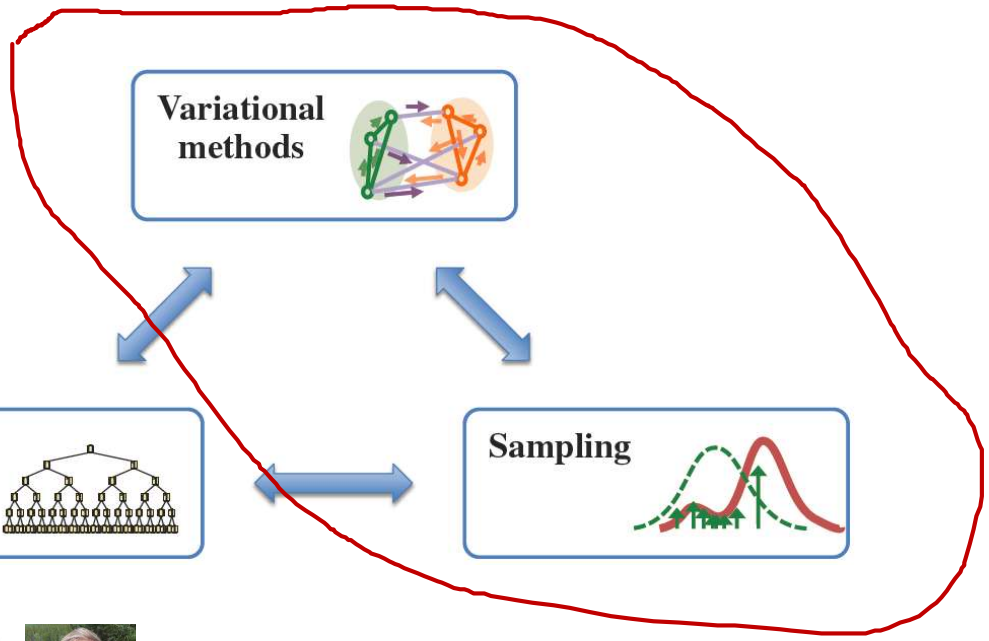
i-bound = 18

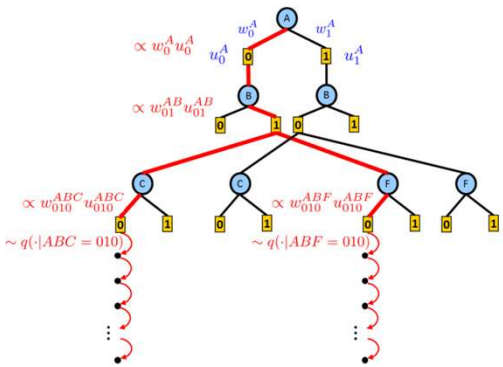




Sampling

$$Z = \sum_{\mathbf{x}} \prod_{\alpha} f_\alpha(\mathbf{x}_\alpha)$$





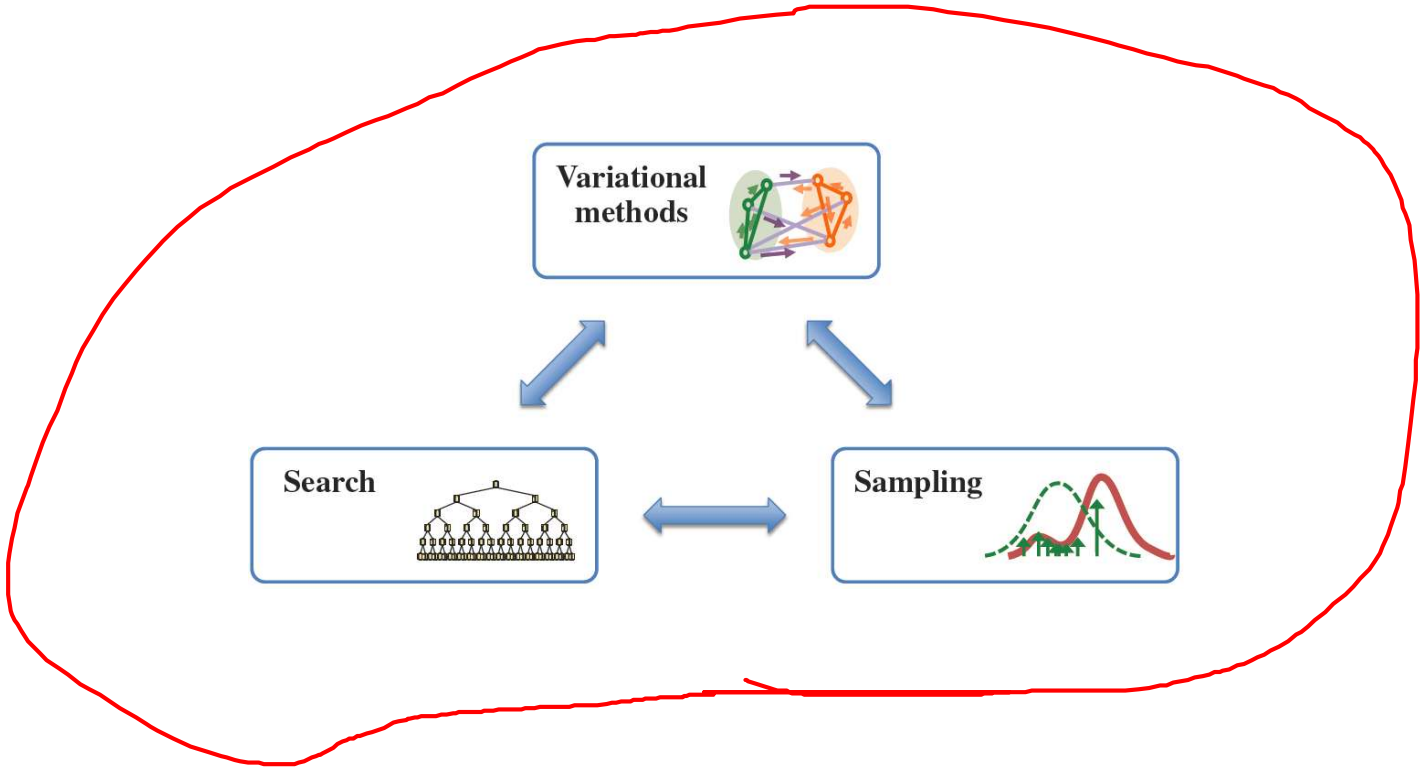

Gogate Bidyuk

[Bidyuk &Dechter, Cutset Sampling, JAIR 2006]
[Gogate & Dechter, SampleSearch, AIJ 2007]
[Gogate & Dechter, AND/OR Sampling, 2012]


Ihler Lou

[Lou, Dechter, Ihler, Dynamic IS, NeurIps, 2017]
- Anytime AnySpace, AAAI, 2017
- AAAI 2018, UAI 2018,

AND-OR Abstraction Sampling



From Sampling to Searching

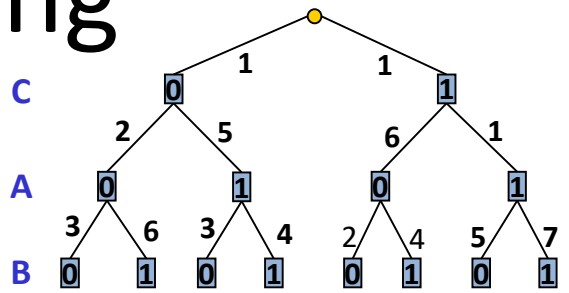
[Filjor, Pezeshki, Kask, Ihler &Dechter, UAI 2018, AAAI 2020, UAI 2024]


Kask


Filjor


Pezeshki

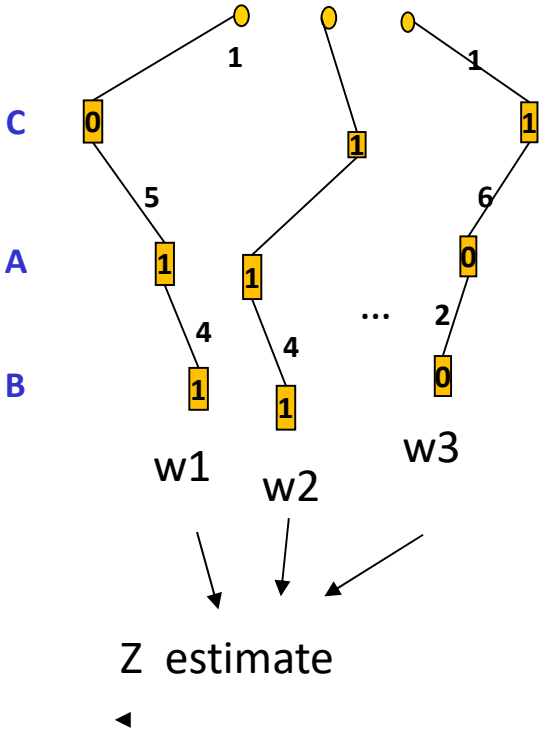

Ihler

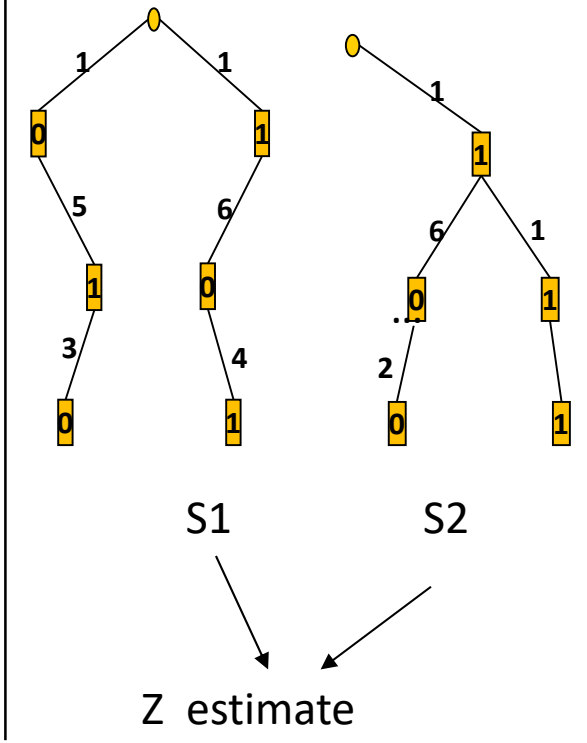


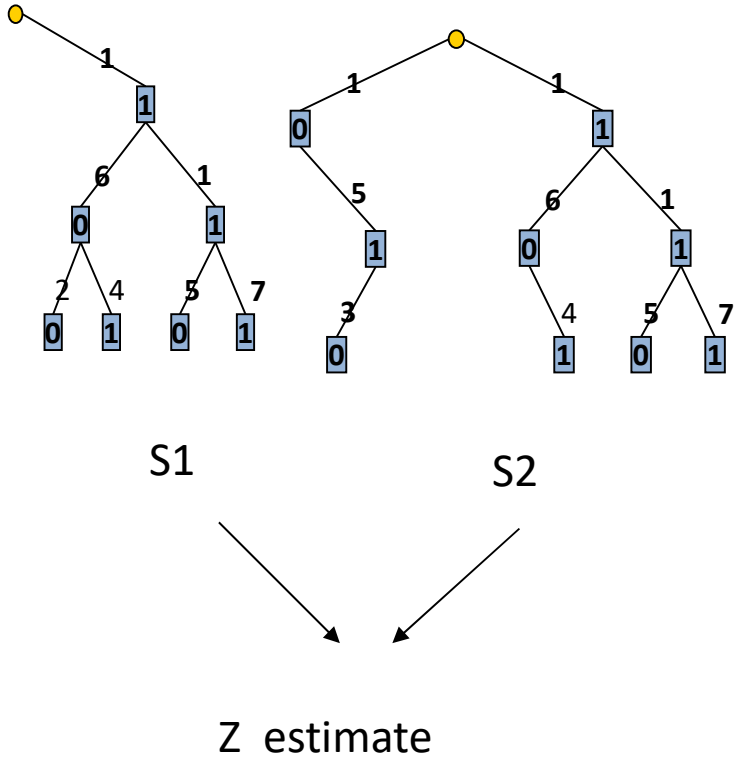
4-config-subtree sampling

2-config-subtree sampling

Importance sampling







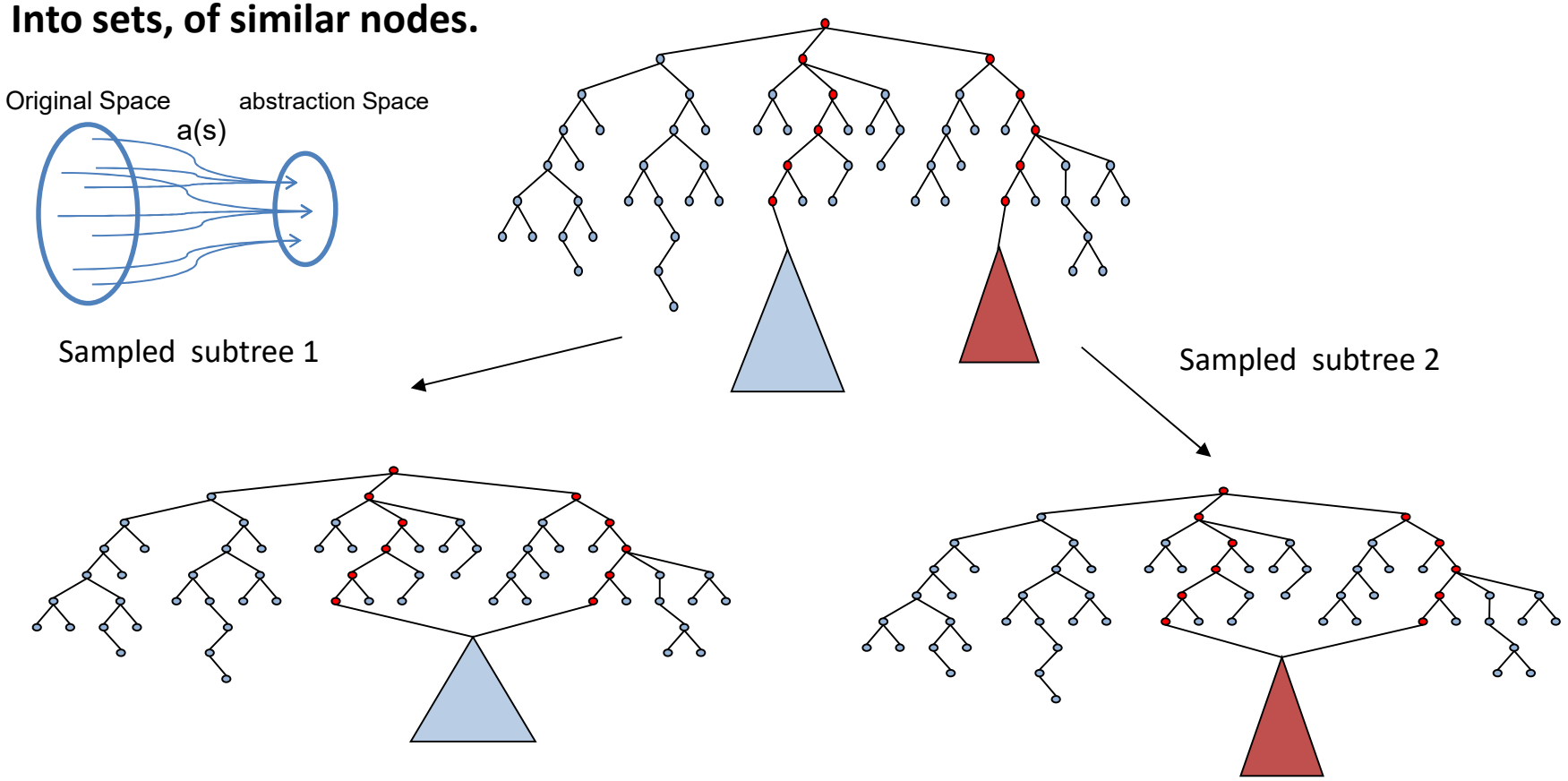
More searching less sampling

From Searching to Sampling

similar

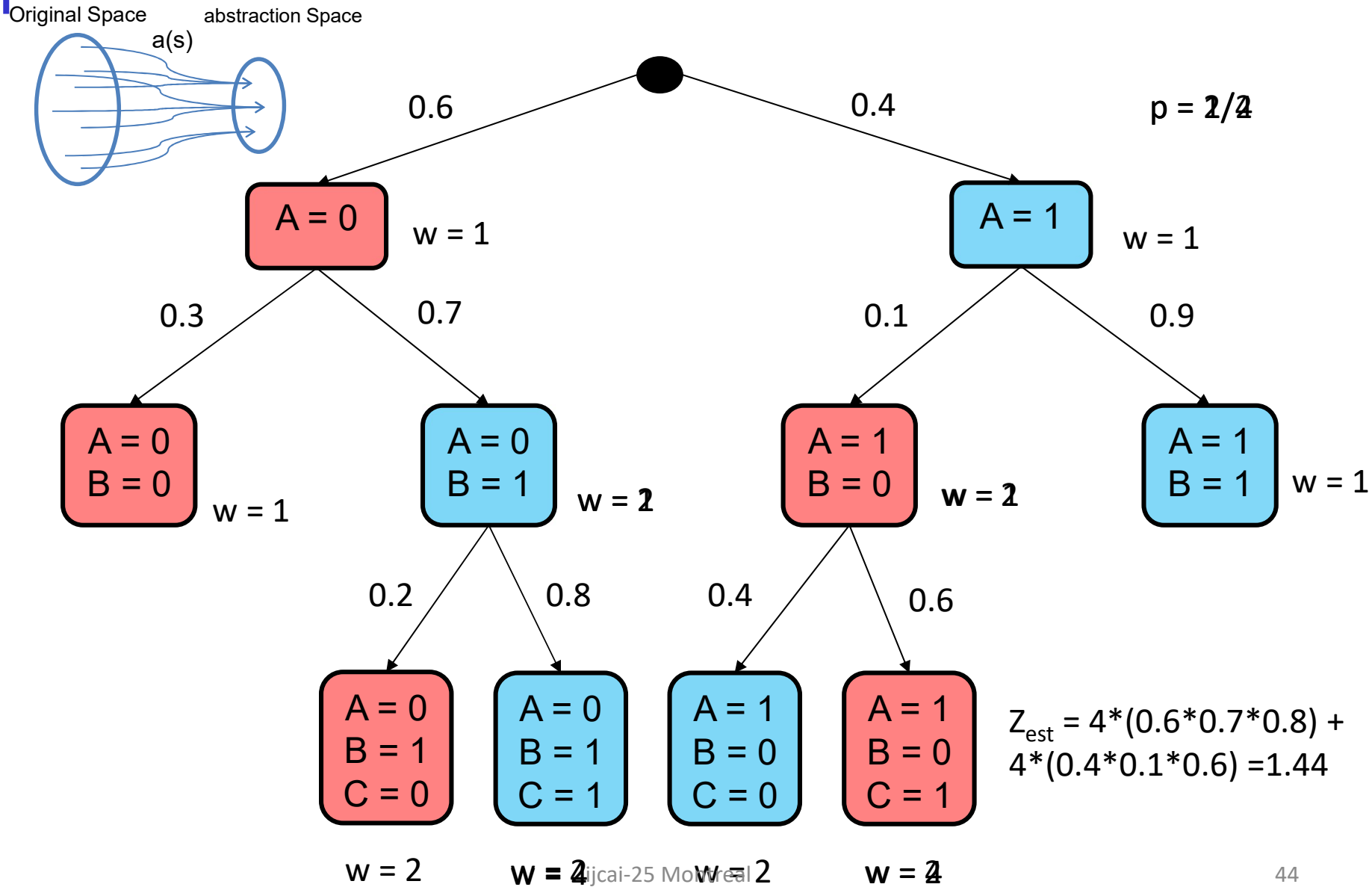
- Merge nodes that root ~~identical~~ subtrees

Abstraction: a partition of nodes
Into sets, of similar nodes.



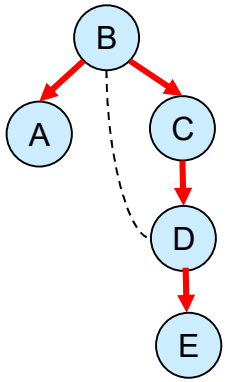
Search: as a license to merge nodes that root similar subtrees, compact representation
Statistics: Stratified sampling. Reducing variance

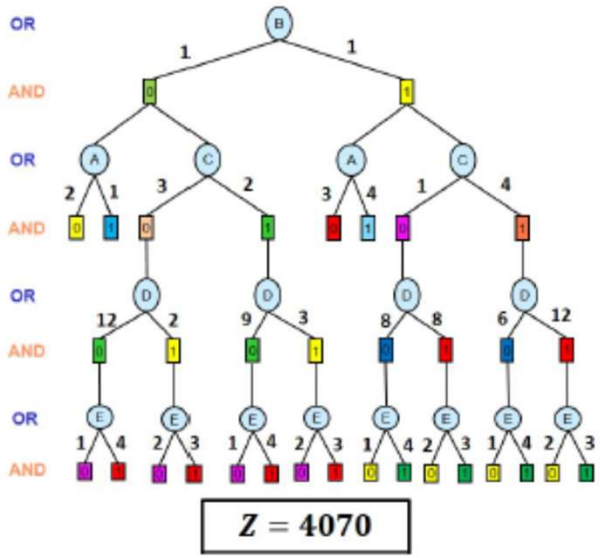
Example, Abstraction Sampling



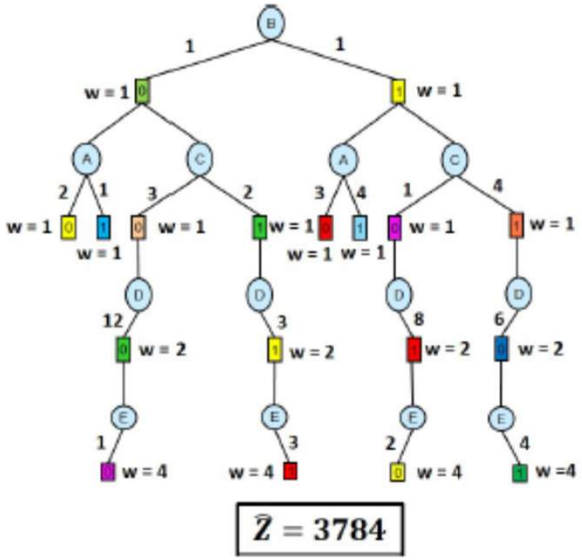
Abstraction Sampling - AND/OR

- Input: Abstraction function a, a sampling proposal p.
- Traverses AND/OR search tree breadth-first
- Compute estimate $\hat{Z}$





(d) Full AND/OR search tree



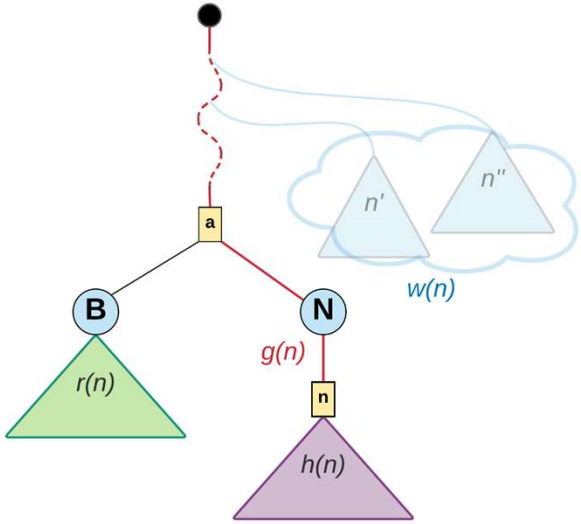
(e) Probe - AND/OR

Proposal and Abstraction Functions

- Proposal distribution

$$p(n) \leftarrow \frac{w(n)\cdot g(n)\cdot h(n)\cdot r(n)}{\sum_{m\in A_i} w(m)\cdot g(m)\cdot h(m)\cdot r(m)}$$

Mini-bucket heuristic



- Abstraction functions

Context-based abstractions, graph-based

Relax context, Randomized relax context

Value-based abstractions.

Guided by a real heuristic h function

simpleHB, minVarHB (proved superior among a selection)

Results

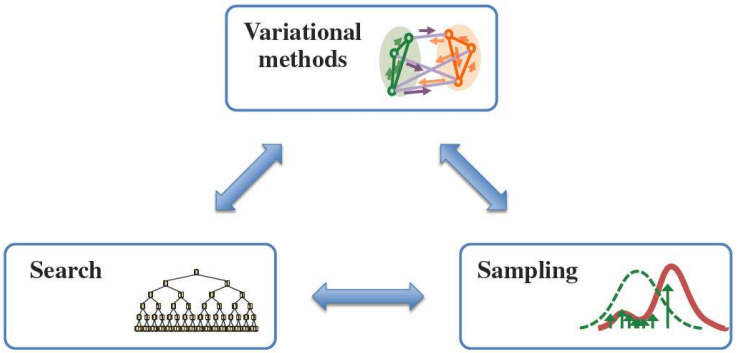
- **Theory:**
  - Abstraction Sampling is unbiased.
  - Size of a probe/sample: O(nm) (n = #variables, m = # abstractions)
- **Empirical**
  - More abstract states per layer is better.
  - Value-based abstractions are superior.
  - Random abstractions are highly competitive
  - Abstraction Sampling is highly competitive against state-of-the-art.

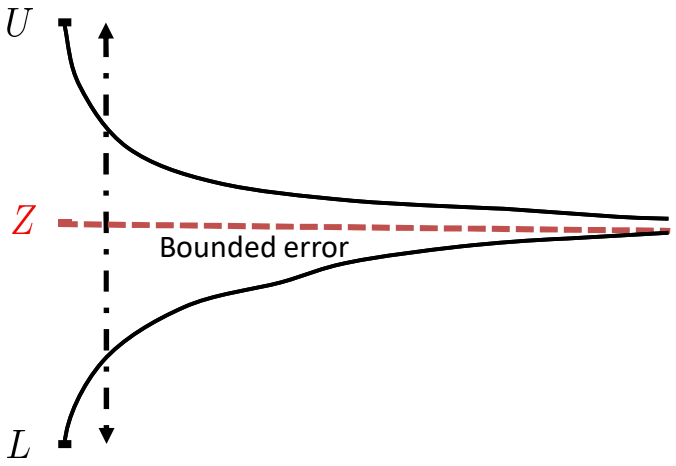
iB-5, t-300sec, Exact	DBN		Grids		Pedigree		Promedas	
Scheme	**Fail**	**Avg. Error**	**Fail**	**Avg. Error**	**Fail**	**Avg. Error**	**Fail**	**Avg. Error**
WMB-IS	0	15.69	0	10.66	20	14.21	46	9.01
IJGP-SS	0	2.31	0	2.93	0	0.30	**0**	**0.15**
DIS	0	8.76	0	7.90	6	4.37	31	4.19
AS + equalDistQB	**0**	**0.13**	**0**	**1.75**	**0**	**0.13**	**0**	**0.44**
AS + RAND	**0**	**0.10**	**0**	**1.50**	**0**	**0.14**	**0**	**0.51**

iB-10, t-1200sec, LARGE	DBN		Grids		Linkage-Type4		Promedas	
Scheme	**Fail**	**Avg. Error**	**Fail**	**Avg. Error**	**Fail**	**Avg. Error**	**Fail**	**Avg. Error**
WMB-IS	14	172.21	7	221.45	67	194.85	120	18.06
IJGP-SS	**0**	**2.20**	10	624.63	0	60.74	**0**	**1.71**
DIS	0	46.45	0	151.07	57	126.07	62	7.34
AS + equalDistQB	**0**	**1.64**	**0**	**18.87**	**16**	**30.51**	**5**	**2.48**
AS + RAND	**0**	**2.12**	**0**	**19.05**	**19**	**33.80**	**10**	**3.94**

Conclusion and …

- Publications: http://www.ics.uci.edu/~dechter/publications.html
- For software see: https://ics.uci.edu/~dechter/software.html





- Probabilistic Programming, Neurosymbolic AI
- Integrating LLMs with automated reasoning.
- Moving to Causal Inference

Publications and Software

https://ics.uci.edu/~dechter/publications.html

https://ics.uci.edu/~dechter/software.html

PUBLICATIONS

2025

[R281] Abstract | PDF | Poster
Rina Dechter, Anna K. Raichev, Jin Tian, and Alexan
Statistics (AISTATS) 2025.

2024

[R280] Abstract | PDF | Slides
Bobak Pezeshki. "Advancing AND/OR Abstraction Sam

[R279] Abstract | PDF |
Anna K. Raichev, Jin Tian, Alexander Ihler, and Rina
Workshop version, presented as part of the 9th Causal Inf

[R278] Abstract | PDF | Poster | Supplemental | Exten
Bobak Pezeshki, Kalev Kask, Alexander Ihler, Rina D
2024).

[R277] PDF | Poster | BibTex
Vincent Hsiao, Dana S. Nau, Rina Dechter. "Surrogate

⋮

[R0] Abstract | PDF
Dechter, R., and Pearl, J., "Generalized Best-First Searc

[M0] PDF (English) | PDF (Hebrew - Full)
Dechter, R., Gillis, J., and Weiss, P., "Moments of the G

SOFTWARE

- **GpuBE** : This code implements an inference-based a algorithms for discrete optimization (e.g., WCSPs). Exact and Approximate Inference for (Distributed) D

- **Merlin** : Merlin is an extensible C++ library that directed and undirected models (e.g., Bayesian netw language processing to name a few. Merlin supports marginals (MAP), as well as MAP (also known as m

- **Daoopt** : DAOOPT: Distributed AND/OR Optimiz Branch-and-Bound enhancement for combinatorial o Otten, University of California, Irvine.

  Also see **Daoopt-Exp** for a branch of the above th AND/OR Best-First Search with heuristics for subpr

- **Abstraction Sampling** : Abstraction Sampling is a function. It works with the aid of an abstraction fu Monte Carlo estimate. The code base and executabl Kalev Kask (modified by Bobak Pezeshki), Universi

- **AOMDD**: AOMDD: AND/OR Multivalued Decisi contains an implementation of bucket elimination us

⋮

Thank You !



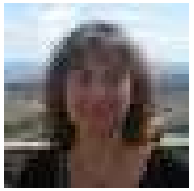






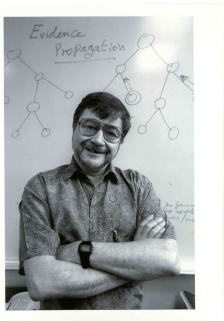


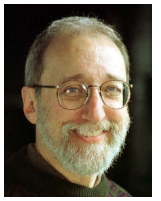
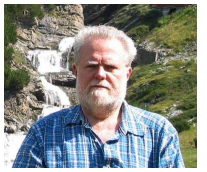


Alex Ihler
Kalev Kask
Itay Meiri
Rachel Ben-Eliyahu
Eddie Schwalb
Dan Frost
Irina Rish
Bozhena Bidyuk
Robert Mateescu
Radu Marinescu
Vibhav Gogate
Emma Rollon
Lars Otten
Natalia Flerova
Andrew Gelfand
William Lam
Junkyu Lee
Qi Lou
Vincent Hsiao
Bobak Pezeshki
Javier Larrosa

Judea Pearl



Ugo Montanari
Alan Mackworth
Eugene Freuder



My family: (Avi Gadi, Dani, Eyal)