A New Bounding Scheme for Influence Diagrams

Radu Marinescu, Junkyu Lee and Rina Dechter

AAAI-2021

Motivation

- Influence diagrams are a powerful formalism for reasoning with sequential decision-making problems under uncertainties
  - Involve random (or chance) variables, decision variables and utility functions
- Task: find the maximum expected utility (MEU) and the corresponding optimal policy
  - Notoriously difficult to solve exactly in practice
- Recent work focused on bounding the MEU
  - E.g., information relaxation, reformulation to Marginal MAP, partitioning over join-trees
- Contribution:
  - Revisit multi-operator cluster DAG (MCDAG) decompositions for influence diagrams
  - Partitioning-based (mini-bucket) approximation for MCDAGs to upper bound the MEU
  - Apply cost-shifting to tighten the upper bounds further
  - Show empirically that the new scheme produces bounds that are several orders of magnitude tighter than those obtained with existing bounding schemes

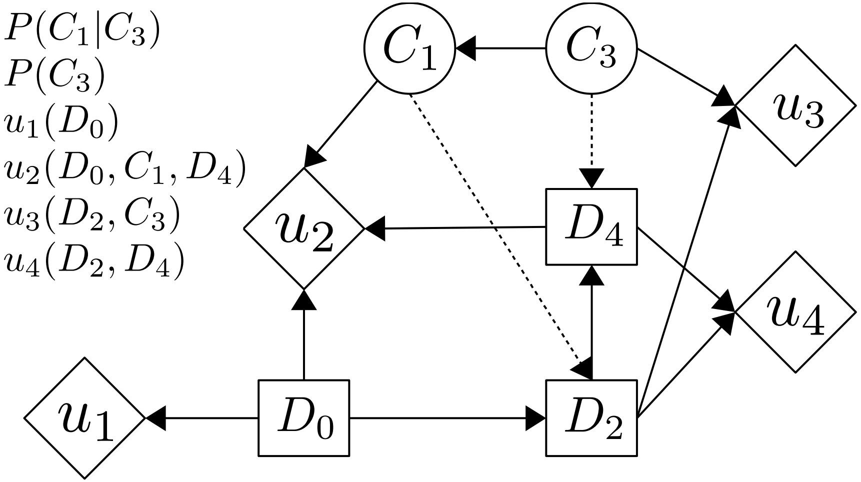
Outline

- Motivation
- Preliminaries
- MCDAG decompositions
- Weighted mini-buckets over MCDAGs
- Experimental results
- Conclusion

Influence Diagrams

- An ID is a tuple (X, D, P, U) where:
  - $X = \{X_1, \dots, X_n\}$ are chance variables
  - $D = \{D_1, \dots, D_m\}$ are decision variables
  - $P = \{P_1, \dots, P_n\}, s.t. P_i = \Pr(X_i | pa(X_i))$ are conditional probability tables (CPTs)
  - $U = \{U_1, \dots, U_r\}$ are local utility functions defining global utility $\mathcal{U} = \sum_{i=1}^{r} U_i$
- No-forgetness and regularity imply a partial ordering: $I_0 \prec D_1 \prec I_1 \prec \cdots \prec D_m \prec I_m$
- MEU: $\sum_{I_0} \max_{D_1} \dots \sum_{I_m} \max_{D_m} \sum_{I_m} (\prod P_i \sum U_j)$
- Variable elimination [Schachter, 1986], [Jensen et al., 1994], [Dechter, 2000] ...

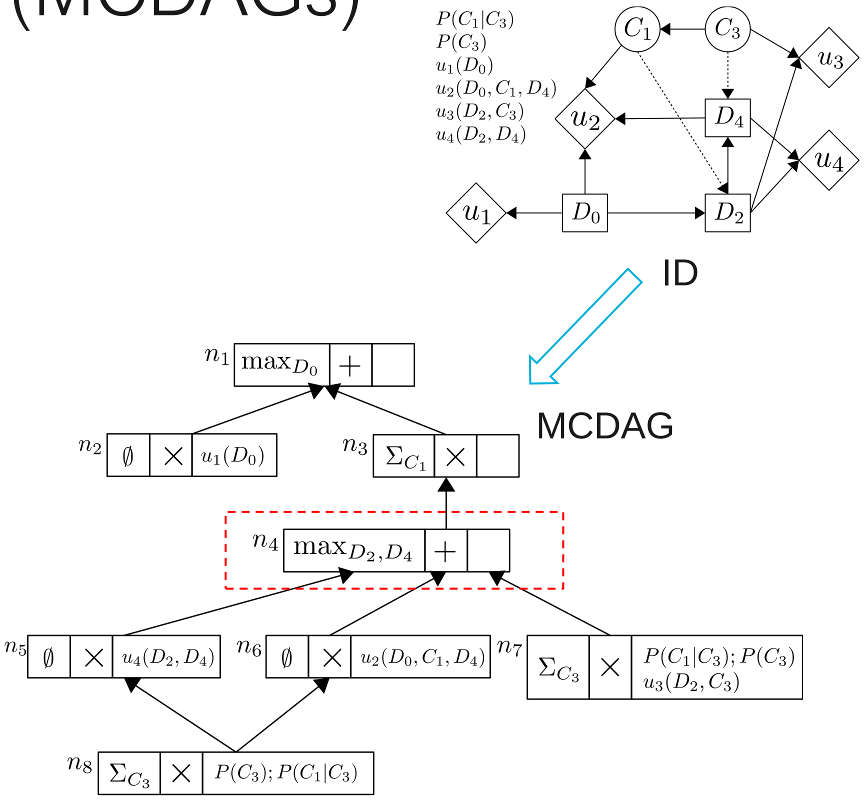
$P(C_1|C_3)$
$P(C_3)$
$u_1(D_0)$
$u_2(D_0, C_1, D_4)$
$u_3(D_2, C_3)$
$u_4(D_2, D_4)$



$$\max_{D_0} \sum_{C_1} \max_{D_2} \sum_{C_3} \max_{D_4} P(C_1|C_3)P(C_3)(u_1 + u_2 + u_3 + u_4)$$

Multi-operator Cluster DAGs (MCDAGs)

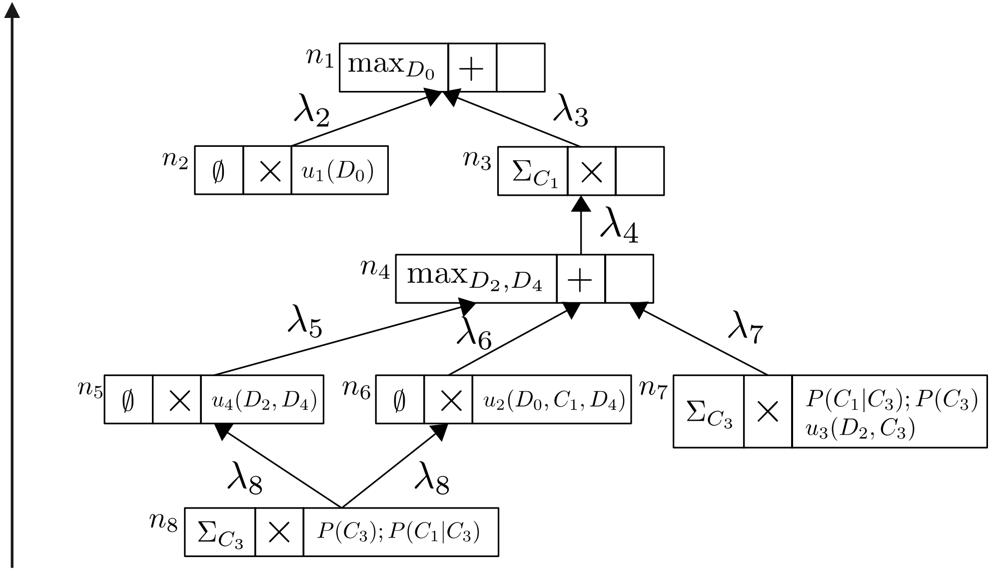
- Recent decomposition for IDs with smaller induced widths than traditional strong join-tree decompositions [Pralet et al., 2006]
- Refines the MEU expression to exploit reordering freedom and normalization conditions on CPTs
- A DAG where each vertex (cluster) c has:
  - Variables $V(c)$, functions $\Psi(c)$
  - Child clusters $ch(c)$
  - Operators $\oplus \in \{\Sigma, max\}$ and $\otimes \in \{+, \times\}$ such that $(\oplus, \otimes, \mathbb{R})$ is commutative semiring
    - $\oplus$: elimination operator
    - $\otimes$: combination operator



$V(n_4) = \{D_2, D_4\}; \Psi(n_4) = \emptyset; ch(n_4) = \{n_5, n_6, n_7\}; \oplus = ma$ … $) = +$

Variable Elimination over MCDAGs

- Compute the MEU via message passing over the MCDAG, from leaves to the root:



$$MEU = \max_{D_0} \lambda_2 + \lambda_3$$

$$\lambda_2 = u_1(D_0)$$

$$\lambda_3 = \sum_{C_1} \lambda_4$$

$$\lambda_4 = \max_{D_2} \max_{D_4} (\lambda_5 + \lambda_6 + \lambda_7)$$

$$\lambda_5 = \lambda_8 \cdot u_4(D_2, D_4)$$

$$\lambda_6 = \lambda_8 \cdot u_2(D_0, C_1, D_4)$$

$$\lambda_7 = \sum_{C_3} P(C_3) \cdot P(C_1|C_3) \cdot u_3(D_2, C_3)$$

$$\lambda_8 = \sum_{C_3} P(C_3) \cdot P(C_1|C_3)$$

Weighted Mini-Buckets for MCDAGs

- Complexity of VE is time and space exponential in the size of the largest message
  - i.e., exponential in the induced width of the MCDAG
- The idea is to approximate the λ-messages by sets of smaller messages (called *compound messages*) via a partitioning-based (or mini-bucket) approximation
  - Compound messages are propagated along the edges of the MCDAG
  - Compound messages must be combined in different ways, either by multiplication or summation depending on whether the sending cluster is a sum or a max one
- Formally, we define two types of compound messages:
  - π-messages: product of functions (i.e., $\pi = \prod_i f_i$)
  - σ-messages: sum of π-messages (i.e., $\sigma = \sum_j \pi_j$)
- The approximation scheme is guaranteed to output an *upper bound* on the MEU value
- Complexity is exponential (time and space) in the i-bound that controls the mini-bucket partitioning (i.e., i-bound dictates the number of distinct variables allowed in a mini-bucket)

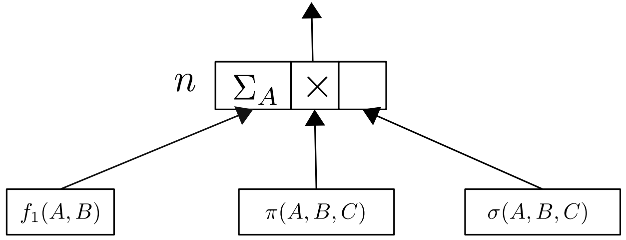
Processing a SUM Cluster

i-bound is 2, therefore we generate mini-buckets with at most 2 distinct variables

Generate a σ-message:

$\sigma = \{\pi_1, \pi_2\}$, where
$\pi_1 = \{\lambda_1(B), \lambda_2(C)\}$,
$\pi_2 = \{\lambda_3(B), \lambda_4(C)\}$

$$\sigma(B,C) = \lambda_1(B) \cdot \lambda_2(C) + \lambda_3(B) \cdot \lambda_4(C)$$



$$\pi(A,B,C) = h_1(A,B) \cdot h_2(A,C)$$
$$\sigma(A,B,C) = g_1(A,B) \cdot g_2(A,C) + g_3(A,B) \cdot g_4(A,C)$$

$$\lambda = \sum_A f_1(A,B) \cdot h_1(A,B) \cdot g_1(A,B) \cdot h_2(A,C) \cdot g_2(A,C)$$
$$+ \sum_A f_1(A,B) \cdot h_1(A,B) \cdot g_3(A,B) \cdot h_2(A,C) \cdot g_4(A,C)$$
$$\leq \left(\sum_A f_1(A,B) \cdot h_1(A,B) \cdot g_1(A,B)\right) \cdot \left(\sum_A h_2(A,C) \cdot g_2(A,C)\right)$$
$$+ \left(\sum_A f_1(A,B) \cdot h_1(A,B) \cdot g_3(A,B)\right) \cdot \left(\sum_A \cdot h_2(A,C) \cdot g_4(A,C)\right)$$
$$= \lambda_1(B) \cdot \lambda_2(C) + \lambda_3(B) \cdot \lambda_4(C)$$

mini-buckets

Processing a MAX cluster

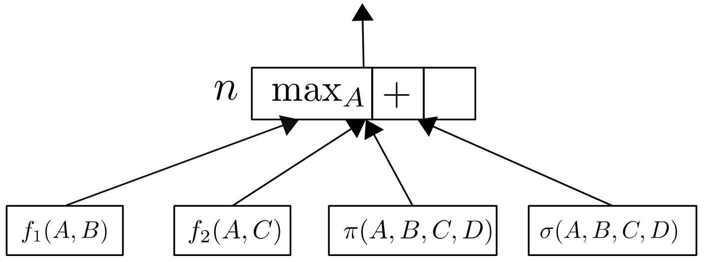
i-bound is 2, therefore we generate mini-buckets with at most 2 distinct variables

Generate a σ-message:

$\sigma = \{\pi_1, \pi_2, \pi_3, \pi_4\}$, where
$\pi_1 = \{\lambda_1^1(B, C)\}$,
$\pi_2 = \{\lambda_2^1(B), h_2(C, D)\}$,
$\pi_3 = \{\lambda_3^1(B, C)\}$,
$\pi_1 = \{g_4(C, D), \lambda_3^2(B)\}$

For MAX clusters, the max operator is pushed both inside summation as well as multiplication (unlike SUM case)

$$\sigma(B,C,D) = \lambda_1^1(B,C) + h_2(C,D)\cdot\lambda_2^1(B) + \lambda_3^1(B,C) + g_4(C,D)\cdot\lambda_3^2(B)$$



$$\pi(A,B,C,D) = h_1(A,B)\cdot h_2(C,D)$$
$$\sigma(A,B,C,D) = g_1(A,B)\cdot g_2(A,C) + g_3(A,B)\cdot g_4(C,D)$$

$$\lambda = \max_A f_1(A,B) + f_2(A,C) + h_1(A,B)\cdot h_2(C,D)$$
$$+ g_1(A,B)\cdot g_2(A,C) + g_3(A,B)\cdot g_4(C,D)$$
$$\leq \max_A (f_1(A,B) + f_2(A,C)) + \max_A h_1(A,B)\cdot h_2(C,D)$$
$$+ \max_A g_1(A,B)\cdot g_2(A,C) + \max_A g_1(A,B)\cdot g_4(C,D)$$
$$= \lambda_1^1(B,C) + \lambda_2^1(B)\cdot h_2(C,D) + \lambda_3^1(B,C) + \lambda_4^1(B)\cdot g \ldots, D))$$

mini-buckets

Tightening the Bounds by Cost-Shifting

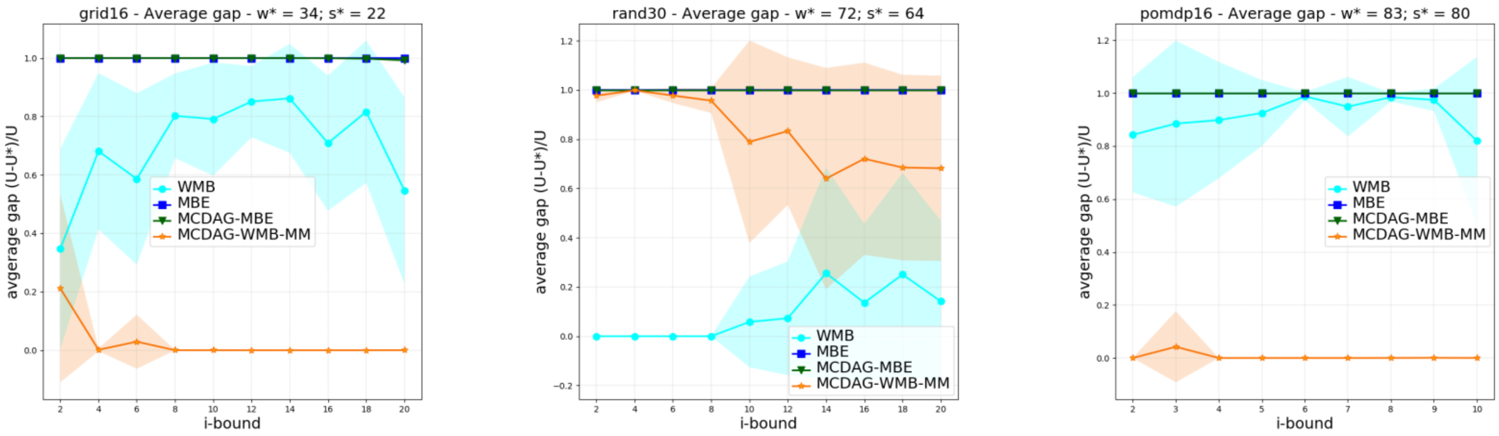
- The upper bounds obtained can be tighten further using cost-shifting
  - Use weighted elimination instead of regular elimination
    - $\sum_X^w f = \left(\sum_X f^{\frac{1}{w}}\right)^w$
  - Moment-matching between mini-buckets for SUM clusters [Marinescu et al., 2014]
    - Let $Q = \{Q_1, \dots, Q_R\}$ be a mini-bucket partitioning such that $\psi_r = \prod_{f \in Q_r} f$ and assign weight $w_r > 0$ to each mini-bucket Q_r such that $\sum_r w_r = 1$ (X is the eliminated)
    - Re-parameterize $\psi_r = \psi_r \left(\frac{\mu}{\mu_r}\right)^{w_r}, \mu_r = \sum_{Y_r} \psi_r^{1/w_r}, \mu = \prod_r \mu_r^{w_r}, Y_r = vars(Q_r) \setminus X$
  - Moment-matching between mini-buckets for MAX clusters
    - Let $Q = \{Q_1, \dots, Q_R\}$ be a mini-bucket partitioning
    - If $\psi_r = \prod_{f \in Q_r} f$ then re-parameterize $\psi_r = \psi_r(\frac{\mu}{\mu_r})$, $\mu_r = \max_{Y_r} \psi_r, \mu = (\prod_r \mu_r)^{1/R}$
    - If $\psi_r = \sum_{f \in Q_r} f$ then re-parameterize $\psi_r = \psi_r - \mu_r + \frac{1}{R}\mu$, $\mu_r = \max_{Y_r} \psi_r, \mu = \sum_r \mu_r$

Experimental Results

- Algorithms for IDs
  - MBE [Dechter, 2000]
    - Mini-bucket approximation over a join-tree
  - WMB [Lee et al., 2019]
    - Weighted mini-buckets using a valuation algebra for influence diagrams
  - MCDAG-MBE
    - Mini-buckets over MCDAGs
  - MCDAG-WMB-MM
    - Weighted mini-buckets over MCDAGs with moment-matching

- Benchmarks
  - Random: grids, random graphs, POMDPs
  - Planning: system administrator [Guestrin et al., 2003]

Results: random influence diagrams



Gap $\rho = \frac{(U-U^*)}{U}$, relative to the tightest upper bound U^*

w^*- induced width (join-tree); s^*- induced width (MCDAG)

Lower (closer to 0) is better

Results: planning instances (sysadmin)

instance	algorithm	i=2	i=10	i=18
sys1_s=10_t=3	MBE	2.09E+34	2.38E+18	7.70E+13
c=79,d=30	MCDAG-MBE	2.82E+24	2.88E+09	2.60E+06
w*=60	WMB	1.02E+10	8.37E+07	4.34E+06
s*=58,k=3	MCDAG-WMB-MM	3.44E+07	8.79E+03	**4.31E+02**
sys1_s=10_t=4	MBE	1.72E+46	3.01E+26	1.11E+19
c=102,d=40	MCDAG-MBE	1.93E+35	3.79E+14	1.39E+10
w*=80	WMB	6.59E+13	2.87E+11	3.04E+08
s*=78,k=3	MCDAG-WMB-MM	1.18E+11	2.68E+06	**3.98E+04**
sys1_s=10_t=5	MBE	1.38E+58	5.43E+30	6.27E+22
c=125,d=50	MCDAG-MBE	1.33E+46	3.18E+19	9.12E+13
w*=100	WMB	1.80E+17	7.67E+13	4.34E+11
s*=98,k=3	MCDAG-WMB-MM	4.09E+14	1.36E+09	**1.46E+07**

Smaller values are better

Conclusion

- Revisit MCDAG decompositions for influence diagrams and develop a partitioning-based approximation scheme for bounding the maximum expected utility
- MCDAGs are more sensitive to the underlying problem structure than strong join-trees
  - Smaller induced width led to a partitioning that yields more accurate bounds
- Apply cost-shifting by moment-matching to tighten the bounds further
- Experiments on difficult benchmark problem instances demonstrate the effectiveness of our proposed bounding scheme compared with existing state-of-the-art approaches
- Future work: using these bounds as heuristics for guiding search algorithms for finding optimal policies, as well as developing a more powerful iterative cost-shifting scheme between the clusters of the MCDAG decomposition