A Weighted Mini-Bucket Bound for Solving Influence Diagrams

Junkyu Lee[1], Radu Marinescu[2], Alexander Ihler[1], Rina Dechter[1]

[1] University of California, Irvine {junkyul, ihler, dechter}@ics.uci.edu [2] IBM Research, Ireland {radu.marinescu}@ie.ibm.com

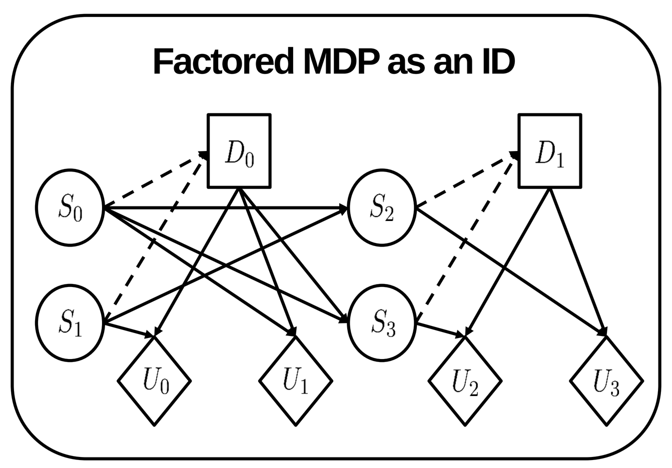
Summary

✓ A bounding scheme that interleaves variable elimination and reparameterization on valuations for influence diagrams (IDs) over the weighted mini-bucket decompositions of influence diagram.

✓ It improves the quality of the bound and computation time compared with state-of-the art decomposition bounds of IDs, and generate admissible heuristic evaluation functions suitable for AND/OR graph search.

Backgrounds

❖ Influence Diagram



- Chance variables $\mathbf{C} = \{S_0, S_1, S_2, S_3\}$
- Decision variables $\mathbf{D} = \{D_0, D_1\}$
- Probability functions $\mathbf{P} = \{P(S_0), \dots, P(S_3|pa(S_3))\}$
- Utility functions $\mathbf{U} = \{U_0(pa(U_0), \cdots, U_3(pa(U_3)\}$
- Partial ordering constraint $\mathcal{O} = \{S_0, S_1\} \prec \{D_0\} \prec \{S_2, S_3\} \prec \{D_1\}$
- Policy functions $\Delta_0 := S_0 \times S_1 \mapsto D_0 \quad \Delta_1 := S_2 \times S_3 \mapsto D_1$
- MEU and optimal policy

$$\text{MEU} := \textstyle\sum_{S_0,S_1} \max_{D_0} \sum_{S_2,S_3} \max_{D_1} [\prod_{P_i \in \mathbf{P}} P_i][\sum_{U_i \in \mathbf{U} U_i}][\prod_{\Delta_i \in \mathbf{\Delta}} \Delta_i]$$

❖ Valuation Algebra for Influence Diagrams [Shenoy 1992, Jensen 1994, Lauritzen 1997, Maua, 2012, Moral 2018]

Valuation for IDs: $\Psi(\mathbf{X}) := (P(\mathbf{X}), V(\mathbf{X}))$ (probability, expected utility value)

Combination

$$\Psi_1 := (P_1, V_1) \quad \Psi_2 := (P_2, V_2)$$
$$\Psi_1 \otimes \Psi_2 := (P_1P_2, P_1V_2 + P_2V_1)$$

Marginalization

$$\Psi := (P(\mathbf{X}), V(\mathbf{X}))$$
$$\textstyle\sum_{\mathbf{Y}}^{\mathbf{w}} \Psi := (\sum_{\mathbf{Y}}^{\mathbf{w}} P, \sum_{\mathbf{Y}}^{\mathbf{w}} V) \quad \mathbf{Y} \subseteq \mathbf{X}$$

Valuations for IDs form a commutative semi-ring

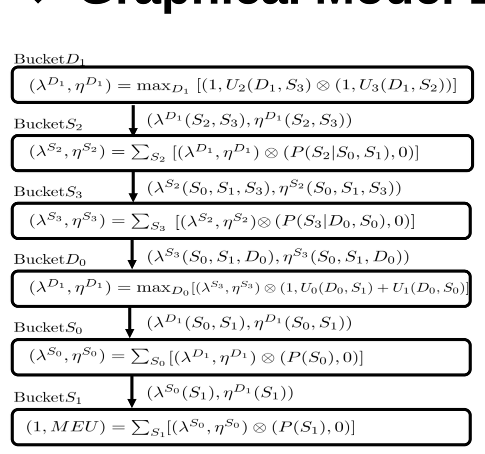
✓ Axiomatization of valuation algebra ensures decomposition by re-arranging combinations and marginalizations (distributive law).

✓ Local computation is implemented by Bucket Elimination. [Dechter, 1999]

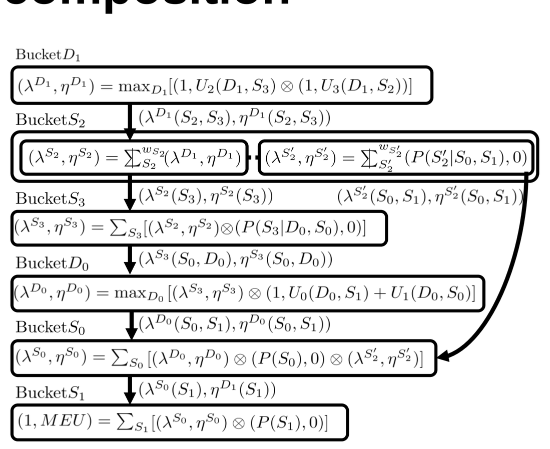
Re-writing MEU Query by Valuation Algebra for IDs

$$\mathbf{\Psi} := \{(P_i, 0)|P_i \in \mathbf{P}\} \cup \{(1, U_i)|U_i \in \mathbf{U}\} \qquad \text{MEU} := \textstyle\sum_{S_0,S_1} \max_{D_0} \sum_{S_2,S_3} \max_{D_1} \otimes_{i \in \mathcal{I}_\Psi} \Psi_i$$

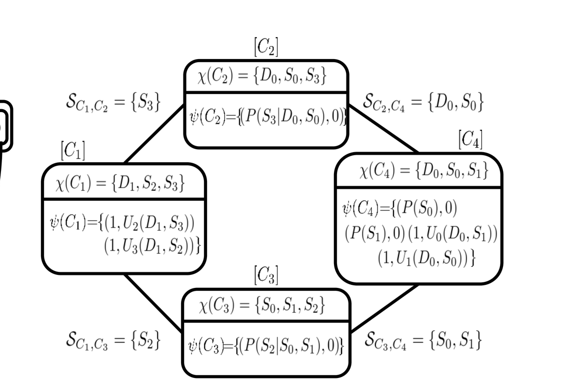
❖ Graphical Model Decomposition



Bucket Tree Decomposition [Dechter, 1999]
✓ **exact**
✓ **local computation**



Mini-Bucket Tree Decomposition [Dechter, 2003]
✓ **Approximation**
✓ **bounded complexity**



Join-Graph Decomposition [Mateescu 2010]
✓ **Approximation**
✓ **iterative and loopy message passing**

- The exact decomposition captures dynamic programming structure.
- The approximate decomposition introduces auxiliary variables and bounds the complexity.
- Messages can be passed over a decomposed join-graph to generate upper bounds.

A Weighted Mini-bucket Bound for IDs

❖ Previous Work: Decomposition Bounds for IDs [Lee, Ihler, Dechter, 2018]

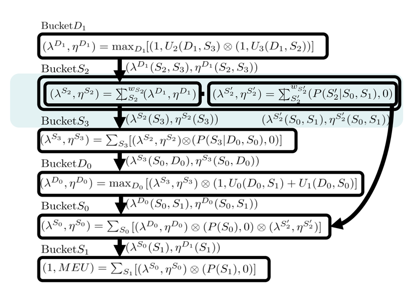
✓ When marginalizing X from the combination of valuations, exchange combination and marginalization.

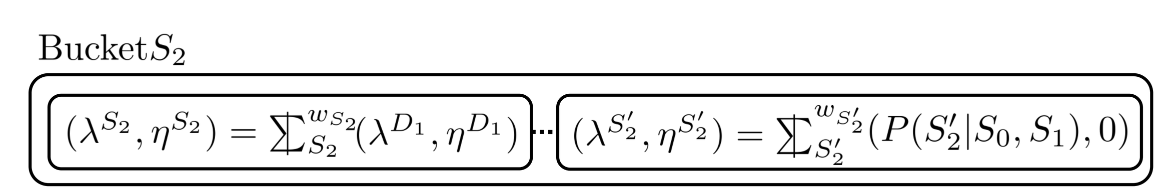
$$\sum_X^{w_X} \Psi_1(\mathbf{X}_1) \bigotimes \Psi_2(\mathbf{X}_2) \le \sum_{X_1}^{w_X^1} \Psi_1(\mathbf{X}_1) \bigotimes \sum_{X_2}^{w_X^2} \Psi_2(\mathbf{X}_2) \qquad w_X = w_X^1 + w_X^2$$

✓ The powered-sum operator

$$\sum_Y^w \Psi(\mathbf{X}) := \Big(\sum_Y^w P(\mathbf{X}), \sum_Y^w V(\mathbf{X})\Big) \qquad \sum_Y^w f(\mathbf{X}) = [\sum_Y |f(\mathbf{X})|^{1/w}]^w \qquad 0 \le w \le 1$$

❖ Weighted Mini-Bucket Decomposition for IDs



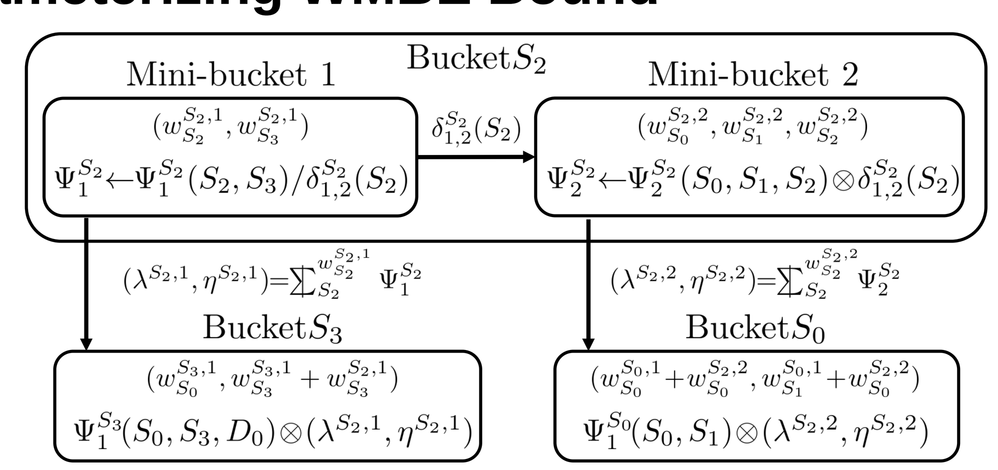


$$\textstyle\sum_{S_2}^{w_{S_2}} (\lambda^{D_1}(S_2,S_3), \eta^{D_1}(S_2,S_3) \bigotimes (P(S_2|S_0,S_1),0) \le \Big[\sum_{S_2}^{w_{S_2}^{S_2,1}} (\lambda^{D_1}(S_2,S_3), \eta^{D_1}(S_2,S_3)\Big] \bigotimes \Big[\sum_{S_2}^{w_{S_2}^{S_2,2}} (P(S_2|S_0,S_1),0)\Big]$$
$$w_{S_2} = 1 \qquad w_{S_2}^{S_2,1} + w_{S_2}^{S_2,2} = 1$$

✓ Bucket S_2 is decomposed to Mini-bucket 1 and 2 with "i-bound" 2.

✓ Processing all remaining buckets generates an upper bound of MEU.

❖ Reparameterizing WMBE Bound

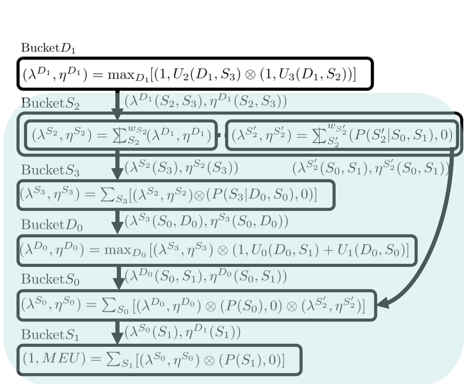


$$\textstyle\sum_{S_2}^{w_{S_2}} \frac{\Psi_1^{S_2}(S_2,S_3)}{\delta_{1,2}^{S_2}(S_2)} \bigotimes \Psi_2^{S_2}(S_0,S_1,S_2) \otimes \delta_{1,2}^{S_2}(S_2) \le \Big[\sum_{S_2}^{w_{S_2}^{S_2,1}} \frac{\Psi_1^{S_2}(S_2,S_3)}{\delta_{1,2}^{S_2}(S_2)}\Big] \bigotimes \Big[\sum_{S_2}^{w_{S_2}^{S_2,2}} \Psi_2^{S_2}(S_0,S_1,S_2) \otimes \delta_{1,2}^{S_2}(S_2)\Big]$$
$$w_{S_2} = 1.0 \qquad w_{S_2}^{S_2,1} + w_{S_2}^{S_2,2} = 1 \qquad \frac{\mathbf{1}(S_2)}{\delta_{1,2}^{S_2}(S_2)} \bigotimes \delta_{1,2}^{S_2}(S_2)$$

✓ Introduce cost shifting functions between mini-buckets.

✓ Tighten the upper bound by optimizing weights and costs.

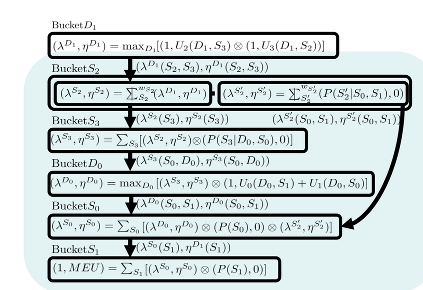
❖ Optimization Objective for WMBE-ID Bounds



$$\begin{aligned}
(1, \text{MEU}) \le \;& \Psi_{\text{MEU}_\text{WMB}} := \textstyle\sum_{S_1}^1 \Psi_1 \sum_{S_0}^1 \otimes \Psi_0 \sum_{D_0}^0 \otimes \Psi_{D_0} \sum_{S_3}^1 \otimes \Psi_3 \\
& \otimes \Big[\textstyle\sum_{S_2}^{w_{S_2}^{S_2,1}} \frac{\Psi_1^{S_2}(S_2,S_3)}{\delta_{1,2}^{S_2}(S_2)}\Big] \otimes \Big[\sum_{S_2}^{w_{S_2}^{S_2,2}} \Psi_2^{S_2}(S_0,S_1,S_2) \otimes \delta_{1,2}^{S_2}(S_2)\Big] \\
\le \;& \Psi_{\text{MEU}_\text{JGDID}} := \Big[\textstyle\sum_{S_1}^{\mathbf{w}^{S_1}} \Psi_1(S_1)\Big] \otimes \Big[\sum_{S_1,S_0}^{\mathbf{w}^{S_0}} \Psi_0(S_0,S_1)\Big] \\
& \otimes \Big[\textstyle\sum_{S_1,S_0,D_0}^{\mathbf{w}^{D_0}} \Psi_{D_0}(S_3,D_0,S_0)\Big] \otimes \Big[\sum_{S_1,S_0,D_0,S_3}^{\mathbf{w}^{S_3}} \Psi_3(S_3,D_0,S_0)\Big] \\
& \otimes \Big[\textstyle\sum_{S_1,S_0,D_0,S_3}^{\mathbf{w}^{S_2,1}} \sum_{S_2}^{w_{S_2}^{S_2,1}} \frac{\Psi_1^{S_2}(S_2,S_3)}{\delta_{1,2}^{S_2}(S_2)}\Big] \\
& \otimes \Big[\textstyle\sum_{S_1,S_0,D_0,S_3}^{\mathbf{w}^{S_2,2}} \sum_{S_2}^{w_{S_2}^{S_2,2}} \Psi_2^{S_2}(S_0,S_1,S_2) \otimes \delta_{1,2}^{S_2}(S_2)\Big]
\end{aligned}$$

✓ "fully" decomposed bound, $\Psi_{\text{MEU}_\text{JGDID}}$ is a surrogate upper bound.

❖ Optimization Setup for Computing Weights and Costs



Objective: value component of $\Psi_{\text{MEU}_\text{JGDID}}$

Parameters: $w_{S_2}^{S_2,1} \quad w_{S_2}^{S_2,2}$

$$\delta_{1,2}^{S_2}(S_2) := (\lambda_{1,2}^{S_2}(S_2), \eta_{1,2}^{S_2}(S_2))$$

Constraints: $w_{S_2}^{S_2,1} + w_{S_2}^{S_2,2} = 1 \quad 0 \le w_{S_2}^{S_2,1}, w_{S_2}^{S_2,2} \le 1$

$$\frac{\eta_1^{S_2}}{\lambda_1^{S_2}} - \frac{\eta_{1,2}^{S_2}}{\lambda_{1,2}^{S_2}} \ge 0 \qquad \frac{\eta_2^{S_2}}{\lambda_2^{S_2}} + \frac{\eta_{1,2}^{S_2}}{\lambda_{1,2}^{S_2}} \ge 0 \qquad \lambda_{1,2}^{S_2}(S_2) \ge 0$$

Algorithm

Algorithm 1 Weighted Mini-Bucket Elimination Bounds for IDs (WMBE-ID)

Require: Influence diagram $\mathcal{M} = \langle \mathbf{X}, \mathbf{\Psi}, \mathcal{O} \rangle$, total constrained elimination order $\mathcal{O} := \{X_N, X_{N-1}, \dots, X_1\}$, i-bound, iteration limit L,
Ensure: an upper bound of the MEU

```
1: Generate a schematic mini-bucket tree and allocate valuations to mini-buckets, {Q_{X_i} | X_i ∈ X}.
2: Initialize weights {w_{1:N}^{X_i,α} | ∀X_i ∈ X, α ∈ Q_{X_i}}, and optionally tune the weights by a single JGDID update.
3: Compute fully decomposed bounds at all mini-buckets.
4: Ub ← (1, 0)
5: for i ← N to 1 do
6:   iter = 0
7:   while iter < L and bounds not converged do
8:     Update cost functions {δ_(α,α+1)|α∈Q_{X_i}} by SLSQP.   [Sequential least squares programming]
9:     Update weights {w_{X_i}^{X_i,α} | α∈Q_{X_i}} by EGD.   [Exponentiated Gradient Descent]
10:  end while
11:  for α ∈ Q_{X_i} do
12:    (λ_α^{X_i}, η_α^{X_i}) ← Σ_{X_i}^{w_{X_i}^{X_i,α}} Ψ~_α^{X_i}(X_α^{X_i})
13:    if (λ_α^{X_i}, η_α^{X_i}) is constant then
14:      Ub ← Ub ⊗ (λ_α^{X_i}, η_α^{X_i})
15:    else
16:      Send message (λ_α^{X_i}, η_α^{X_i}) downward.
17:      Combine the message and the valuation at the destination mini-bucket.
18:      Merge the weights of the source and destination mini-buckets.
19:      Recompute fully decomposed bound at the destination mini-bucket.
20:    end if
21:  end for
22: end for
23: Return value component of Ub
```

Initialization
- Ln1: Create a mini-bucket tree and join-graph, allocate functions to mini-buckets and initialize weights uniformly.
- Ln2: WMBE-WC2: optimize weights w.r.t. the initial function allocations (optional).

Interleaving VE and optimization
- Ln5: Process one layer of mini-buckets at a time.
- Ln7-Ln10: Jointly optimize cost functions, and weights over the mini-buckets for iteration limits.
- Ln12-Ln16: Generate messages by marginalization and pass messages downward.
- Ln17-18: Combine messages and weights before eliminating the next layer.

Algorithm configurations
- MBE: no optimization, no weights.
- WMBE-U : no optimization, use uniform weights.
- WMBE-UC: optimize cost functions only.
- WMBE-WC1: optimize both but skip initial weight optimization step over the join graph.
- WMBE-WC2: optimize both and perform optional weight optimization step over the join graph.

Experiments

❖ Comparing Decomposition Bounds for IDs

Instance (n,f,k,s,w)	Algorithm	i=1 (time (sec), bound) iter=1	5	10	20	i=5 (time (sec), bound) iter=1	5	10	20
pomdp8-14_9_3_12_4 (96, 140, 2, 6, 47)	WMBE-U	(312, 9.3E+14)	-	-	-	(2, 3.4E+14)	-	-	-
	WMBE-UC	(899, 9.3E+14)	(1198, 9.3E+14)	(1540, 9.3E+14)	(574, 9.3E+14)	(980, 2.4E+13)	(1478, 2.4E+13)	(1229, 2.4E+13)	(894, 2.4E+13)
	WMBE-WC1	(502, 3.30E+13)	**(1659, 1.62E+13)**	(1923, 1.62E+13)	(3772, 1.62E+13)	(499, 1.67E+12)	(1534, 1.08E+12)	**(1378, 1.08E+12)**	(1892, 1.08E+12)
	WMBE-WC2	(708, 2.41E+12)	(2200, 2.32E+12)	(2631, 2.33E+12)	**(4283, 2.24E+12)**	(1531, 3.12E+11)	(2567, 3.14E+11)	**(3451, 2.96E+11)**	(2967, 2.97E+11)
	JGDID	(2.05, 1.8E+16)	(1920, 3.73E+10)	(4458, 8.65E+9)	**(7111, 6.12E+8)**	**(3, 2.75E+19)**	-	-	-
	MBE	(3, 1.94E+22)	-	-	-	(3, 2.57E+18)	-	-	-
	MBE-Re	(4, 2.9E+18)	-	-	-	(2, 3.4E+14)	-	-	-
	Algorithm	i=10 (time (sec), bound) iter=1	5	10	20	i=15 (time (sec), bound) iter=1	5	10	20
	WMBE-U	(755, 1.90E+10)	-	-	-	(1861, 9.49E+8)	-	-	-
	WMBE-UC	(773, 1.90E+10)	(1575, 1.90E+10)	(968, 1.90E+10)	(607, 1.90E+10)	(2022, 9.49E+8)	(3186, 9.49E+8)	(1606, 9.49E+8)	(2363, 9.49E+8)
	WMBE-WC1	**(510, 3.97E+9)**	(1409, 4.33E+9)	(1209, 4.33E+9)	(1867, 4.33E+9)	**(792, 4.42E+8)**	(2409, 1.19E+10)	(3588, 4.71E+8)	(9236, 4.70E+8)
	WMBE-WC2	(4582, 2.63E+9)	**(4099, 2.23E+9)**	(4394, 2.23E+9)	(4970, 2.23E+9)	(5923, 6.82E+7)	(12317, 6.72E+7)	(11869, 6.72E+7)	**(7892, 6.72E+7)**
	JGDID	(4, 3.2E+18)	(3224, 3.48E+12)	(7101, 5.55E+11)	**(14635, 4.42E+9)**	(12, 1.4E+18)	(12196, 4.39E+11)	(28338, 1.29E+11)	**(49188, 6.81E+9)**
	MBE	(2, 1.4E+15)	-	-	-	(3, 7.3E+12)	-	-	-
	MBE-Re	(1, 3.26E+8)	-	-	-	(1, 2.84E+5)	-	-	-
Instance (n,f,k,s,w)	Algorithm	i=1 (time (sec), bound) iter=1	5	10	20	i=5 (time (sec), bound) iter=1	5	10	20
rand-c70d7o1-01 (91, 91, 2, 3, 41)	WMBE-U	(64, 1028113)	-	-	-	(109, 9227)	-	-	-
	WMBE-UC	(371, 1.05E+6)	(232, 1.05+6)	(263, 1.05+6)	(266, 1.05+6)	(481, 7588)	(341, 8196)	(687, 8196)	(344, 7625)
	WMBE-WC1	**(76, 3.98E+5)**	(171, 4.93E+5)	(241, 4.90E+5)	(314, 4.89E+5)	**(417, 6081.49)**	(415, 7247.53)	(324, 6235.09)	(249, 8662.08)
	WMBE-WC2	(407, 6.04E+6)	**(856, 1.38E+5)**	(796, 1.35E+5)	(687, 1.60E+5)	(251, 4201.44)	**(633, 2567.81)**	(868, 2877.79)	(678, 3639.72)
	JGDID	(1, 2.15E+7)	(337, 61104)	(842, 758)	**(1453, 686)**	(1, 1.84E+7)	(1248, 11213)	(3195, 762)	**(4397, 736)**
	MBE	(1, 2.69E+9)	-	-	-	(1, 4.46E+5)	-	-	-
	MBE-Re	(1, 2.61E+9)	-	-	-	(2, 7.39E+5)	-	-	-
	Algorithm	i=10 (time (sec), bound) iter=1	5	10	20	i=15 (time (sec), bound) iter=1	5	10	20
	WMBE-U	(160, 1863)	-	-	-	(193, 1542)	-	-	-
	WMBE-UC	(793, 1835)	(1460, 1820)	(576, 1819)	(911, 1820)	(833, 1521)	(1560, 1518)	(1056, 1516)	(930, 1512)
	WMBE-WC1	**(315, 1803.52)**	(357, 2041.08)	(585, 1870.45)	(524, 1866.48)	(702, 2.08E+6)	(1005, 1956.82)	(1111, 1846.16)	**(1231, 1828.79)**
	WMBE-WC2	(1326, 1107.85)	(1083, 1105.79)	(953, 1140.48)	**(819, 1098.92)**	(2625, 1086.30)	**(2081, 1061.17)**	(2006, 1077.61)	(2583, 1077.98)
	JGDID	(1, 20049757)	(4228, 2167)	(8358, 1303)	**(15787, 795)**	(2, 2.37E+7)	(7837, 2006)	**(15134, 1357)**	-
	MBE	(1, 4937)	-	-	-	(1, 7027)	-	-	-
	MBE-Re	(1, 40695)	-	-	-	(1, 16139)	-	-	-

- Compared 4 different configurations of the WMBE-ID, state-of-the-art JGDID, and MBE, MBE with relaxed ordering.
- i-bounds 1, 5, 10, 15, maximum iterations 1, 5, 10, 20
- Non-iterative algorithms (WMBE-U, MBE, MBE-Re) produced very loose bounds.
- At lower i-bounds, JGIDID performs the best.
- At higher i-bounds, WMBE-WC1,2 improved the bounds and produced tighter bounds in shorter time.

❖ WMBE-ID vs. JGDID(i=1)

	WMBE-WC2												JGDID (i=10)	
Instance	i=10, iter=1		i=10, iter=5		i=15, iter=1		i=15, iter=5		i=20, iter=1		i=20, iter=5		i=10, max iter 100	
ID_from_BN_0_w28d6	13%	2.60	19%	2.74	61%	1.40	40%	2.13	193%	1.33	248%	1.37	410%	1.30
ID_from_BN_0_w29d6	10%	1.56	29%	1.52	26%	**0.96**	30%	**0.94**	132%	1.01	210%	1.02	255%	1.58
ID_from_BN_78_w19d3	40%	1.55	70%	1.51	31%	**0.92**	56%	**0.95**	244%	**0.64**	299%	**0.64**	1582%	1.70
ID_from_BN_78_w23d6	5%	1.53	5%	1.51	6%	1.24	8%	1.16	31%	**0.87**	38%	**0.84**	127%	1.79
ID_from_BN_78_w24d6	16%	2.12	21%	3.42	17%	1.15	31%	1.08	54%	**0.72**	90%	**0.70**	167%	1.83
mdp5-16_3_8_10	10%	**0.88**	13%	**0.87**	88%	**0.76**	91%	**0.76**	-	-	-	-	114%	4.68E+11
mdp6-20_5_5_5	13%	**0.94**	20%	**0.94**	28%	**0.85**	34%	**0.86**	771%	**0.80**	659%	**0.80**	339%	2.69E+04
mdp7-28_3_6_5	14%	1.14E+07	17%	**0.95**	20%	**0.93**	26%	**0.93**	-	-	333%	**0.87**	128%	1.56E+07
mdp8-28_3_6_4	21%	**0.97**	41%	**0.96**	34%	**0.91**	42%	**0.90**	287%	**0.87**	365%	**0.87**	228%	8.69E+03
mdp9-32_3_8_3	24%	**0.93**	46%	**0.91**	46%	**0.90**	81%	**0.88**	503%	**0.84**	520%	**0.83**	142%	2.65E+03
pomdp10-12_7_3_8_4	80%	**0.10**	93%	**0.07**	667%	17.11	1008%	**4.06E-03**	-	-	-	-	264%	**0.32**
pomdp6-12_6_2_6_3	26%	8.59	78%	4.25	43%	**0.17**	98%	**0.16**	607%	**0.01**	489%	**0.01**	109%	7.36
pomdp7-20_10_2_10_3	27%	1083.13	53%	20.72	51%	**0.89**	59%	34.93	827%	1.78	1247%	**0.03**	174%	72.80
pomdp8-14_9_3_12_4	42%	4.30	38%	3.64	55%	**0.11**	114%	**0.11**	1504%	**0.03**	2022%	**0.01**	199%	7.22
pomdp9-14_8_3_10_4	56%	1.27	87%	**0.87**	1455%	**0.02**	1775%	**0.02**	-	-	-	-	395%	17.00
rand-c50d15o1-03	56%	1.83	25%	1.98	51%	1.07	82%	1.04	379%	**0.87**	576%	**0.87**	1515%	1.35
rand-c50d5o1-01	42%	**0.91**	86%	**0.90**	44%	**0.69**	78%	**0.66**	61%	**0.61**	73%	**0.61**	1687%	1.02
rand-c70d14o1-01	79%	9.64	121%	3.31	172%	11.55	252%	13.53	2073%	1.65	2284%	1.78	1872%	1.23
rand-c70d21o1-01	56%	3.70	28%	3.72	85%	1.36	122%	1.24	604%	1.17	788%	1.22	885%	1.16
rand-c70d7o1-01	91%	1.61	75%	1.61	181%	1.58	143%	1.55	338%	**0.73**	422%	**0.73**	1475%	1.15
geometric mean	27%	4.96	37%	1.62	59%	**0.89**	81%	**0.71**	313%	**0.56**	384%	**0.41**	361%	66.03

- Compared the ratio of the time and bound from WMBE-WC2 against JGDID(i=1).
- At high i-bounds, WMBE-WC2 produced tighter bounds (ratio less than 1.0) than JGDID(i=1) in most of the instances.

This work supported in part by NSF grants IIS=1526842 and IIS-1254071, the U.S. Air Force (Contract FA9453-16-C-0508), and DARPA (contract W911NF-18-C-0015).