Heuristic AND/OR Search for Solving Influence Diagrams

Junkyu Lee,[1] Radu Marinescu,[2] Rina Dechter,[1]
[1]University of California, Irvine, [2]IBM Research, Ireland
junkyul@uci.edu, radu.marinescu@ie.ibm.com, dechter@ics.uci.edu

Abstract

We address an AND/OR search for solving influence diagrams with heuristics derived from graphical model decomposition bounds. Then, we present how the heuristics guide AND/OR branch and bound search and show its potential for solving influence diagrams on a preliminary experiment.

Introduction

An influence diagram (ID) (Howard and Matheson 1981) is a graphical model for the MEU task, where a directed acyclic graph represents probability, utility, and policy functions capturing their local structure. Our focus is on solving influence diagrams using the framework of *AND/OR search with decomposition-based heuristics*. This framework (Dechter and Mateescu 2007) proved effective for Maximum A Posteriori (MAP) queries (Marinescu and Dechter 2009), for summation queries (Dechter and Mateescu 2007), and for marginal MAP (Marinescu et al. 2018). We summarize AND/OR branch and bound algorithms for influence diagrams (AOBB-ID), described in (Marinescu and Dechter 2009), and our more recent decomposition bounds (Lee, Ihler, and Dechter 2018), (Lee et al. 2019). Preliminary empirical results show the potential of augmenting AOBB-ID with heuristics derived from our recent decomposition bounds.

Graphical Models for Solving IDs

An ID is a tuple $\mathcal{M} \coloneqq \langle \mathbf{C}, \mathbf{D}, \mathbf{P}, \mathbf{U}, \mathcal{O}, \bigotimes \rangle$ consisting of a set of discrete random variables $\mathbf{C}$, a set of discrete decision variables $\mathbf{D}$, a set of conditional probability functions $\mathbf{P}$, a set of utility functions $\mathbf{U}$, and a constrained variable ordering $\mathcal{O}$ on chance and decision variables. The valuation algebra for IDs (Mauá, de Campos, and Zaffalon 2012) evaluates the conditional expected utility by single combination operator $\bigotimes$ over a semi-ring on a pair of probability and value functions called potential (Jensen, Jensen, and Dittmer 1994). A potential $\Psi(\mathbf{Y})$ defined over $\mathbf{Y}$ is a pair of functions $\Psi(\mathbf{Y}) = (P(\mathbf{Y}), V(\mathbf{Y}))$ and $(P_1, V_1) \bigotimes (P_2, V_2) := (P_1 P_2, P_1 V_2 + P_2 V_1)$. The marginalization operators apply component-wise maximization or summation, denoted by $\Downarrow X$. The MEU task can be written as

$$\sum_{\mathbf{I}_0} \max_{D_0} \dots \sum_{\mathbf{I}_{|\mathbf{D}|-1}} \max_{D_{|\mathbf{D}|-1}} \sum_{\mathbf{I}_{|\mathbf{D}|}} \bigotimes_{\Psi_\alpha \in \mathbf{P} \cup \mathbf{U}} \Psi_\alpha(\mathbf{X}_\alpha). \tag{1}$$

The reformulation of MEU task by valuation algebra not only compacts the notation but also allows a relative smooth extension of the AND/OR search framework.

Heuristic AND/OR Search for IDs

Primal graph and its Pseudo-tree. A primal graph $\mathcal{G}_p$ of an ID is an undirected graph over nodes associated with variables, and its edges connect nodes appearing together in some function. The decomposition structure of an ID relative to a variable ordering is captured by a pseudo tree $\mathcal{T}_{\mathcal{G}_p}$. A *pseudo tree* of an ID $\mathcal{T}_{\mathcal{G}_p}$ is a spanning tree of $\mathcal{G}_p$ satisfying the following properties: (1) all edges appear in $\mathcal{G}_p$ are back-arcs in $\mathcal{T}_{\mathcal{G}_p}$, and (2) any total order from root to a leaf of the $\mathcal{T}_{\mathcal{G}_p}$ conforms to the ordering $\mathcal{O}$ inherent in the given ID. A pseudo-tree is also called a *bucket-tree* where each variable X is associated with a bucket B_X which is subset of the ID's functions. Ψ_{B_X} is the combined function in B_X (see (Marinescu 2010)).

An AND/OR search tree $\mathcal{S}_\mathcal{T}(\mathcal{M})$ has alternating levels of nodes, organized along the pseudo-tree $\mathcal{T}_{\mathcal{G}_p}$, where one level corresponds to a variable and the next to its value assignments. There are 4 types of nodes: chance variable nodes, decision variable nodes, chance assignment nodes, and decision assignment nodes. Only the decision variable nodes are *OR* nodes in the ID's AND/OR search tree.

Let π be an assignment along the path from the root to a node n, $asgn_\pi(\Psi)$ assign the values to potential Ψ. The arc weight from a variable node to its assignment node can be expressed by the valuation algebra as $w_{(n,m)} := asgn_\pi(\Psi_{B_X})$ and the weight from an assignment node to a variable child node is $w_{(n,m)} := (1, 0)$. We define the value Ψ_n of a node $n \in \mathcal{S}_\mathcal{T}(\mathcal{M})$ to be the MEU below the node (i.e., restricted to the path values leading to it) and, having K child nodes $\{m_1, \dots, m_K\}$, it can be computed recursively from leaves to root by $\Psi_n := (1, 0)$ for a terminal assignment node n, $\Psi_n := \otimes_{i=1}^K \Psi_{m_i}$ for a non-terminal assignment node n, $\Psi_n := \sum_{i=1}^K w_{(n,m_i)} \otimes \Psi_{m_i}$ for a chance variable node n, and $\Psi_n := \max_{i=1}^K w_{(n,m_i)} \otimes \Psi_{m_i}$ for a decision variable node n.

Any subtree $\mathcal{P}_{\mathcal{S}_\mathcal{T}}$ of an AND/OR search tree $\mathcal{S}_\mathcal{T}$ defines a policy function if it contains the root of $\mathcal{S}_\mathcal{T}$, all the child nodes of *AND* nodes in the subtree, and only one child node from each *OR* node in the subtree. Its terminal nodes are leaves of $\mathcal{S}_\mathcal{T}$. A partial policy tree $\mathcal{P}'_{\mathcal{S}_\mathcal{T}}$ is a subtree of a policy tree containing the root of $\mathcal{S}_\mathcal{T}$. The value of a policy tree $\mathcal{P}_{\mathcal{S}_\mathcal{T}}$ is the value of its root node. An AND/OR search algorithm search for an optimal full policy tree. Often, different nodes in the AND/OR search tree root identical subtrees. Those can be merged, converting the search tree into a search graph thus allowing more efficient search via caching. Such Identical subproblems can be identified by *contexts* (Dechter and Mateescu 2007).

Heuristics derived from Decomposition Bounds. As noted, a pseudo tree $\mathcal{T}_{\mathcal{G}_p}$ has a corresponding bucket tree $\mathcal{BT}$. Mini-bucket elimination (MBE) scheme (Dechter and Rish 2003) provides a decomposition bound that has a compatible structure with a pseudo tree so it can generate heuristic functions for AND/OR search. The quality of the upper bounds can be improved by optimizing the bounds using additional parameters yielding the weighted mini-buckets (WMB) scheme (Liu and Ihler 2011).

The WMB bound for ID (WMBE-ID) specializes the WMB scheme to produce upper bounds for the MEU. Following a total constrained order $\hat{\mathcal{O}}$ consistent with the constrained orderings $\mathcal{O}$ imposed by the ID, the pseudo-tree defines a bucket tree. We partition each bucket B_X along the pseudo tree $\mathcal{T}_{\mathcal{G}_p}$ to mini-buckets $\mathcal{I}_{B_X} := \{B_{X^1}, \ldots, B_{X^p}\}$ by introducing auxiliary variables $X^1, \ldots, X^p$ for each mini-bucket and constraints $X = X^1 = \ldots = X^p$ ensuring that the total number of variables does not exceed i+1, for a given i-bound i. Then, we generate messages by eliminating the labeling variable from the combined potential and send the result to a mini-bucket in a lower layer. The mini-bucket partitioning stage yields an upper bound since it amounts to exchanging combination and marginalization operator:

$$\Downarrow_X \otimes_{i \in \mathcal{I}_{B_X}} \Psi_{B_{X^i}} \leq \bigotimes_{i \in \mathcal{I}_{B_X}} \Downarrow_X \Psi_{B_{X^i}}, \qquad (2)$$

where $\Downarrow_X$ is either $\max$ or $\sum$ depending on the type of the variable X. The RHS of Eq. (2) can be tightened by introducing additional optimization parameters yielding

$$\Downarrow_X \otimes_{i \in \mathcal{I}_{B_X}} \Psi_{B_{X^i}} \leq \bigotimes_{i \in \mathcal{I}_{B_X}} \sum_X^{w_X^i} [\Psi_{B_{X^i}} \otimes \Psi_{\delta_i}(X^i)], \qquad (3)$$

where the auxiliary optimization parameters are non-negative weights w_X^i for a chance variable X satisfying $\sum_{i \in \mathcal{I}_{B_X}} w_X^i = 1$, and cost shifting potentials $\Psi_{\delta_i}(X^i)$ satisfying $\otimes_{i \in \mathcal{I}_{B_X}} \Psi_{\delta_i}(X^i) = (1, 0)$. The powered-sum marginalization operator $\sum_X^w f(\mathbf{X})$ is defined by $[\sum_x |f(\mathbf{X})|^{\frac{1}{w}}]^w$.

We denote by $de(X)$ the set of descendants of a node associated with a variable X in $\mathcal{T}_\mathcal{P}$, by $\mathcal{I}_{B_Y}$ the set of mini-buckets for B_Y, by $asgn_\pi$ the values assigned to variables from the root to the current node in $\mathcal{T}_{\mathcal{G}_p}$, and by M_Y^p the message generated by the mini-bucket B_{Y^p}. Then, the heuristic function $h(n)$ from a mini-bucket tree optimized by WMBE-ID algorithm can be computed as follows. $h(n) = asgn_\pi \bigotimes_{Y \in de(X), p \in \mathcal{I}_{B_Y}} M_Y^p(X = x)$ if n is a node assigning value x to X, and $h(n) = \Downarrow_m \Psi_{(n,m)} \otimes h(m)$ if n is a variable node.

Instances	n, f, w, i	WMBE	OPT	AO	AOBB+MBE	AOBB+WMBE
SA1-T5	130,180,20,10	98.4	96.6	81	60	**13**
SA1-T10	250,350,20,10	197.2	183.5	180	164	**31**
SA2-T5	190,265,30,15	147.5	139.7	998(m)	2218(m)	**2062**
SA2-T10	365,515,30,15	295.7	NA	1037(m)	2518(m)	3597(m)

Table 1: Experiment Results. n is the number of variables, f is the number of functions, w is the constrained induced width, i is the i-bound, WMBE is the upper bound of the MEU from WMBE-ID, and OPT is the optimal MEU. AO, AOBB+MBE, and AOBB+WMBE show the time in seconds, and (m) indicates the failure by 24 GB memory limit.

Experiment Results

Table 1 summarizes the preliminary experiment over 4 instances of the $SysAdminMdp$ domain, where `SA1` and `SA2` instances model the problems with 10 and 15 servers up to 5 and 10 time horizons.

We see that WMBE-ID heuristic improved the running time of AOBB-ID compared with AO (Marinescu 2010) (no heuristic) and AOBB+MBE. In particular one SA2-T5, AOBB+WMBE is the only terminating algorithm while others failed due to the 24 GB memory limit.

References

Dechter, R., and Mateescu, R. 2007. And/or search spaces for graphical models. *Artificial intelligence* 171(2-3):73–106.

Dechter, R., and Rish, I. 2003. Mini-buckets: A general scheme for bounded inference. *Journal of the ACM* 50(2):107–153.

Howard, R. A., and Matheson, J. E. 1981. Influence diagrams. *Readings on the Principles and Applications of Decision Analysis* 721–762.

Jensen, F.; Jensen, F. V.; and Dittmer, S. L. 1994. From influence diagrams to junction trees. In *Proceedings of the 10th international conference on Uncertainty in artificial intelligence*, 367–373.

Lee, J.; Marinescu, R.; Ihler, A.; and Dechter, R. 2019. A weighted mini-bucket bound for solving influence diagrams. In *Proceedings of the 35th Conference on Uncertainty in Artificial Intelligence*.

Lee, J.; Ihler, A.; and Dechter, R. 2018. Join graph decomposition bounds for influence diagrams. In *Proceedings of the 34th Conference on Uncertainty in Artificial Intelligence*, 1053–1062.

Liu, Q., and Ihler, A. 2011. Bounding the partition function using Hölder's inequality. In *Proceedings of the 28th International Conference on Machine Learning*, 849–856.

Marinescu, R., and Dechter, R. 2009. And/or branch-and-bound search for combinatorial optimization in graphical models. *Artificial Intelligence* 173(16-17):1457–1491.

Marinescu, R.; Lee, J.; Dechter, R.; and Ihler, A. 2018. And/or search for marginal map. *Journal of Artificial Intelligence Research* 63:875–921.

Marinescu, R. 2010. *A New Approach to Influence Diagrams Evaluation*. Springer London. 107–120.

Mauá, D. D.; de Campos, C. P.; and Zaffalon, M. 2012. Solving limited memory influence diagrams. *Journal of Artificial Intelligence Research* 44:97–140.