Weighted Heuristic Anytime Search

Flerova, Marinescu, and Dechter

Daniel Padé[1]

[1]University of South Carolina

April 24, 2017

Basic Heuristic Searches

- Best-first Search
  Blindly follows the heuristic
- Weighted A^* Search
  For $w > 1$

$$f(n) = g(n) + w \cdot h(n)$$

Larger w yields 'greedier' searches

Basic Heuristic Searches

- Best-first Search
  Blindly follows the heuristic
- Weighted A^* Search
  For $w > 1$

$$f(n) = g(n) + w \cdot h(n)$$

Larger w yields 'greedier' searches

Graphical Models

Definition (Graphical Model)

A tuple $\mathcal{M} = \langle \boldsymbol{X}, \boldsymbol{D}, \boldsymbol{F}, \bigotimes \rangle$ where

1. $\boldsymbol{X} = \{X_0, \ldots, X_{n-1}\}$ is a set of *variables*
2. $\boldsymbol{D} = \{D_0, \ldots, D_{n-1}\}$ is a set of *domains*
3. $\boldsymbol{F} = \left\{f_0(\boldsymbol{X}_{S_0}), \ldots, f_{r-1}(\boldsymbol{X}_{S_{r-1}})\right\}$ is a set of *scopes*:
   - $\boldsymbol{X}_{S_i} \subseteq \boldsymbol{X}$
   - $\forall i.\ f_i : \boldsymbol{X}_{S_i} \to \mathbb{R}^+$
4. A *combination operator* $\bigotimes \in \{\sum, \prod\}$

The model $\mathcal{M}$ represents the function

$$C(X) = \bigotimes_{i=0}^{r-1} f_i(X_{S_i})$$

Graphical Models

Definition (Graphical Model)

A tuple $\mathcal{M} = \langle \boldsymbol{X}, \boldsymbol{D}, \boldsymbol{F}, \bigotimes \rangle$ where

1. $\boldsymbol{X} = \{X_0, \ldots, X_{n-1}\}$ is a set of *variables*
2. $\boldsymbol{D} = \{D_0, \ldots, D_{n-1}\}$ is a set of *domains*
3. $\boldsymbol{F} = \left\{f_0(\boldsymbol{X}_{S_0}), \ldots, f_{r-1}(\boldsymbol{X}_{S_{r-1}})\right\}$ is a set of *scopes*:
   - $\boldsymbol{X}_{S_i} \subseteq \boldsymbol{X}$
   - $\forall i.\ f_i : \boldsymbol{X}_{S_i} \to \mathbb{R}^+$
4. A *combination operator* $\bigotimes \in \{\sum, \prod\}$

The model $\mathcal{M}$ represents the function

$$C(\boldsymbol{X}) = \bigotimes_{i=0}^{r-1} f_i(\boldsymbol{X}_{S_i})$$

Given a model $\mathcal{M} = \langle \boldsymbol{X}, \boldsymbol{D}, \boldsymbol{F}, \bigotimes \rangle$, the most common optimization task is either *most probable explanation* or *maximum a posteriori*

MPE Find the optimal value C^*:

$$C^* = C(\boldsymbol{x}^*) = \max_{\boldsymbol{X}} \bigotimes_{i=0}^{r-1} f_i(\boldsymbol{X}_{S_i})$$

MAP Find the optimizing configuration x^*:

$$x^* = \operatorname*{argmax}_{\boldsymbol{X}} \bigotimes_{i=0}^{r-1} f_i(\boldsymbol{X}_{S_i})$$

Given a model $\mathcal{M} = \langle \boldsymbol{X}, \boldsymbol{D}, \boldsymbol{F}, \bigotimes \rangle$, the most common optimization task is either *most probable explanation* or *maximum a posteriori*

MPE Find the optimal value C^*:

$$C^* = C(\boldsymbol{x}^*) = \max_{\boldsymbol{X}} \bigotimes_{i=0}^{r-1} f_i(\boldsymbol{X}_{S_i})$$

MAP Find the optimizing configuration x^*:

$$x^* = \operatorname{argmax}_{\boldsymbol{X}} \bigotimes_{i=0}^{r-1} f_i(\boldsymbol{X}_{S_i})$$

Given a model $\mathcal{M} = \langle \boldsymbol{X}, \boldsymbol{D}, \boldsymbol{F}, \bigotimes \rangle$, the most common optimization task is either *most probable explanation* or *maximum a posteriori*

MPE Find the optimal value C^*:

$$C^* = C(\boldsymbol{x}^*) = \max_{\boldsymbol{X}} \bigotimes_{i=0}^{r-1} f_i(\boldsymbol{X}_{S_i})$$

MAP Find the optimizing configuration x^*:

$$x^* = \operatorname*{argmax}_{\boldsymbol{X}} \bigotimes_{i=0}^{r-1} f_i(\boldsymbol{X}_{S_i})$$

MPE

$$C^*_{\text{MPE}} = C(\boldsymbol{x}^*) = \max_{\boldsymbol{X}} \bigotimes_{i=0}^{r-1} f_i(\boldsymbol{X}_{S_i})$$

WCSP Weighted Constraint Satisfaction Problem (MPE in negative log-space)

$$C^*_{\text{WCSP}} := C(\boldsymbol{x}^*) = \min_{\boldsymbol{X}} \bigotimes_{i=0}^{r-1} f_i(\boldsymbol{X}_{S_i})$$

MPE

$$C^*_{\text{MPE}} = C(\boldsymbol{x}^*) = \max_{\boldsymbol{X}} \bigotimes_{i=0}^{r-1} f_i(\boldsymbol{X}_{S_i})$$

WCSP Weighted Constraint Satisfaction Problem (MPE in negative log-space)

$$C^*_{\text{WCSP}} := C(\boldsymbol{x}^*) = \min_{\boldsymbol{X}} \bigotimes_{i=0}^{r-1} f_i(\boldsymbol{X}_{S_i})$$

Definition

The *primal graph* of a model is a graph where the vertices are the variables and edges connect variables within the same scope

Scopes

- $X_{S_0} = \{A, B\}$
- $X_{S_1} = \{A, C\}$
- $X_{S_2} = \{C, D\}$
- $X_{S_3} = \{B, D\}$
- $X_{S_4} = \{B, F\}$
- $X_{S_5} = \{E, F\}$

Figure: Primal

AND/OR Search Graphs
Primal Graph

Definition

The *primal graph* of a model is a graph where the vertices are the variables and edges connect variables within the same scope

Scopes

- $X_{S_0} = \{A, B\}$
- $X_{S_1} = \{A, C\}$
- $X_{S_2} = \{C, D\}$
- $X_{S_3} = \{B, D\}$
- $X_{S_4} = \{B, F\}$
- $X_{S_5} = \{E, F\}$

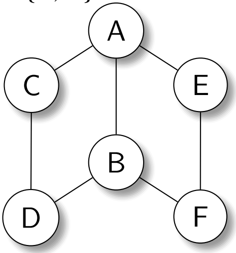
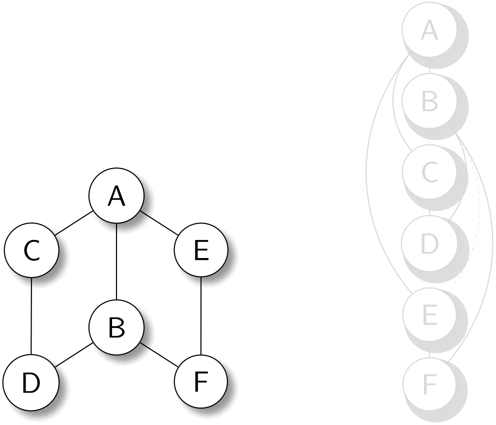


Figure: Primal

AND/OR Search Graphs
Pseudotrees

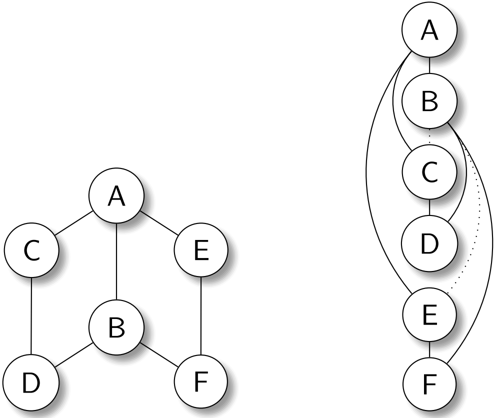


(a) Primal (b) Induced Graph

Figure: Induced graph over the natural ordering.

AND/OR Search Graphs

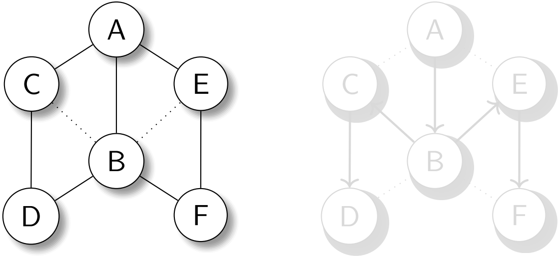
Pseudotrees



(a) Primal (b) Induced Graph

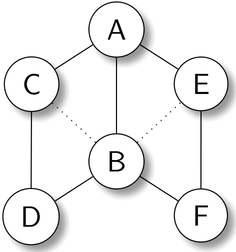
Figure: Induced graph over the natural ordering.

AND/OR Search Graphs
Pseudotrees

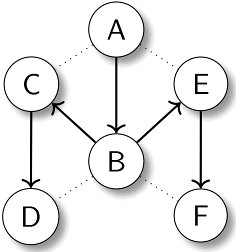


(a) Orig + Ind. Edges (b) Pseudo-tree

Figure: Pseudo-tree with edges chosen to respect the order



(a) Orig + Ind. Edges



(b) Pseudo-tree

Figure: Pseudo-tree with edges chosen to respect the order

AND/OR Search Graphs

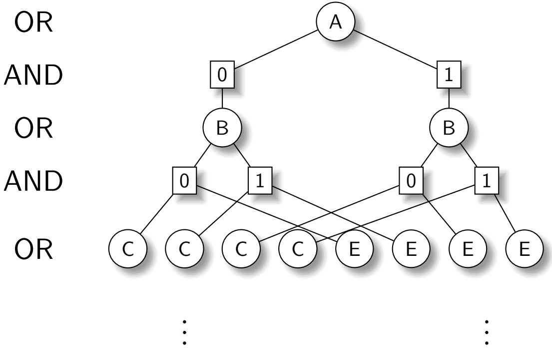


Figure: Context-Minimal AND/OR Graph For Pseudotree

Assume a graphical model $\mathcal{M} = \langle \boldsymbol{X}, \boldsymbol{D}, \boldsymbol{F}, \bigotimes \rangle$ with primal graph G, pseudotree $\mathcal{T}$, and AND/OR search tree $\mathcal{S}_{\mathcal{T}}$

Definition

The *context-minimal AND/OR search graph*, denoted $C_{\mathcal{T}}$, is the AND/OR search graph obtained after merging all identical subproblems.

$C_{\mathcal{T}}$ is exponential in the depth of $\mathcal{T}$

Assume a graphical model $\mathcal{M} = \langle \boldsymbol{X}, \boldsymbol{D}, \boldsymbol{F}, \bigotimes \rangle$ with primal graph G, pseudotree $\mathcal{T}$, and AND/OR search tree $\mathcal{S}_{\mathcal{T}}$

Definition

The *context-minimal AND/OR search graph*, denoted $C_{\mathcal{T}}$, is the AND/OR search graph obtained after merging all identical subproblems.

$C_{\mathcal{T}}$ is exponential in the depth of $\mathcal{T}$

Assume a graphical model $\mathcal{M} = \langle \boldsymbol{X}, \boldsymbol{D}, \boldsymbol{F}, \bigotimes \rangle$ with primal graph G, pseudotree $\mathcal{T}$, and AND/OR search tree $\mathcal{S}_{\mathcal{T}}$

Definition

A *solution tree* T of $C_{\mathcal{T}}$ is a subtree satisfying the following conditions:

1. It contains the root of $C_{\mathcal{T}}$
2. If an internal AND node n is in T, then all the children of n are in T
3. if an internal OR node n is in T, then exactly one child of n is in T
4. Every leaf in T is a terminal node

Assume a graphical model $\mathcal{M} = \langle \boldsymbol{X}, \boldsymbol{D}, \boldsymbol{F}, \bigotimes \rangle$ with primal graph G, pseudotree $\mathcal{T}$, and AND/OR search tree $\mathcal{S}_{\mathcal{T}}$

Definition

A *solution tree* T of $C_{\mathcal{T}}$ is a subtree satisfying the following conditions:

1. It contains the root of $C_{\mathcal{T}}$
2. If an internal AND node n is in T, then all the children of n are in T
3. if an internal OR node n is in T, then exactly one child of n is in T
4. Every leaf in T is a terminal node

Assume a graphical model $\mathcal{M} = \langle \mathbf{X}, \mathbf{D}, \mathbf{F}, \bigotimes \rangle$ with primal graph G, pseudotree $\mathcal{T}$, and AND/OR search tree $\mathcal{S}_{\mathcal{T}}$

Definition

A *solution tree* T of $C_{\mathcal{T}}$ is a subtree satisfying the following conditions:

1. It contains the root of $C_{\mathcal{T}}$
2. If an internal AND node n is in T, then all the children of n are in T
3. if an internal OR node n is in T, then exactly one child of n is in T
4. Every leaf in T is a terminal node

Assume a graphical model $\mathcal{M} = \langle \mathbf{X}, \mathbf{D}, \mathbf{F}, \bigotimes \rangle$ with primal graph G, pseudotree $\mathcal{T}$, and AND/OR search tree $\mathcal{S}_{\mathcal{T}}$

Definition

A *solution tree* T of $C_{\mathcal{T}}$ is a subtree satisfying the following conditions:

1. It contains the root of $C_{\mathcal{T}}$
2. If an internal AND node n is in T, then all the children of n are in T
3. if an internal OR node n is in T, then exactly one child of n is in T
4. Every leaf in T is a terminal node

AND/OR Best First Search

State-of-the-art A^* for AND/OR search space.

Too complicated to fit on a slide

Highlights

Input:
- Graphical Model $\mathscr{M} = \langle \boldsymbol{X}, \boldsymbol{D}, \boldsymbol{F}, \Sigma \rangle$
- Initial weight w_0
- Pseudotree $\mathscr{T}$ rooted at X_1
- heuristic h_i (precalculated)

Output: Optimal solution to $\mathscr{M}$

AND/OR Best First Search

State-of-the-art A^* for AND/OR search space.
Too complicated to fit on a slide

Highlights

Input:
- Graphical Model $\mathscr{M} = \langle \boldsymbol{X}, \boldsymbol{D}, \boldsymbol{F}, \Sigma \rangle$
- Initial weight w_0
- Pseudotree $\mathscr{T}$ rooted at X_1
- heuristic h_i (precalculated)

Output: Optimal solution to $\mathscr{M}$

AND/OR Best First Search

State-of-the-art A^* for AND/OR search space.
Too complicated to fit on a slide

Highlights

Input:
- Graphical Model $\mathscr{M} = \langle \boldsymbol{X}, \boldsymbol{D}, \boldsymbol{F}, \Sigma \rangle$
- Initial weight w_0
- Pseudotree $\mathscr{T}$ rooted at X_1
- heuristic h_i (precalculated)

Output: Optimal solution to $\mathscr{M}$

AND/OR Best First Search

Example

Let $\mathscr{M} = \{\boldsymbol{X}, \boldsymbol{D}, \boldsymbol{F}, \Sigma\}$ where

- $\boldsymbol{X} = \{A, B, C, D\}$
- $\boldsymbol{D} = \bigcup_{s \in \boldsymbol{X}} \{0, 1\}_s$
- $\boldsymbol{F}$ is given by the following tables:

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

B	C	$f(B,C)$
0	0	3
0	1	2
1	0	2
1	1	1

B	$f(B)$
0	1
1	9

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

AND/OR Best First Search
Example

Scopes

$$F = \{f(A,B), f(B,C), f(A,D), f(B)\}$$

Figure: Primal Graph

AND/OR Best First Search
Example

Scopes

$$\boldsymbol{F} = \{f(A, B), f(B, C), f(A, D), f(B)\}$$

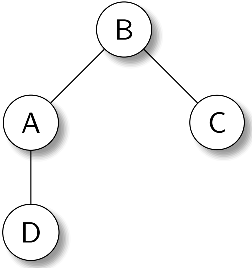
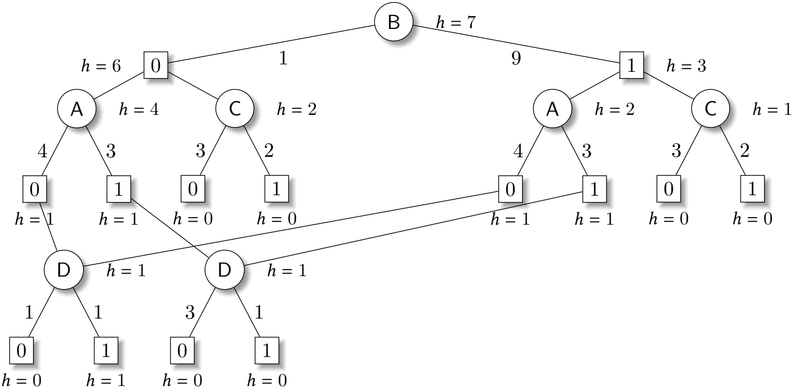


Figure: Primal Graph

AND/OR Best First Search
Example



Algorithm Sketch

Down Pass: Expand nodes and mark terminal nodes solved

Up Pass: Update $v(n)$ for each node according to the following rules:

- OR Nodes: $v(n) = \min_{k \in \mathrm{succ}(n)} w(n,k) + v(k)$
- AND Nodes: $v(n) = \sum_{k \in \mathrm{succ}(n)} v(k)$

AND/OR Best First Search

Example