A Generic Framework for Local Computation

Marc Pouly

marc.pouly@unifr.ch

Department of Informatics
University of Fribourg, Switzerland

Private PhD Defense

7th of May 2008

Outline

1. Valuation Algebras
   - Valuation Algebras & Neutral Elements
   - Projection Problems
   - Valuation Algebra Instances
2. Generic Constructions
   - Semiring Valuation Algebras
   - Projection Problems with Semiring VAs
   - Set-Based Semiring Valuation Algebras
3. Optimization Problems
   - Optimization Problems & Solution Configurations
   - Dynamic Programming
   - Compiling Solution Configurations
4. Distributed Local Computation
   - Distributed Knowledgebases
   - Minimizing Communication Costs

Valuation Algebras & Neutral Elements

Variables: $X, Y, \ldots \in r$

Domains: sets of variables $s \in \mathcal{P}(r)$

Valuations: $\phi, \psi, \ldots \in \Phi$

Operations: labeling $d(\phi)$, combination $\phi \otimes \psi$, marginalization $\phi^{\downarrow s}$

Axioms: 6 axioms (traditional system with Neutral Elements)

Contribution (Schneuwly, Pouly, Kohlas)

- Introduction of VAs without Neutral Elements
- Adjoin an identity element to ensure equal computational possibilities
- VA properties are kept under this extension

Valuation Algebras without Neutral Elements

Algebraic Consequences:

- Only 5 (more general) axioms
- Includes formalisms without neutral elements $\rightsquigarrow$ densities

Computational Consequences:

- Simplifies treatment of formalisms with neutral elements that cannot be represented $\rightsquigarrow$ relations
- Better local computation performance when multiple queries are processed

Projection Problem

Projection problems state the computational interest in VAs:

- Knowledgebase: $\{\phi_1, \ldots, \phi_n\}$, Queries: $\{x_1, \ldots, x_s\}$

$$(\phi_1 \otimes \cdots \otimes \phi_n)^{\downarrow x_i}$$

- $\phi = \phi_1 \otimes \ldots \otimes \phi_n$ is called objective function
- ϕ cannot be computed explicitly $\rightsquigarrow$ complexity concerns
- LC methods confine the domains of all intermediate factors

Distance Functions: A new VA instance

Contribution (Pouly, de Groote)

- Introduction of Distance Functions as Valuation Algebra
- Shortest Path Routing as Projection Problem

- Set of graph vertices r
- $\phi : s \times s \rightarrow \mathbb{N} \cup \{0, \infty\}$
- $\phi(X, Y)$ shortest distance from X to Y
- $\otimes$: shortest distances over factor domains
- $\downarrow$: forgetting of distances

Identifying new VA instances

How to identify new VA instances ?

- Verify the axioms for a given formalisms
- Generic constructions based on semirings

Definition (Commutative Semiring)

Algebraic structure with two operations $+$ and $\times$ over a set A

- $+$ and $\times$ are associative & commutative
- $\times$ distributes over $+$: $a \times (b + c) = (a \times b) + (a \times c)$

Semiring Valuations

- We consider only variables X with finite frame Ω_X
- Ω_s denotes the configuration set of a set of variables s
- By convention: $\Omega_\emptyset = \{\diamond\}$
- Semiring Valuation: $\phi : \Omega_s \to A$
- Operations:

$$\phi \otimes \psi(\mathbf{x}) = \phi(\mathbf{x}^{\downarrow d(\phi)}) \times \psi(\mathbf{x}^{\downarrow d(\psi)})$$

$$\phi^{\downarrow t}(\mathbf{x}) = \sum_{\mathbf{y} \in s-t} \phi(\mathbf{x}, \mathbf{y})$$

Semiring Valuation Algebras

Theorem (Kohlas, Wilson)

This construction induces a VA for every commutative semiring

- Arithmetic Semiring $\langle \mathbb{R}, +, \cdot \rangle \rightsquigarrow$ Probability Potentials
- Boolean Semiring $\langle \{0, 1\}, \max, \min \rangle \rightsquigarrow$ Indicator Functions
- Lattice $\langle \mathbb{N}, \text{lcm}, \text{gcd} \rangle \rightsquigarrow$?

The semiring view allows to identify projection problems directly

Projection Problems with Semiring VAs

$$(\phi_1 \otimes \cdots \otimes \phi_n)^{\downarrow x_i} = \sum_{d(\phi)-x_i} \phi_1(\mathbf{x}^{\downarrow s_1}) \times \cdots \times \phi_n(\mathbf{x}^{\downarrow s_n})$$

Using this formula as pattern, we detected the following projection problems:

Contribution (Pouly, Aji, McEliece)

- Discrete Fourier Transform $\rightsquigarrow$ Apply LC for the Fast Fourier Transform
- Discrete Cosine Transform $\rightsquigarrow$ Apply LC for the Fast Cosine Transform
- Hadamard Transform $\rightsquigarrow$ Apply LC for the Fast Hadamard Transform
- Linear Decoding $\rightsquigarrow$ Apply LC for the Gallager-Tanner-Wiberg Algorithm

Set-Based Semiring Valuation Algebras

Contribution (Pouly)

A second generic construction motivated by Dempster-Shafer theory

- Set-based Semiring Valuation: $\phi : \mathcal{P}(\Omega_s) \rightarrow A$
- Operations:

$$(\phi \otimes \psi)(A) = \sum_{B \bowtie C = A} \phi(B) \times \psi(C)$$

$$\phi^{\downarrow t}(A) = \sum_{\pi_t(B) = A} \phi(B)$$

Ordered Semirings

A semiring is called idempotent, if $a + a = a$ for all $a \in A$

Idempotent semirings allow to define a partial order

$$a \leq_A b \text{ if and only if } a + b = b$$

$$\boxed{0 \leq_A a \leq_A a + b = \sup\{a, b\}}$$

If $\leq_A$ is total:

$$\boxed{0 \leq_A a \leq_A a + b = \max\{a, b\}}$$

Optimization as Projection Problem

In a totally ordered semiring with idempotent $+$:

$$\phi^{-Y}(\mathbf{x}) \;=\; \phi(\mathbf{x}, y) + \phi(\mathbf{x}, \overline{y}) = \max\{\phi(\mathbf{x}, y), \phi(\mathbf{x}, \overline{y})\}$$

$$\phi^{\downarrow t}(\mathbf{x}) \;=\; \sum_{\mathbf{y} \in \Omega_{s-t}} \phi(\mathbf{x}, \mathbf{y}) \;=\; \max\{\phi(\mathbf{x}, \mathbf{y}),\ \mathbf{y} \in \Omega_{s-t}\}$$

and in particular for $x = \emptyset$:

$$\boxed{\phi^{\downarrow \emptyset}(\diamond) \;=\; \max\{\phi(\mathbf{x}),\ \mathbf{x} \in \Omega_s\}}$$

We are not only interested in the optimum value $\phi^{\downarrow \emptyset}(\diamond)$ but also in the configurations that adopt this value $\rightsquigarrow$ solution configurations

Solution Configurations

Some Applications:

- Semiring $\langle [0,1], \max, \cdot \rangle \rightsquigarrow$ Most probable Explanation
- Semiring $\langle [0,1], \min, \cdot \rangle \rightsquigarrow$ Least probable Explanation
- Semiring $\langle \mathbb{N}_0, \min, + \rangle \rightsquigarrow$ Max SAT

Contribution (Pouly)

- Shenoy speaks about optimization problems when $\downarrow$ corresponds to min or max. We give more proper requirements using the semiring view
- Shenoy proposes a dynamic programming algorithm based on fusion to compute solution configurations. We adapt this algorithm to collect

Dynamic Programming

- Can we find all solution configurations ?
- Answer: Yes, if collect and distribute are executed
- And with collect only ?

Contribution (Pouly)

It is possible in case of strict monotonicity.

An idempotent semiring is called strictly monotonic if for $c \neq \mathbf{0}$:

$$a <_{id} b \quad \Rightarrow \quad a \times c <_{id} b \times c$$

Example: $\langle \mathbb{N}_0, \min, \max \rangle$ is not strictly monotonic

Limitations of Dynamic Programming

The dynamic programming approach aims at the enumeration of solution configurations. This is sometimes inappropriate:

- counter-solution configurations
- solution counting
- validity check ...

Contribution (Pouly, Haenni, Wachter)

A new algorithm expresses solution configurations as models of a Boolean function, compiled into a convenient representation to allow fast querying

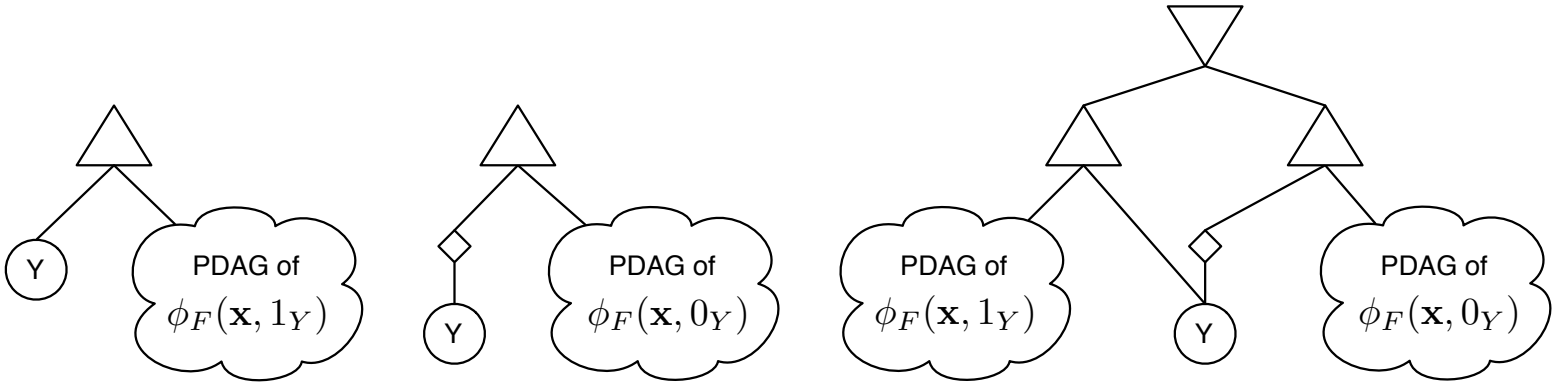
Memorizing Semiring Valuations I

- Totally ordered, idempotent and strictly monotonic semiring
- $\mathcal{F}_r$ denotes the set of Boolean functions over variables in r
- Memorizing Semiring Valuation: $\phi : \Omega_s \to A \times \mathcal{F}_r$
- $\mathbf{x} \mapsto \left(\phi_A(\mathbf{x}), \phi_F(\mathbf{x})\right)$: semiring value and memory
- Operations:

$$(\phi \otimes \psi)(\mathbf{x}) = \left(\phi_A(\mathbf{x}^{\downarrow d(\phi)}) \times \psi_A(\mathbf{x}^{\downarrow d(\psi)}),\ \phi_F(\mathbf{x}^{\downarrow d(\phi)}) \wedge \psi_F(\mathbf{x}^{\downarrow d(\psi)})\right)$$

Memorizing Semiring Valuations II

$$\phi^{-Y}(\mathbf{x}) = \Big(\phi_A(\mathbf{x}, 0_Y) + \phi_A(\mathbf{x}, 1_Y), \quad \ldots \quad \Big)$$



$\phi_A(\mathbf{x}, 0_Y) < \phi_A(\mathbf{x}, 1_Y)$ $\quad$ $\phi_A(\mathbf{x}, 0_Y) > \phi_A(\mathbf{x}, 1_Y)$ $\quad$ $\phi_A(\mathbf{x}, 0_Y) = \phi_A(\mathbf{x}, 1_Y)$

The Algorithm

Contribution (Pouly)

Memorizing Semiring Valuations form a Valuation Algebra

Compilation Algorithm for Solution Configurations:

1. Input: Knowledgebase $\{\phi_1, \ldots, \phi_n\}$
2. Initialization: $\tilde{\phi}_i(\mathbf{x}) = (\phi_i(\mathbf{x}), \top)$
3. Execute fusion/collect with empty query on $\{\tilde{\phi}_1, \ldots, \tilde{\phi}_n\}$
4. Extract $\tilde{\phi}_F^{\downarrow\emptyset}(\diamond)$ from the result of the fusion/collect algorithm

Theorem (Pouly)

$$\textbf{Models}\left(\tilde{\phi}_F^{\downarrow\emptyset}(\diamond)\right) = c_\phi$$

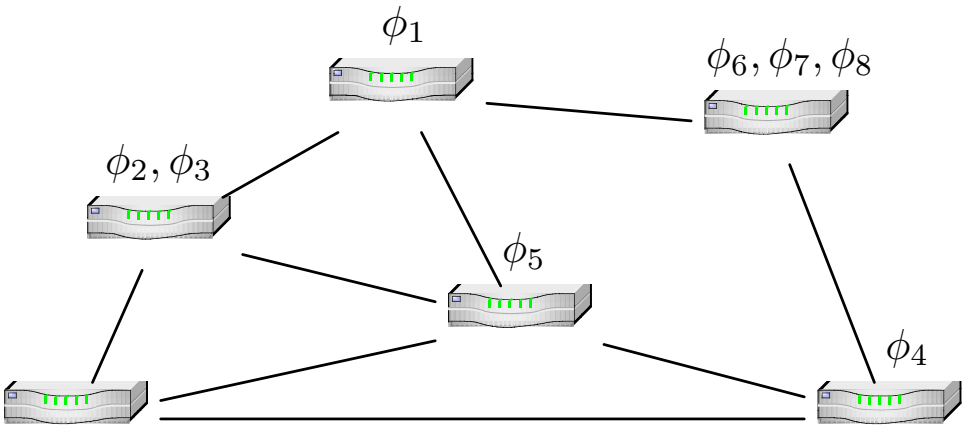
Efficient Querying

- $\tilde{\phi}_F^{\downarrow\emptyset}(\diamond)$ is a d-DNNF (= cdn-PDAG)
- Model enumeration on d-DNNF is polytime (Darwiche)
- Query table of PDAG languages: (Haenni, Wachter)

	CO/CE/ME	VA/IM/ME^c	CT/PR/PEQ	EQ	SE
PDAG	○	○	○	○	○
f-PDAG	○	○	○	○	○
cd-PDAG	√	√	√	?	○
dn-PDAG / d-NNF	○	○	○	○	○
cn-PDAG / DNNF	√	○	○	○	○
cdn-PDAG / d-DNNF	√	√	√	?	○
BDD	○	○	○	○	○
FBDD	√	√	√	?	○
OBDD	√	√	√	√	○
OBDD$_<$	√	√	√	√	√
DNF	√	○	○	○	○
CNF	○	√	○	○	○
PI	√	√	○	√	√
IP	√	√	○	√	√
MODS	√	√	√	√	√

Local Computation with Remote Valuations

- Distributed Knowledgebase: $\chi : \{\phi_1, \ldots, \phi_n\} \rightarrow P$
- Motivation: distributed inference with distributed data
- Communication costs: $c_\phi(i,j) = \omega(\phi) \cdot d_{i,j}$
- Assumption: Costs shrink under $\downarrow$ (otherwise LC makes no sense ...)



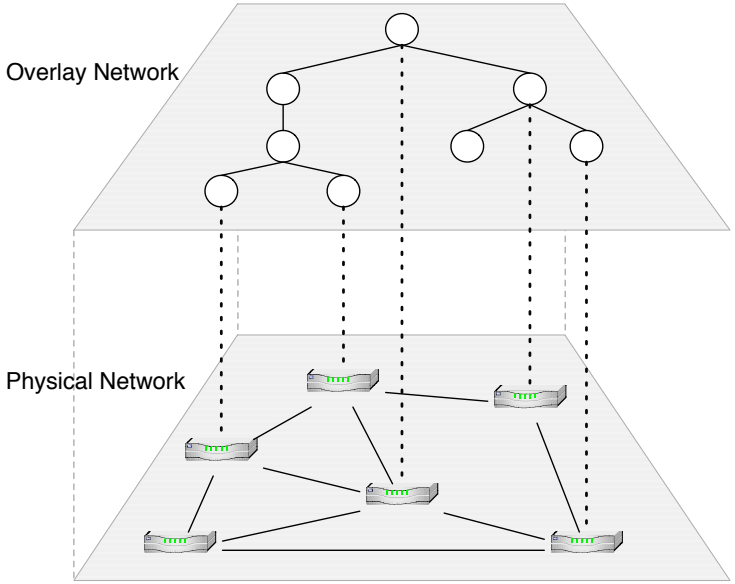
Total Communication Costs

- Identify join tree node with processor that hosts its factor

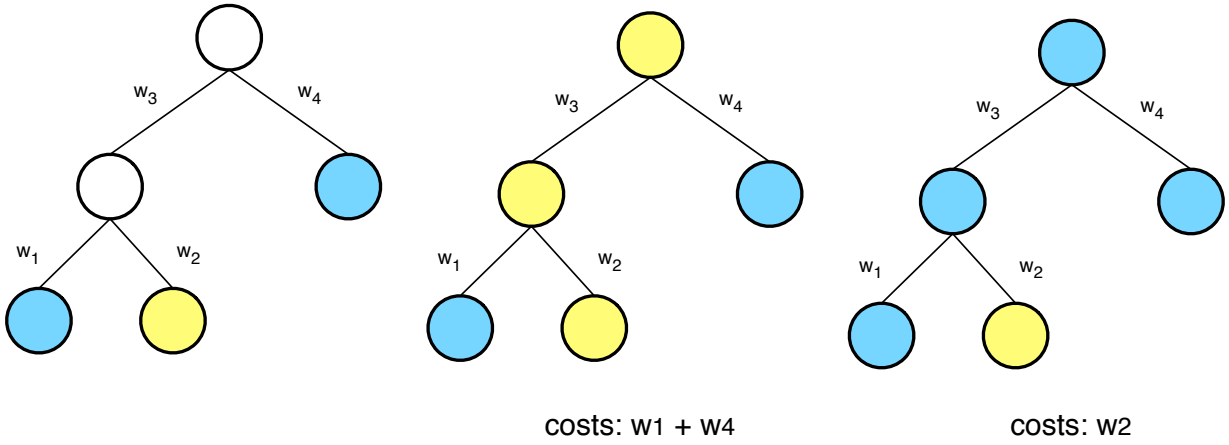
  $\Rightarrow$ Join tree serves as overlay network for distributed computations

  $\Rightarrow$ Total communication costs = costs of all LC messages

- Task: Minimize inter-processor communication !



Minimization of Total Communication Costs



- Requirement: The message costs must be predictable !
- Weight predictability: $f : \mathcal{P}(r) \rightarrow \mathbb{N}_0$ such that $\omega(\phi) = f(d(\phi))$

Contribution (Pouly, Erdös & Szekely)

- On general graphs, this problem is called Multiterminal Cut Problem
- Although NPC in general, there is an $O(|V| \cdot |P|^2)$ algorithm for trees

Discussion