

A GENERIC APPROACH FOR SPARSE PATH PROBLEMS

Marc Pouly

Cork Constraint Computation Center, University College Cork, Ireland

Keywords: Sparse path problems, Valuation algebras, Local computation, Tree-decomposition methods.

Abstract: This paper shows how sparse path problems can be solved by tree-decomposition techniques. We analyse the properties of closure matrices and prove that they satisfy the axioms of a valuation algebra, which is known to be sufficient for the application of generic tree-decomposition methods. Given a sparse path problem where only a subset of queries are required, we continually compute path weights of smaller graph regions and deduce the total paths from these results. The decisive complexity factor is no more the total number of graph nodes but the induced treewidth of the path problem.

1 INTRODUCTION

In recent years, a large number of formalisms for automated inference have been proposed. Typical examples are: probability potentials from Bayesian networks, Dempster-Shafer theory, different constraint systems and logics, Gaussian potentials and density functions, relational algebra, possibilistic formalisms and many more. Inference based on these formalisms is a computationally hard task which motivated the introduction of tree-decomposition methods. But it also turned out that they all share some common algebraic properties which are pooled in the *valuation algebra* framework (Shenoy and Shafer, 1990; Kohlas, 2003). Based on this framework, a collection of generic tree-decomposition methods has been derived. Thus, instead of re-inventing such inference procedures for each different formalism, it is sufficient to verify a small axiomatic system to gain access to efficient generic procedures and implementations. This is known as the *local computation* framework. In parallel to these developments, tree-decomposition methods were successfully applied for the solution of sparse linear systems. For equation systems over fields, it has been shown that these approaches, which aim at the minimization of fill-ins in matrices, are subject to the valuation algebra framework, and that the according tree-decomposition procedures are specializations of the generic local computation methods (Kohlas, 2003; Pouly and Kohlas, 2010). However, many important applications can be reduced to the so-called *algebraic path problem* which requires to solve a fixpoint equation system with values from a semir-

ing. Essentially, there are two approaches which focus on solving sparse fixpoint systems over semirings by tree-decomposition techniques: Similar to the inverse matrix in case of linear systems over fields, the *quasi-inverse* matrix provides a solution to a semiring fixpoint system. Such quasi-inverse matrices always exist for *closed semirings* (Lehmann, 1976) and can be computed by the well-known Floyd-Warshall-Kleene algorithm. (Radhakrishnan et al., 1992) combined this insight with LDU decomposition for semiring matrices (Backhouse and Carré, 1975) to obtain a tree-decomposition algorithm for sparse fixpoint equations over closed and idempotent semirings. Again, this approach is covered by the local computation framework with the fixpoint equations satisfying the valuation algebra axioms. Alternatively, (Chaudhuri and Zaroliagis, 1997) proposed a second method for the particular problem of computing shortest distances. In this paper, we will identify the algebraic requirements of this second method and show that it complies with the valuation algebra framework. This enables the application of existing, generic inference procedures for the solution of sparse path problems. Moreover, we generalize this idea from shortest distances to a wider class of semirings called *Kleene algebras* which further includes other graph related path problems as for example the computation of maximum capacities or reliabilities, and also many other problems that are not directly related to graphs but which can nevertheless be reduced to a path problem. We refer to (Rote, 1990) for an extensive listing of such examples.

2 VALUATION ALGEBRAS

The basic elements of a valuation algebra are so-called *valuations*. Intuitively, a valuation can be regarded as a representation of knowledge about the possible values of a set r of variables. It can be said that each valuation ϕ refers to a finite set of variables $d(\phi) \subseteq r$ called its *domain*. Further, let D be the power set of r and Φ a set of valuations with their domains in D . We assume the following operations in (Φ, D) :

- *Labeling*: $\Phi \rightarrow D; \phi \mapsto d(\phi)$,
- *Combination*: $\Phi \times \Phi \rightarrow \Phi; (\phi, \psi) \mapsto \phi \otimes \psi$,
- *Projection*: $\Phi \times D \rightarrow \Phi; (\phi, x) \mapsto \phi^{\downarrow x}$ for $x \subseteq d(\phi)$.

We further impose the following axioms on Φ and D :

1. *Commutative Semigroup*: Combination is associative and commutative.
2. *Labeling*: For $\phi, \psi \in \Phi$, $d(\phi \otimes \psi) = d(\phi) \cup d(\psi)$.
3. *Projection*: For $\phi \in \Phi$ and $x \subseteq d(\phi)$, $d(\phi^{\downarrow x}) = x$.
4. *Transitivity*: For $\phi \in \Phi$ and $x \subseteq y \subseteq d(\phi)$,

$$(\phi^{\downarrow y})^{\downarrow x} = \phi^{\downarrow x}.$$

5. *Combination*: For $\phi, \psi \in \Phi$ with $d(\phi) = x$, $d(\psi) = y$ and $z \in D$ such that $x \subseteq z \subseteq x \cup y$,

$$(\phi \otimes \psi)^{\downarrow z} = \phi \otimes \psi^{\downarrow z \cap y};$$

6. *Domain*: For $\phi \in \Phi$ with $d(\phi) = x$, $\phi^{\downarrow x} = \phi$.
7. *Idempotency*: For $\phi \in \Phi$ and $x \subseteq d(\phi)$, $\phi \otimes \phi^{\downarrow x} = \phi$.

These axioms require natural properties regarding knowledge modelling. The first axiom indicates that if knowledge comes in pieces, the sequence does not influence their combination. The labeling axiom tells us that the combination of valuations gives knowledge over the union of the involved variables. Transitivity says that projection can be performed in several steps, and the combination axiom states that we either combine a new piece to the already given knowledge and focus afterwards to the desired domain, or we first cut the uninteresting parts of the new knowledge out and combine it afterwards. The domain axiom expresses that trivial projection has no effect and finally, idempotency states that combining a piece of knowledge with a part of itself gives nothing new.

Definition 1. A system (Φ, D) satisfying the axioms 1 to 6 is called a valuation algebra. If axiom 7 holds, then it is called an idempotent valuation algebra.

A listing of formalisms that adopt the structure of a valuation algebra was already given in the introduction. Among them, the valuation algebras of relations and crisp constraints are idempotent. We refer to (Pouly, 2008; Kohlas, 2003) for further examples.

2.1 Generic Inference Problem

The computational interest in valuation algebras is stated by the *inference problem*. Given a set of valuations $\{\phi_1, \dots, \phi_n\} \subseteq \Phi$ called *knowledgebase* and a set of *queries* $x = \{x_1, \dots, x_s\}$, the inference problem consists in computing

$$\phi^{\downarrow x_i} = (\phi_1 \otimes \dots \otimes \phi_n)^{\downarrow x_i} \quad (1)$$

for $1 \leq i \leq s$. For example, if the knowledgebase consists of the CPTs from a Bayesian network, then the inference problem reflects the computation of marginals from the join probability distribution. If the knowledgebase models a constraint system, then the inference problem with the empty query corresponds to satisfiability, if the knowledgebase contains relations, then the inference problem mirrors query answering in relational databases.

The complexity of combination and projection generally depends on the size of the factor domains and often shows an exponential behaviour. According to axiom 2 and 3, the domains of valuations grow under combination and shrink under projection. Efficient inference algorithms must therefore confine in some way the size of intermediate results, which can be achieved by alternating the operations of combination and projection. This is the promise of local computation. The valuation algebra axioms are sufficient for the definition of general local computation procedures which solve the inference problem independently of the underlying formalism. These algorithms include *fusion* (Shenoy, 1992) and *bucket elimination* (Dechter, 1999) for inference problems with a single query, and the Shenoy-Shafer architecture (Shafer and Shenoy, 1988) for multiple queries. If (some weaker condition of) axiom 7 is present, other local computation architectures with a more efficient scheduling of computations can be derived (Lauritzen and Spiegelhalter, 1988; Jensen et al., 1990; Kohlas, 2003).

3 LOCAL COMPUTATION

Local computation methods are usually described as message-passing schemes on *covering join trees* (also called *tree-decomposition*):

Definition 2. A join tree is a labeled tree (V, E, λ, D) whose labeling function $\lambda : V \rightarrow D$ satisfies the running intersection property, i.e. for two nodes $v_1, v_2 \in V$, if $X \in \lambda(v_1) \cap \lambda(v_2)$, then X is contained in every node label on the unique path between v_1 and v_2 . A join tree covers the inference problem if

- for each factor ϕ_i , there is a node $v \in V$ such that $d(\phi_i) \subseteq \lambda(v)$,

- for each query x_i , there is a node $v \in V$ such that $x_i \subseteq \lambda(v)$.

A detailed description of all local computation methods can be found in (Pouly, 2008). Here, we only cite the main theorem that all multi-query procedures have in common:

Theorem 1. *At the end of the message-passing, each node i contains $\phi^{\downarrow \lambda(i)}$.*

If the query x_j is covered by some node $i \in V$, we obtain the query answer as a consequence of the transitivity axiom by one last projection

$$\phi^{\downarrow x_j} = \left(\phi^{\downarrow \lambda(i)} \right)^{\downarrow x_j}. \quad (2)$$

In local computation methods, all computations take place in the join tree nodes. The domains of intermediate factors, which determine the complexity of combination and projection, are therefore bounded by the largest node label in the join tree. This measure is called *treewidth*. In other words, the smaller the treewidth, the more efficient is local computation. Finding a covering join tree with a minimum treewidth is NP-complete (Arnborg et al., 1987). But there are good heuristics (Lehmann, 2001).

4 ALGEBRAIC PATH PROBLEM

The algebraic path problem aims at the unification of various path problems in terms of the solution of a generic fixpoint equation with values from a semiring.

Definition 3. *A tuple $\langle E, +, \times, \mathbf{0}, \mathbf{1} \rangle$ with binary operations $+$ and \times is called semiring if:*

- $+$ and \times are associative and $+$ is commutative;
- for $a, b, c \in E$: $a \times (b + c) = a \times b + a \times c$;
- for $a, b, c \in E$: $(a + b) \times c = a \times c + b \times c$;
- $+$ and \times have neutral elements $\mathbf{0}$ and $\mathbf{1}$;
- $a \times \mathbf{0} = \mathbf{0} \times a = \mathbf{0}$ for all $a \in E$.

If the semiring is *idempotent*, satisfying $a + a = a$ for all $a \in E$, then the following relation is a partial order:

$$a \leq b \quad \text{if, and only if} \quad a + b = b. \quad (3)$$

Typical examples of idempotent semirings are the Boolean semiring $\langle \{0, 1\}, \max, \min, 0, 1 \rangle$, the tropical semiring $\langle \mathbb{N} \cup \{0, \infty\}, \min, +, \infty, 0 \rangle$, the arctic semiring $\langle \mathbb{R} \cup \{-\infty\}, \max, +, -\infty, 0 \rangle$, the probabilistic semiring $\langle [0, 1], \max, \cdot, 0, 1 \rangle$ and the bottleneck semiring $\langle \mathbb{R} \cup \{+\infty, -\infty\}, \max, \min, -\infty, \infty \rangle$.

For $n \in \mathbb{N}$ we next consider the set of $n \times n$ matrices $M \in \mathcal{M}(E, n)$ with values from an idempotent

semiring E . This set forms itself an idempotent semiring. We define the power sequence of matrices:

$$M^{(r)} = I + M + M^2 + \dots + M^r. \quad (4)$$

If we interpret M as the adjacency matrix of a graph with edge weights from the tropical semiring, then $M^{(r)}$ corresponds to the shortest distances containing at most r edges. Consequently, we obtain the shortest distances between each pair of graph nodes by

$$\bigoplus_{r \geq 0} M^r = I + M + M^2 + \dots \quad (5)$$

Observe that this infinite sum is not always defined. But if the above sequence converges with a suitable notion of topology (Gondran and Minoux, 2008), it can be shown that its limit M^* always satisfies

$$M^* = MM^* + I = M^*M + I. \quad (6)$$

This motivates the following definition.

Definition 4. *The algebraic path problem consists in solving the fixpoint equation $X = MX + I = XM + I$.*

There may be no solution, one solution or infinitely many solutions to this equation. However, for computational purposes it is often convenient to avoid this difficulty by ensuring the existence of a solution axiomatically. Such semirings are called *closed semirings* (Lehmann, 1976) and they provide for each $a \in E$ an element $a^* \in E$ such that $a^* = aa^* + \mathbf{1} = a^*a + \mathbf{1}$. Moreover, given a matrix M with values from a closed semiring, it is possible to compute M^* inductively from the values of the underlying semiring:

- For $n = 1$ we define $(a)^* = (a^*)$.
- For $n > 1$ we decompose the matrix M into submatrices B, C, D, E such that B and E are square and define

$$\begin{pmatrix} B & C \\ D & E \end{pmatrix}^* = \begin{pmatrix} B^* + B^*CF^*DB^* & B^*CF^* \\ F^*DB^* & F^* \end{pmatrix} \quad (7)$$

where $F = E + DB^*C$.

M^* as a result of this construction is a solution to the algebraic path problem (Lehmann, 1976). In other words, matrices over closed semirings themselves form a closed semiring. The proof of this statement affords the Floyd-Warshall-Kleene algorithm which performs this task in time $O(n^3)$. But to derive a valuation algebra, we first introduce an even more structured class of semirings called *Kleene algebras* which are closed and idempotent semiring with an additional monotonicity property:

Definition 5. *A tuple $\langle E, +, \times, *, \mathbf{0}, \mathbf{1} \rangle$ with an unary operation $*$ is called Kleene algebra if:*

1. $\langle E, +, \times, \mathbf{0}, \mathbf{1} \rangle$ is an idempotent semiring;

2. $a^* = \mathbf{1} + a^*a = \mathbf{1} + aa^*$ for $a \in E$;
3. $ax \leq x$ implies that $a^*x \leq x$ for $a, x \in E$;
4. $xa \leq x$ implies that $xa^* \leq x$ for $a, x \in E$.

For example, the Boolean semiring is a Kleene algebra with $0^* = 1^* = 1$. The tropical semiring of non-negative integers is a Kleene algebra with $a^* = 0$ for all $a \in \mathbb{N} \cup \{0, \infty\}$. The arctic semiring is a Kleene algebra with $a^* = 0$ for $a \geq 0$ and $a^* = -\infty$ otherwise. The probabilistic semiring is a Kleene algebra with $a^* = 1$ for $a \in [0, 1]$, and the bottleneck semiring is a Kleene algebra with $a^* = \infty$ for all $a \in \mathbb{R} \cup \{-\infty, \infty\}$.

Kleene algebras have many interesting properties. Most important among them are the closure properties: $a \leq a^*$, $a^{**} = a^*$ and $a \leq b$ implies that $a^* \leq b^*$, but also $\mathbf{1} = \mathbf{1}^* \leq a^*$ and

$$(a+b)^* = (a^*+b)^* = (a^*+b^*) \quad (8)$$

for all $a, b \in E$. We refer to (Kozen, 1994) for the proofs of these and other elementary properties.

Since Kleene algebras are closed semirings, we may compute M^* of a matrix M with values from a Kleene algebra by the same algorithm and the result again satisfies monotonicity (Conway, 1971). This proves that matrices over Kleene algebras themselves form a Kleene algebra, and as a further implication of the monotonicity law, it confirms that computing M^* over the tropical semiring indeed gives shortest distances. Accordingly, the arctic semiring gives maximum capacities, the probabilistic semiring maximum reliabilities and the Boolean semiring connectivities.

5 KLEENE VALUATIONS

We prove in this section that closures matrices satisfy the valuation algebra axioms. Let $r = \{X_1, \dots, X_n\}$ be a finite set of variables and D its powerset. For a Kleene algebra $\langle E, +, \times, *, \mathbf{0}, \mathbf{1} \rangle$ and $s \in D$, we consider *labeled matrices* $M : s \times s \rightarrow E$ and refer to $d(M) = s$ as their domain. We then write $\mathcal{M}(E, s)$ for the set of all labeled matrices with domain $s \in D$ and also define the set of all closures of labeled matrices with domain in D as $\Phi = \{M^* | M \in \mathcal{M}(E, s) \text{ and } s \in D\}$. We next introduce some operations in Φ starting with the projection. For $M^* \in \Phi$, $t \subseteq d(M^*)$ and $X, Y \in t$,

$$(M^*)^{\downarrow t}(X, Y) = M^*(X, Y). \quad (9)$$

This simply corresponds to the restriction of the matrix M^* to the variables in t . It is easy to prove that Φ is closed under projection, i.e. the restriction of a closure matrix again results in a closure matrix. Intuitively, considering a subgraph of a graph with shortest distances still contains shortest distances. Clearly,

the projection operator satisfies transitivity. We next introduce the *direct sum* of labeled matrices: Let $M_1 \in \mathcal{M}(E, s)$ and $M_2 \in \mathcal{M}(E, t)$ with $s \cap t = \emptyset$ and $X, Y \in s \cup t$, we define

$$(M_1 \oplus M_2)(X, Y) = \begin{cases} M_1(X, Y) & \text{if } X, Y \in s, \\ M_2(X, Y) & \text{if } X, Y \in t, \\ \mathbf{0} & \text{otherwise.} \end{cases}$$

It follows from the inductive definition of M^* that the closure operation distributes over the direct sum, i.e.

$$(M_1 \oplus M_2)^* = M_1^* \oplus M_2^*. \quad (10)$$

This allows us to define an operation of vacuous extension for $M \in \mathcal{M}(E, s)$ and $s \subseteq t$ by $M^{\uparrow t} = M \oplus I_{t-s}$. The application of the closure operation and vacuous extension are interchangeable. It follows directly from (10) and $\mathbf{1}^* = \mathbf{1}$ that

$$(M^{\uparrow t})^* = (M \oplus I_{t-s})^* = (M^* \oplus I_{t-s}) = (M^*)^{\uparrow t}. \quad (11)$$

Thus, Φ is also closed under vacuous extension. Finally, we introduce a very intuitive combination rule for elements in Φ . Imagine that we have two closure matrices which express shortest distances in two possibly overlapping regions of a large graph. Then, the shortest distance matrix for the unified region is found by vacuously extending both matrices to their union domain, taking the component-wise minimum which corresponds to semiring addition and computing the new shortest distances. Thus, for $M_1^*, M_2^* \in \Phi$ with $d(M_1^*) = s$ and $d(M_2^*) = t$ we define

$$M_1^* \otimes M_2^* = \left((M_1^*)^{\uparrow s \cup t} + (M_2^*)^{\uparrow s \cup t} \right)^*. \quad (12)$$

We directly conclude from this definition that Φ is closed under combination and also that combination is commutative. Proving associativity is more involved but follows from (11) and (8). Furthermore, this definition of combination also fulfills the combination axiom:

Lemma 1. *If $M_1^*, M_2^* \in \Phi$ with $d(M_1^*) = s$, $d(M_2^*) = t$ and $s \subseteq z \subseteq s \cup t$ we have*

$$(M_1^* \otimes M_2^*)^{\downarrow z} = M_1^* \otimes (M_2^*)^{\downarrow z \cap t}. \quad (13)$$

Decompose $M_1^* \otimes M_2^* = ((M_1^*)^{\uparrow s \cup t} + (M_2^*)^{\uparrow s \cup t})^*$ with respect to z and $t - z$. The statement then follows from (7). Finally, we observe that closure matrices fulfill idempotency. For $M^* \in \Phi$ with $s \subseteq t = d(M^*)$,

$$M^* \otimes (M^*)^{\downarrow s} = \left[M^* + ((M^*)^{\downarrow s})^{\uparrow t} \right]^* = M^{**} = M^*.$$

This follows from the idempotency of addition and the properties $\mathbf{1} \leq a^*$ and $a^{**} = a^*$. Altogether, we proved the following theorem:

Theorem 2. *Closures of Kleene valued matrices with labeling, projection (9) and combination (12) satisfy the axioms of a valuation algebra.*

Every Kleene algebra therefore induces an idempotent valuation algebra of matrix closures. For the particular case of the tropical semiring of non-negative integers, this corresponds to the result of (Chaudhuri and Zaroliagis, 1997) that implicitly used the above properties by referring to Bellmann’s principle of optimality. The final section shows how local computation with valuation algebras of matrix closures are used for the solution of path problems.

6 SOLVING PATH PROBLEMS

Considering decompositions of large graphs or networks is very natural. A typical example is a road map of Europe that is decomposed into smaller road maps for each country. Thus, we assume an adjacency matrix M of a large graph which is decomposed into a set of matrices $\{M_1, \dots, M_n\}$ taking values from a Kleene algebra. These matrices correspond to the adjacency matrices of some smaller graph regions. Representing the nodes of the total graph by the variable set $s = d(M)$ we obtain $M = M_1^{\uparrow s} + \dots + M_n^{\uparrow s}$ and using the properties (8) and (11)

$$M^* = \left(M_1^{\uparrow s} + \dots + M_n^{\uparrow s} \right)^* = M_1^* \otimes \dots \otimes M_n^*.$$

Assume next that we are interested in some specific path weights $M^*(X, Y)$ for variables $X, Y \in s$. These pairs of variables form the query set x and the task consists in computing

$$(M^*)^{\downarrow\{X, Y\}} = (M_1^* \otimes \dots \otimes M_n^*)^{\downarrow\{X, Y\}} \quad (14)$$

for each query $\{X, Y\} \in x$. This defines an inference problem according to Section 2.1 which can be solved by local computation. We first construct a join tree covering all knowledgebase factors M_i^* and all queries $\{X, Y\} \in x$ and then execute a multi-query local computation architecture. At the end of the message-passing, the query answers are obtained from Equation (2). During the local computation process, two messages are sent along each edge, and each message is combined to some node content. For a join tree (V, E, λ, D) we thus perform $2(|V| - 1)$ combinations and $2(|V| - 1)$ projections. Combination is surely the more costly operation since projection only corresponds to matrix restriction. Using the algorithm of Floyd-Warshall-Kleene we obtain the closure M^* of a matrix M in time $O(|d(M)|^3)$. Moreover, the largest matrix domain that occurs during the local computation process is bounded by the treewidth ω .

We obtain $O(|V|\omega^3)$ for the time complexity of computing Equation (14) since only matrices are stored, the space complexity is $O(|V|\omega^2)$. However, there is an important issue regarding the treewidth complexity. When dealing with path problems, people are often interested in a large number of paths. These query sets may either be structured (e.g. single-source problems), or they may be arbitrary sets of queries. But since the join tree must cover all queries, the treewidth may grow significantly if a large number of queries is present. In the worst case, when all possible path weights have to be computed, the join tree consists of a single node containing all variables. Except for such extreme cases, this problem can be addressed by exploiting idempotency. If the valuation algebra is idempotent, it holds that

$$\phi = \bigotimes_{i=1}^m \phi^{\downarrow\lambda(i)}. \quad (15)$$

This was shown by (Kohlas, 2003). The following lemma can be derived from this important result:

Lemma 2. *Let (v_1, \dots, v_k) be a path in the join tree from node v_1 to node v_k with $v_i \in V$ for $1 \leq i \leq k$. We then have for an idempotent valuation algebra*

$$\phi^{\downarrow\lambda(v_1) \cup \dots \cup \lambda(v_k)} = \bigotimes_{i=1}^k \phi^{\downarrow\lambda(v_i)}. \quad (16)$$

For an uncovered query $\{X, Y\}$ we always find two nodes $v_1, v_k \in V$ such that $X \in \lambda(v_1)$ and $Y \in \lambda(v_k)$. For the path (v_1, \dots, v_k) connecting the two nodes, it follows from the transitivity axiom that

$$\phi^{\downarrow\lambda(v_1) \cup \lambda(v_k)} = \left(\bigotimes_{i=1}^k \phi^{\downarrow\lambda(v_i)} \right)^{\downarrow\lambda(v_1) \cup \lambda(v_k)},$$

from which we obtain the query answer by one last projection to $\{X, Y\}$. Computing this formula is yet too expensive. But a clever application of the combination axiom uncovers the following algorithm:

1. initialize $\eta := \phi^{\downarrow\lambda(v_1)}$;
2. repeat for $i = 1 \dots k - 1$

$$\eta := \phi^{\downarrow\lambda(v_{i+1})} \otimes \eta^{\downarrow\lambda(v_1) \cup (\lambda(v_i) \cap \lambda(v_{i+1}))}$$

3. return $\eta^{\downarrow\{X, Y\}}$.

Theorem 3. *The algorithm outputs $\phi^{\downarrow\{X, Y\}}$ for $\{X, Y\} \subseteq \lambda(v_1) \cup \lambda(v_k)$ and $v_1, v_k \in V$.*

The domain of η will never exceed the union of two node labels. Therefore, its time complexity is driven by the double of the treewidth. Also, the number of combinations is bounded by the largest path in the tree with has at most $|V|$ nodes. We obtain for

the time complexity $O(|V| \cdot (2\omega)^3) = O(|V| \cdot \omega^3)$. To sum it up, given a factorized path problem and some query set, we only consider the knowledgebase for the construction of the join tree and ignore the query set. This gives us the smallest treewidth that can be found for this inference problem. After local computation, each node $v \in V$ contains $\phi^{\downarrow\lambda(v)}$ according to Theorem 1. For each query $\{X, Y\} \in x$, we search two nodes $v_1, v_k \in V$ that cover this query $\{X, Y\} \subseteq \lambda(v_1) \cup \lambda(v_2)$ and identify the path between them. Then, the above query answering algorithm is executed. Doing so, all queries in x can be computed with a total time complexity of $O(|x| \cdot |V| \cdot \omega^3)$. It is clear that for the complete query set, we have $|x| = |V|^2/2$ which makes the time complexity of this algorithm worse than the direct computation of M^* . However, by storing intermediate results in the above algorithm, it is possible to reduce the complexity of the all-pairs problem to $O(|V|^2 \cdot \omega^3)$ which corresponds to the construction of an optimum path tree (Pouly and Kohlas, 2010). Thus, for extremely sparse path problems this approach may still be worthwhile. If however only some smaller subset of queries is required, the performance of this algorithm is equal to other sparse matrix techniques which proved their worth in many applications.

7 CONCLUSIONS

We have shown in this article that closure matrices over Kleene algebras always induce a valuation algebra. This uncovers many new and important instances of the local computation framework which can now be studied in this more general setting. It further gives a general and efficient algorithm for the solution of sparse path problems when either only a subset of all queries are of interest, or if a high sparsity rate is present. There is no need to specify the query set in advance. The propagated join tree can thus be considered as the result of a pre-compilation, upon which queries can later be answered in a dynamic way. This approach does not assume any structure in the query set which makes it more generally applicable than other path algorithms. Finally, the query answering algorithm is only based on the properties of idempotent valuation algebras and can thus be applied to other formalisms than matrices over Kleene algebras. This however still deserves closer investigation.

REFERENCES

- Amborg, S., Corneil, D., and Proskurowski, A. (1987). Complexity of finding embeddings in a k-tree. *SIAM J. of Algebraic and Discrete Methods*, 8:277–284.
- Backhouse, R. C. and Carré, B. A. (1975). Regular algebra applied to path-finding problems. *Journal of the Institute for Mathematics and its Applications*, 15:161–186.
- Chaudhuri, S. and Zaroliagis, C. (1997). Shortest paths in digraphs of small treewidth. part i: Sequential algorithms. *Algorithmica*, 27:212–226.
- Conway, J. H. (1971). *Regular Algebra and Finite Machines*. Chapman and Hall Mathematics Series. Chapman and Hall.
- Dechter, R. (1999). Bucket elimination: A unifying framework for reasoning. *Artificial Intelligence*, 113:41–85.
- Gondran, M. and Minoux, M. (2008). *Graphs, Dioids and Semirings: New Models and Algorithms*. Operations Research Computer Science Interfaces Series. Springer Publishing Company, Incorporated.
- Jensen, F., Lauritzen, S., and Olesen, K. (1990). Bayesian updating in causal probabilistic networks by local computation. *Computational Statistics Quarterly*, 4.
- Kohlas, J. (2003). *Information Algebras: Generic Structures for Inference*. Springer-Verlag.
- Kozen, D. (1994). A completeness theorem for kleene algebras and the algebra of regular events. *Information and Computing*, 110(2):366–390.
- Lauritzen, S. L. and Spiegelhalter, D. J. (1988). Local computations with probabilities on graphical structures and their application to expert systems. *J. Royal Statist. Soc. B*, 50:157–224.
- Lehmann, D. J. (1976). Algebraic structures for transitive closure. Technical report, Department of Computer Science, University of Warwick.
- Lehmann, N. (2001). *Argumentation System and Belief Functions*. PhD thesis, Department of Informatics, University of Fribourg.
- Pouly, M. (2008). *A Generic Framework for Local Computation*. PhD thesis, Department of Informatics, University of Fribourg.
- Pouly, M. and Kohlas, J. (2010). *Generic Inference*. John Wiley & Sons, Inc.
- Radhakrishnan, V., Hunt, H., and Stearns, R. (1992). Efficient algorithms for solving systems of linear equations and path problems. In *STACS '92: Proceedings of the 9th Annual Symposium on Theoretical Aspects of Computer Science*, pages 109–119, London, UK. Springer-Verlag.
- Rote, G. (1990). Path problems in graphs. *Computing Suppl.*, 7:155–198.
- Shafer, G. and Shenoy, P. (1988). Local computation in hypertrees. Technical report, University of Kansas.
- Shenoy, P. P. (1992). Valuation-based systems: A framework for managing uncertainty in expert systems. In Zadeh, L. and Kacprzyk, J., editors, *Fuzzy Logic for the Management of Uncertainty*, pages 83–104. John Wiley & Sons.
- Shenoy, P. P. and Shafer, G. (1990). Axioms for probability and belief-function propagation. In *Readings in uncertain reasoning*, pages 575–610. Morgan Kaufmann Publishers Inc.