International Journal of Approximate Reasoning 80 (2017) 377–392

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

www.elsevier.com/locate/ijar

Concise representations and construction algorithms for semi-graphoid independency models

Stavros Lopatatzidis [a,*], Linda C. van der Gaag [b]

[a] Ghent University, SYSTeMS Research Group, Technologiepark-Zwijnaarde 914, 9052 Zwijnaarde, Belgium
[b] Utrecht University, Department of Information and Computing Sciences, Princetonplein 5, 3584 CC Utrecht, The Netherlands

ARTICLE INFO

Article history:
Received 25 November 2015
Received in revised form 22 June 2016
Accepted 24 June 2016
Available online 29 June 2016

Keywords:
Conditional independence
Semi-graphoid axioms
Closure
Closure representation
Dominant independence statements

ABSTRACT

The conditional independencies from a joint probability distribution constitute a model which is closed under the semi-graphoid properties of independency. These models typically are exponentially large in size and cannot be feasibly enumerated. For describing a semi-graphoid model therefore, researchers have proposed a more concise representation. This representation is composed of a representative subset of the independencies involved, called a basis, and lets all other independencies be implicitly defined by the semi-graphoid properties. An algorithm is available for computing such a basis for a semi-graphoid independency model. In this paper, we identify some new properties of a basis in general which can be exploited for arriving at an even more concise representation of a semi-graphoid model. Based upon these properties, we present an enhanced algorithm for basis construction which never returns a larger basis for a given independency model than currently existing algorithms.

1. Introduction

Mathematical models capturing joint probability distributions over sets of random variables are employed in numerous real-world applications. Especially probabilistic graphical models have become quite popular as appropriate models for describing distributions for problems in a range of societal fields. The practicability of computing probabilities of interest from these models typically derives from inference algorithms which exploit the modelled independency relation among the variables involved [5,6]. Independency relations embedded in joint probability distributions and their (concise) representation have therefore been subjects of extensive studies [2–4,9–11].

Pearl and his co-researchers were among the first to formalise qualitative properties of probabilistic independency in an axiomatic system [7,8]. The axioms from this system, which are known as the *semi-graphoid axioms*, are often looked upon as derivation rules for generating new independencies from a basic set of independency statements; any set of independencies that is closed under finite application of these rules is then called a *semi-graphoid independency model*. Semi-graphoid models constitute a quite general class among the various types of conditional independency model, in the sense that they provide for describing any independency relation embedded in a real-world probability distribution, even if this distribution is not strictly positive.

* Corresponding author.
E-mail addresses: stavros.lopatatzidis@ugent.be (S. Lopatatzidis), L.C.vanderGaag@uu.nl (L.C. van der Gaag).

http://dx.doi.org/10.1016/j.ijar.2016.06.011
0888-613X/

A range of computational problems on semi-graphoid independency models are being addressed in the literature. Two closely-related problems are the *implication problem* and the *representation problem*. The implication problem is the problem of deciding whether a given statement of independency can be derived from a given set of such statements [14]. The representation problem is the problem of finding a small subset of independency statements that fully describes a given independency model. Semi-graphoid independency models in general include exponentially many statements. Representing these models by enumeration of their element independencies therefore is not feasible in practice. Studený was the first to propose a concise representation of an independency model, based on the semi-graphoid axioms [12,13]. The idea is to explicitly enumerate a representative subset of independency statements from a semi-graphoid model and let all other independencies be defined implicitly through the derivation rules; such a representative subset of statements is termed a *basis* for the model at hand. Studený designed an efficient algorithm for computing a basis for a semi-graphoid model from a given starting set of independency statements, which was later improved by Baioletti and his co-researchers [1].

In this paper, we revisit the representation of semi-graphoid independency models, and show that the subset of independency statements which have to be represented explicitly, can often be further reduced in size. We introduce the new notion of maximal non-symmetric basis for this purpose, with an associated algorithm for its computation. Our algorithm is shown to never result in a larger basis for a given independency model than existing algorithms. Also, the intermediate bases constructed in the various iterations of our algorithm will never be larger than those constructed by existing algorithms. Our enhanced algorithm as a consequence improves not just upon the size of the resulting basis for representation but upon the runtime complexity of its construction as well.

The paper is organised as follows. We provide some preliminaries on semi-graphoid independency models in Section 2, and review concise representations and their associated construction algorithms in Section 3. In Section 4 we detail, among other notions and properties, our notion of maximal non-symmetric basis. Section 5 then describes our enhanced algorithm for basis construction, and demonstrate the practicability of our algorithm by means of a number of example independency models. The paper ends with our concluding observations in Section 6.

2. Semi-graphoid independency models

We briefly review semi-graphoid independency models [7,13], and thereby introduce our notational conventions. We consider a finite, non-empty set S of random variables. A *triplet* θ over S is a statement of the form $\theta = \langle A, B \mid C\rangle$, where $A, B, C \subseteq S$ are mutually disjoint subsets of S with $A, B \neq \varnothing$; we will use $X_\theta = A \cup B \cup C$ to refer to the triplet's set of variables. A triplet $\langle A, B \mid C\rangle$ states that the sets of variables A and B are mutually independent given the set C; in view of a joint probability distribution Pr over S, the triplet states that $\Pr(A, B \mid C) = \Pr(A \mid C) \cdot \Pr(B \mid C)$. The set of all triplets over S is denoted by $S^{(3)}$. A (sub-)set of triplets now constitutes a *semi-graphoid independency model* if it satisfies the four so-called semi-graphoid properties stated in the following definition.

Definition 1. A *semi-graphoid independency model* is a subset of triplets $J \subseteq S^{(3)}$ which satisfies the following properties:

G1: if $\langle A, B \mid C\rangle \in J$, then $\langle B, A \mid C\rangle \in J$ *(symmetry);*
G2: if $\langle A, B \mid C\rangle \in J$, then $\langle A, B' \mid C\rangle \in J$ for any non-empty subset $B' \subseteq B$ *(decomposition)*;
G3: if $\langle A, B_1 \cup B_2 \mid C\rangle \in J$ with $B_1 \cap B_2 = \varnothing$, then $\langle A, B_1 \mid C \cup B_2\rangle \in J$ *(weak union)*;
G4: if $\langle A, B \mid C \cup D\rangle \in J$ and $\langle A, C \mid D\rangle \in J$, then $\langle A, B \cup C \mid D\rangle \in J$ *(contraction).*

The four semi-graphoid properties jointly convey the idea that learning irrelevant information does not alter the independencies among the variables discerned [7]. The weak union property G3 for example, states that learning information about B_2 which is known to be irrelevant with respect to A given C cannot help irrelevant information about B_1 to become relevant to A. We note that the contraction rule G4 cannot always be applied to two arbitrarily chosen triplets; we will return to this observation in Section 4.1.

The semi-graphoid properties of independency are often viewed, and referred to, as derivation rules for generating (new) triplets from a given set of triplets. Given a starting set of triplets $J \subseteq S^{(3)}$ and a designated triplet $\theta \in S^{(3)}$, we write $J \vdash^* \theta$ if the triplet θ can be derived from J by finite application of the semi-graphoid rules G1, G2, G3 and G4. We note that the four rules thereby induce a derivational relation among the triplets of a semi-graphoid model. A variety of problems on semi-graphoid independency models are being studied by building upon this derivational relation. A well-known problem is the *implication problem* [14], which is the problem of deciding whether a specific triplet θ can be derived from a given set of triplets J. More formally, the problem asks whether a given triplet θ is included in the *closure* of a triplet set J, where the notion of closure is defined as follows.

Definition 2. Let $J \subseteq S^{(3)}$ be a set of triplets. Then, the *closure* of J, denoted by $\overline{J}$, is the set of all triplets $\theta \in S^{(3)}$ such that $J \vdash^* \theta$.

Another, closely related problem on semi-graphoid independency models is the *representation problem* [13], which is the problem of finding a (small) subset of triplets J that fully describes a given semi-graphoid independency model M. More formally, the problem asks for a *basis* for a given independency model, where the notion of basis is defined as follows.

Definition 3. Let M be a semi-graphoid independency model over a set of random variables S. Then, a set of triplets $J \subseteq S^{(3)}$ is called a *basis* for M if $\overline{J} = M$.

We observe that any triplet set J constitutes a basis for its own closure $\overline{J}$. The set J may not be a minimum nor a minimal basis for $\overline{J}$, however.

3. Representing semi-graphoid independency models

Semi-graphoid independency models typically are exponentially large in size, and representing them by enumeration of their element triplets is not feasible in practice. Studený was the first to propose a more concise representation of a semi-graphoid model based on the four semi-graphoid derivation rules [13]. The idea of his representation is to explicitly capture a basis for a model and let all other triplets be defined implicitly through these rules. In Sections 3.1 and 3.2 we review the basic notions underlying Studený's representation; in Section 3.3 we describe his algorithm for constructing a concise basis for the closure of a given triplet set. Throughout the three sections, we discuss, in addition, the notions introduced by Baioletti and his co-researchers to improve upon Studený's original algorithm.

3.1. The derivational relation between triplet pairs

For constructing concise representations of semi-graphoid independency models in general, several researchers have investigated and contributed to a characterisation of the derivational relation between triplets which is induced by application of the semi-graphoid rules. We begin by reviewing the notion of *dominance* which underlies the representation of independency models proposed by Studený [13].

Definition 4. Let $J \subseteq S^{(3)}$ be a semi-graphoid independency model, and let *G2s* and *G3s* be the following derivation rules over J:

G2s: if $\langle A, B \mid C\rangle \in J$, then $\langle A', B \mid C\rangle \in J$ for any non-empty subset $A' \subseteq A$;
G3s: if $\langle A_1 \cup A_2, B \mid C\rangle \in J$ with $A_1 \cap A_2 = \varnothing$, then $\langle A_1, B \mid C \cup A_2\rangle \in J$.

Now, let $\theta_i \in J$, $i = 1, 2$. If θ_1 can be derived from θ_2 by finite application of the rules *G2*, *G3*, *G2s* and *G3s*, we say that θ_1 is *dominated* by θ_2, denoted $\theta_1 \prec \theta_2$. A triplet $\theta \in J$ is called *dominant* in J if it is not dominated by any triplet $\tau \in J$ with $\tau \neq \theta$.

The notion of dominance pertains to a single triplet and the triplets that can be derived from it by means of the rules G2, G3, G2s and G3s, where the latter two rules incorporate the property of symmetry into the basic rules G2 and G3. We observe, from Definition 4, that any triplet θ is dominated by itself but not by the symmetric transpose θ^T which is obtained from θ by a single application of the symmetry rule G1. The following lemma reviews necessary and sufficient conditions for dominance of a triplet in general [13].

Lemma 1. *Let $\theta_i = \langle A_i, B_i \mid C_i\rangle \in S^{(3)}$ with $X_i = A_i \cup B_i \cup C_i$, $i = 1, 2$. Then, $\theta_1 \prec \theta_2$ if and only if the following conditions hold:*

- $C_2 \subseteq C_1 \subseteq X_2$;
- $A_1 \subseteq A_2$ *and* $B_1 \subseteq B_2$.

Informally spoken, the conditions stated in the lemma capture all possible ways in which the triplet θ_1 can be derived from the triplet $\theta_2 = \langle A_2, B_2 \mid C_2\rangle$ by means of the rules G2, G3, G2s and G3s. Multiple applications of the decomposition (G2) and weak union (G3) rule to θ_2 result in triplets $\theta_1 = \langle A_2, B_1 \mid C_1\rangle$ with $B_1 \subseteq B_2$ and $C_2 \subseteq C_1 \subseteq C_2 \cup (B_2 \setminus B_1)$; application of their symmetric counterparts G2s and G3s further induce the condition $A_1 \subseteq A_2$.

Similar to Studený's notion of dominance, Baioletti et al. [1] introduced the notion of *g-inclusion* for describing the derivational relation between pairs of triplets.

Definition 5. Let $J \subseteq S^{(3)}$ be a semi-graphoid independency model and let $\theta_i \in J$, $i = 1, 2$. Then, θ_1 is *g-included* in θ_2, denoted $\theta_1 \sqsubseteq \theta_2$, if θ_1 can be derived from θ_2 by finite application of the rules *G1*, *G2* and *G3*. A triplet $\theta \in J$ is called *g-maximal* in J if it is not g-included in any triplet τ from J with $\tau \notin \{\theta, \theta^T\}$.

Also for g-inclusion necessary and sufficient conditions have been formulated [1], which are summarised in the following lemma.

Lemma 2. *Let $\theta_i = \langle A_i, B_i \mid C_i\rangle \in S^{(3)}$ with $X_i = A_i \cup B_i \cup C_i$, $i = 1, 2$. Then, $\theta_1 \sqsubseteq \theta_2$ if and only if the following conditions hold:*

- $C_2 \subseteq C_1 \subseteq X_2$;
- *either* $A_1 \subseteq A_2$ *and* $B_1 \subseteq B_2$, *or* $B_1 \subseteq A_2$ *and* $A_1 \subseteq B_2$.

The definitions of dominance and g-inclusion respectively, show that the two notions are closely related. More specifically, for any two triplets θ_1, θ_2 with $\theta_1 \notin \{\theta_2, \theta_2^T\}$, the following property holds:

$$\theta_1 \sqsubseteq \theta_2 \quad \text{if and only if} \quad \theta_1 \prec \theta_2 \text{ or } \theta_1 \prec \theta_2^T$$

The main difference between the definitions of the two notions is that, while the notion of dominance incorporates symmetry implicitly through the symmetrical counterparts of the decomposition and weak union rules, the notion of g-inclusion involves symmetry explicitly through the symmetry rule itself. By including the rule of symmetry separately into the derivational system, any triplet is g-included in its symmetric transpose; we recall that, in contrast, a triplet is not dominated by its transpose. These properties are summarised in the following corollary.

Corollary 3. *For any triplet* $\theta \in S^{(3)}$, *the following properties hold:*

- $\theta \sqsubseteq \theta$ *and* $\theta \prec \theta$;
- $\theta \sqsubseteq \theta^T$, $\theta^T \sqsubseteq \theta$ *and* $\theta \not\prec \theta^T$, $\theta^T \not\prec \theta$.

For ease of notation in the sequel, we extend the notions of dominance and g-inclusion to apply to triplet sets.

Definition 6. Let $J_i \subseteq S^{(3)}$, $i = 1, 2$, be sets of triplets. Then,

- J_1 is *g-included* in J_2, denoted by $J_1 \sqsubseteq J_2$, if for each triplet $\theta \in J_1$ there exists a triplet $\theta' \in J_2$ such that $\theta \sqsubseteq \theta'$;
- J_1 is *dominated* by the set J_2, denoted by $J_1 \prec J_2$, if for each triplet $\theta \in J_1$, there exists a triplet $\theta' \in J_2$ such that $\theta \prec \theta'$.

3.2. Including the contraction rule into the derivational relation

The properties of the derivational relation between triplets reviewed above involve just the derivation rules G1, G2 and G3. The remaining rule G4 differs from these three rules in that it constructs a triplet by combining information from two triplets rather than by building upon a single triplet. We write $\{\theta_1, \theta_2\} \vdash_{G4} \theta$ to denote application of the contraction rule G4 to the two triplets θ_1, θ_2 to yield a third triplet θ. We observe from its definition that the contraction rule cannot always be applied to two arbitrary triplets. To accommodate the rule, Studený designed a dedicated operator, called the *gc*-operator [13], which constructs from two triplets θ_1, θ_2 triplets θ_1', θ_2' by application of the derivation rules G2, G3, G2s and G3s, to which the contraction rule can be applied to yield a possibly new triplet θ. This *gc*-operator is defined as follows.

Definition 7. For all triplet pairs $\theta_i = \langle A_i, B_i \mid C_i \rangle \in S^{(3)}$ with $X_i = A_i \cup B_i \cup C_i$, $i = 1, 2$, such that

- $A_1 \cap A_2 \neq \varnothing$,
- $C_1 \setminus X_2 = \varnothing$,
- $C_2 \setminus X_1 = \varnothing$, and
- $(B_2 \setminus C_1) \cup (B_1 \cap X_2) \neq \varnothing$,

the *gc-operator* is defined through:

$$gc(\theta_1, \theta_2) = \langle A_1 \cap A_2, (B_2 \setminus C_1) \cup (B_1 \cap X_2) \mid C_1 \cup (A_1 \cap C_2) \rangle$$

For all pairs of triplets θ_1, θ_2 for which the four conditions stated above do not hold, $gc(\theta_1, \theta_2)$ is undefined.

Studený showed that if the *gc*-operator is applicable to two triplets θ_1, θ_2 to result in a valid triplet θ, then this triplet θ dominates all elements that can be derived from θ_1, θ_2 by applying the rules G2, G3, G4, G2s and G3s. A slightly more detailed result is stated in the following lemma by Baioletti et al. [1].

Lemma 4. *Let* $J \subseteq S^{(3)}$ *be a semi-graphoid independency model, and let* $\theta_i = \langle A_i, B_i \mid C_i \rangle \in J$ *with* $X_i = A_i \cup B_i \cup C_i$, $i = 1, 2$. *Let the operator gc be as in Definition 7 and let*

$$H_{G4}(\theta_1, \theta_2) = \{\theta \mid \exists\, \theta_1' \prec \theta_1, \theta_2' \prec \theta_2 \in J \text{ such that } \theta_1', \theta_2' \vdash_{G4} \theta \text{ and } \theta \text{ is valid}\}$$

Then,

- $H_{G4}(\theta_1, \theta_2) = \varnothing$ *if and only if at least one of the following conditions fails:*
  C.1 $A_1 \cap A_2 \neq \varnothing$;
  C.2 $C_1 \subseteq X_2$ *and* $C_2 \subseteq X_1$;
  C.3 $B_2 \setminus C_1 \neq \varnothing$;
  C.4 $B_1 \cap X_2 \neq \varnothing$;
  C.5 $|(B_2 \setminus C_1) \cup (B_1 \cap X_2)| \geq 2$;
- *if* $H_{G4}(\theta_1, \theta_2) \neq \varnothing$, *then* $gc(\theta_1, \theta_2) \in H_{G4}(\theta_1, \theta_2)$ *and* $\tau \prec gc(\theta_1, \theta_2)$ *for any triplet* $\tau \in H_{G4}(\theta_1, \theta_2)$ *with* $\tau \neq gc(\theta_1, \theta_2)$.

The lemma in essence states that the *gc*-operator indeed constructs dominating triplets. We observe that the set $H_{G4}(\theta_1, \theta_2)$ includes all triplets which can be derived by applying the contraction rule G4 to the triplet pair θ_1, θ_2 and to all triplets which are pairwise dominated by θ_1, θ_2. The conditions C1–C5 mentioned in the lemma now are taken as the conditions under which the *gc*-operator can be applied to give a valid, dominating triplet θ.

The notion of g-inclusion introduced by Baioletti et al. is connected with the contraction rule G4 as follows [1].

Lemma 5. *Let* $\theta_1, \theta_1', \theta_2, \theta_2' \in S^{(3)}$ *be triplets such that* $\theta_1' \sqsubseteq \theta_1$, $\theta_2' \sqsubseteq \theta_2$. *If* $\theta_1', \theta_2' \vdash_{G4} \theta'$ *and* $\theta_1, \theta_2 \vdash_{G4} \theta$, *then* $\theta' \sqsubseteq \theta$.

Informally spoken, the lemma states that the contraction of two g-included triplets θ_1', θ_2' is itself g-included in the contraction of their including triplets θ_1, θ_2.

3.3. Computing a basis for a semi-graphoid model

For representing a semi-graphoid independency model, it suffices to find a subset of triplets which captures the same information as the entire model itself, that is, it suffices to find a basis. Studený designed an algorithm for this purpose based upon the observation that dominated triplets do not convey any additional information about a model and thus are not required explicitly for its representation [13]. His algorithm generates, from a given set of triplets, all dominant triplets of the semi-graphoid model defined by this starting triplet set, thereby establishing a basis for the model. More specifically, the algorithm constructs all dominant triplets directly using the *gc*-operator, without explicitly generating the full closure of the starting triplet set.

Studený's original algorithm takes a starting triplet set for its input and, in a pre-processing step, adds any symmetric triplet which is not yet included. It then applies the *gc*-operator to any pair of triplets for which the conditions C1–C5 from Lemma 4 hold, adding the results to the basis under construction. Subsequently, all dominated triplets are removed. These steps are re-iterated until the basis no longer changes. Studený's original algorithm was later improved by Baioletti and his co-researchers through the definition of a generalised operator and contraction rule. In the sequel, we will refer to the overall improved algorithm as the *Studený–Baioletti algorithm*.

Given a starting set of triplets, Studený's original algorithm begins with adding all symmetric transposes, to allow all possible applications of the derivation rules G2, G3, G2s, G3s and G4. To forestall the need of having to explicitly add symmetric triplets to a basis under construction, Baioletti et al. generalised the *gc*-operator as defined below [1].

Definition 8. For all triplet pairs $\theta_i = \langle A_i, B_i \mid C_i \rangle \in S^{(3)}$ with $X_i = A_i \cup B_i \cup C_i$, $i = 1, 2$, the GC-*operator* is defined through:

$$GC(\theta_1, \theta_2) = \{\theta \mid \hat{\theta}_1 \in \{\theta_1, \theta_1^T\}, \hat{\theta}_2 \in \{\theta_2, \theta_2^T\} \text{ with } gc(\hat{\theta}_1, \hat{\theta}_2) = \theta \text{ a valid triplet}\}$$

We note that the *GC*-operator constructs not just the single triplet from applying Studený's *gc*-operator to θ_1, θ_2, but those from applying this operator to all combinations involving symmetric transposes as well. To accommodate application of the *GC*-operator, the contraction rule G4 is generalised to fully integrate the property of symmetry into the derivation. The generalised contraction rule G4* is as defined below [1].

Definition 9. Let $J \subseteq S^{(3)}$ be a semi-graphoid independency model. Then, G4* is the following derivation rule over J:

$$\text{G4}^*: \text{ if } \theta_1, \theta_2 \in J, \text{ then } GC(\theta_1, \theta_2) \cup GC(\theta_2, \theta_1) \subseteq J$$

Forestalling the pre-processing step of Studený's original algorithm for basis construction, the Studený–Baioletti algorithm now applies the G4* derivation rule to any pair of triplets for which the conditions of Lemma 4 hold. After repeated application of G4* to a starting set J, a triplet set results which is related to the closure of J as stated in the following lemma [1].

Lemma 6. *Let* $J \subseteq S^{(3)}$ *and let* $\overline{J}$ *be its closure. Let* J^{G4^*} *be the set of all triplets that are derived from* J *by the derivation rule* G4*. *Then,* $J^{G4^*} \subseteq \overline{J}$ *and* $\overline{J} \sqsubseteq J^{G4^*}$.

The property $J^{G4^*} \subseteq \overline{J}$ mentioned in the lemma states that application of the derivation rule G4* does not yield any triplets which are not in the closure of the starting set J; the property $\overline{J} \sqsubseteq J^{G4^*}$ implies that all triplets from $\overline{J}$ are represented in

BASIS CONSTRUCTION

Input: $J \subseteq S^{(3)}$
Output: J_k with $\overline{J_k} = \overline{J}$

1: **function** *Basis-Construction*(J)
2: $J_0 \leftarrow J$
3: $N_0 \leftarrow J$
4: $k \leftarrow 0$
5: **repeat**
6: $k \leftarrow k+1$
7: $N_k \leftarrow \bigcup_{\theta_1 \in J_{k-1}, \theta_2 \in N_{k-1}} (GC(\theta_1, \theta_2) \cup GC(\theta_2, \theta_1))$
8: $J_k \leftarrow (J_{k-1} \cup N_k)_{/\sqsubseteq}$
9: **until** $J_k = J_{k-1}$
10: **return** J_k
11: **end function**

Fig. 1. The Studený–Baioletti algorithm for basis construction.

J^{G4^*} through g-inclusion. The lemma thereby states that finite application of G4* serves to generate essentially the same information from the set J as finite application of the four semi-graphoid rules G1, G2, G3, and G4, and hence that J^{G4^*} constitutes a basis for the closure of the starting set J. The triplet set J^{G4^*} does not necessarily constitute a minimal basis for the model however, as it may include various redundant triplets. To reduce the set J^{G4^*} without losing any information, it is restricted to its subset of g-maximal triplets. We define the notion of g-maximal triplet subset for triplet sets in general.

Definition 10. Let $J \subseteq S^{(3)}$ be a triplet set. A *g-maximal triplet subset* $J_{/\sqsubseteq}$ of J is a set of triplets such that $J_{/\sqsubseteq} = \{\theta \in J \mid \nexists \theta' \in J$ with $\theta' \notin \{\theta, \theta^T\}$ such that $\theta \sqsubseteq \theta'\}$.

Given a starting set J, a g-maximal triplet subset of its closure $\overline{J}$ contains the same independency information as the closure itself and hence constitutes a basis for the semi-graphoid model at hand. More formally, the following lemma [1] holds for g-maximal triplet subsets of the closure $\overline{J}$ and of the triplet set J^{G4^*} constructed from J, respectively.

Lemma 7. *Let $J \subseteq S^{(3)}$ be a triplet set, and let $\overline{J}_{/\sqsubseteq}$ and $J^{G4^*}_{/\sqsubseteq}$ be g-maximal triplet subsets of $\overline{J}$ and J^{G4^*} as defined above. Then, $J^{G4^*}_{/\sqsubseteq} \sqsubseteq \overline{J}_{/\sqsubseteq}$ and $\overline{J}_{/\sqsubseteq} \sqsubseteq J^{G4^*}_{/\sqsubseteq}$.*

The lemma states that any pair of g-maximal triplet subsets, of the closure $\overline{J}$ and of the set J^{G4^*} respectively, share exactly the same information even though the two g-maximal sets may differ.

From Lemma 7 we now conclude that constructing a basis for a semi-graphoid independency model amounts to taking a g-maximal triplet subset $J^{G4^*}_{/\sqsubseteq}$ from the set of triplets which result from repeated application of the G4* derivation rule. The overall Studený–Baioletti algorithm for basis construction now is summarised in Fig. 1. Starting with the initial triplet set J, the algorithm computes, in each iteration, the triplets which result from all possible applications of the G4* derivation rule (line 7) and adds these to the triplet set under construction. The g-included triplets are subsequently removed (line 8), and the procedure is repeated until the basis under construction no longer changes in the sense that no new g-maximal triplets are being added.

In the worst case, the Studený–Baioletti algorithm has a runtime complexity which is exponential in the size of the starting triplet set J. The essence of the algorithm is the iterative application of the G4* derivation rule in line 7. In the first iteration, the *GC*-operator is applied twice for each pair of triplets from the starting triplet set J_0. Each application of the *GC*-operator involves four applications of the *gc*-operator, each of which requires verifying conditions C1–C5 from Lemma 4 and a (limited) number of set manipulations. With $|J_0| = n$ being the number of triplets in $J_0 = J$, the first iteration of line 7 of the algorithm thus takes $O(n^2)$ time. In the worst case, this iteration may result in a set N_1 with $O(n^2)$ new triplets. Since $O(n^2)$ triplets in the constructed set $J_0 \cup N_1$ can be g-maximal, the intermediate basis J_1 for the algorithm's next iteration may include $O(n^2)$ triplets. In the second iteration, $|J_1| \cdot |N_1|$ triplet pairs are considered, which amounts to $O(n^4)$ applications of the *gc*-operator; in the worst case, these applications may result in an intermediate basis of size $O(n^4)$. In general, the kth iteration of line 7 may take $O(n^{2^k})$ time and result in an intermediate basis of size $O(n^{2^k})$. Each iteration of the algorithm thus takes polynomial time, of a power dependent on the iteration. In practical applications, the power of the polynomial for the kth iteration is also dependent on the sizes of the triplet sets N_i and intermediate bases J_i constructed in the earlier iterations. The number of iterations required before meeting the stopping criterion from line 9 for the algorithm's main loop, may in the worst case be exponential in the size of the starting triplet set, as the algorithm may need to investigate a sizeable part of the closure of this triplet set.

4. Revisiting the derivational relation among triplets

Upon revisiting the derivational relation among the triplets of a semi-graphoid independency model, we identified two properties which can be exploited for enhancing the Studený–Baioletti algorithm for basis construction. Our first enhance-

ment is based on the observation that particular elements from a starting triplet set can be excluded from consideration during basis computation; we state, in Section 4.1, necessary conditions for identifying such triplets. In Section 4.2, we further argue that symmetric transposes need not be added or kept throughout the computations. We will enhance the Studený–Baioletti algorithm with these two properties in Section 5. As a preliminary to our enhanced algorithm, Section 4.3 introduces a tailored representation of triplets which supports their efficient manipulation.

4.1. Excluding lonely triplets from consideration

The Studený–Baioletti algorithm for basis construction builds on application of the G4* derivation rule, and on using the *gc*-operator more specifically. As we have already mentioned in Section 2, the contraction property underlying this rule does not always apply to any arbitrarily chosen pair of triplets. In fact, a starting triplet set may include triplets to which the gc-operator can *never* be applied to yield (potentially) new triplets. The following lemma identifies such *lonely* triplets.

Lemma 8. *Let $J \subseteq S^{(3)}$ be a triplet set and let $\theta = \langle A, B \mid C\rangle \in J$ with $X = A \cup B \cup C$. If at least one of the following conditions holds for all other triplets $\theta' = \langle A', B' \mid C'\rangle \in J$, $\theta' \notin \{\theta, \theta^T\}$, with $X' = A' \cup B' \cup C'$:*

E1. $C \not\subseteq X'$;
E2. $A \cap (A' \cup B') = \varnothing$ *and* $B \cap (A' \cup B') = \varnothing$;
E3. $(A \cup B) \setminus C' = \varnothing$ *and* $(A' \cup B') \setminus C = \varnothing$;
E4. $(A \cup B) \cap X' = \varnothing$ *and* $(A' \cup B') \cap X = \varnothing$;

then $J^{G4^*} \setminus \{\theta, \theta^T\} = (J \setminus \{\theta, \theta^T\})^{G4^*}$.

Proof. We focus on the triplet $\theta = \langle A, B \mid C\rangle$ with $X = A \cup B \cup C$ as stated in the lemma, and consider another triplet $\theta_i = \langle A_i, B_i \mid C_i\rangle$, $\theta_i \notin \{\theta, \theta^T\}$, from J, with $X_i = A_i \cup B_i \cup C_i$. Application of the G4* derivation rule to the pair of triplets θ, θ_i produces the triplet set $GC(\theta, \theta_i) \cup GC(\theta_i, \theta)$. We know that this set is non-empty only if applying the gc-operator yields at least one valid triplet, that is, only if all conditions C1–C5 from Lemma 4 hold.

We assume that at least one of the four conditions E1–E4 stated above holds for all triplet pairs θ, θ_i alike. With condition E1, that is, with $C \not\subseteq X_i$, we find that condition C2 from Lemma 4 does not hold as it requires $C \subseteq X_i$; the triplet set derived from θ, θ_i by a single application of G4* thus is empty. Using analogous arguments with respect to the conditions E2, E3 and E4, we conclude that by a single application of the G4* derivation rule to any triplet pair involving θ no valid triplets can result.

It now remains to be shown that application of the G4* derivation rule to θ and any triplet that may have been generated by applying G4* to another triplet pair from J, cannot result in a well-defined, valid triplet. To this end, we consider two triplets $\theta_j = \langle A_j, B_j \mid C_j\rangle$ and $\theta_k = \langle A_k, B_k \mid C_k\rangle$, $\theta_j, \theta_k \notin \{\theta, \theta^T\}$, from J such that application of G4* to the pair θ_j, θ_k results in at least one valid triplet $\theta' = \langle A', B' \mid C'\rangle$ with $X' = A' \cup B' \cup C'$. Since a valid triplet is constructed, we know from Lemma 4 that, among other properties, $C_j \subseteq X_k$, $C_k \subseteq X_j$ must hold; we further know that $X' \subseteq X_j \cup X_k$.

From our assumption, we have that at least one of the conditions E1–E4 must hold for both the triplet pair θ, θ_j and the triplet pair θ, θ_k. With each condition therefore, we have to show that application of the G4* derivation rule to θ, θ' does not yield any valid triplets. We prove the property for the condition E1; the proofs pertaining to the other three conditions E2, E3 and E4 are analogous. We assume that the condition E1 holds for both triplet pairs θ, θ_j and θ, θ_k alike. We then have that $C \not\subseteq X_j$ and $C \not\subseteq X_k$. We now distinguish between two cases:

- if $C \not\subseteq (X_j \cup X_k)$, then there must be a variable $V \in C$ with $V \notin X_j$ and $V \notin X_k$. Since $X' \subseteq X_j \cup X_k$, we find that $C \not\subseteq X'$, from which we conclude, by Lemma 4, that application of the G4* derivation rule results in an empty triplet set;
- if $C \subseteq (X_j \cup X_k)$, then there must be two variables V, W such that $\{V, W\} \subseteq C$, $V \in X_j$, $V \notin X_k$ and $W \in X_k$, $W \notin X_j$. Since $C_j \subseteq X_k$, we know that $V \notin C_j$ and hence that $V \in (A_j \cup B_j)$; similarly, $W \in (A_k \cup B_k)$. From the construction of θ', we observe that $\{V, W\} \not\subseteq X'$. We thus find that $C \not\subseteq X'$ and conclude, by Lemma 4, that application of the G4* derivation rule does not result in any valid triplet.

From the above considerations, we have that the triplet set $GC(\theta, \theta_i) \cup GC(\theta_i, \theta)$ is empty for all triplets θ_i which are themselves in J or are constructed from triplets in J. We conclude that finite application of the G4* derivation rule to a triplet pair involving the triplet θ does not yield any valid triplets, from which we conclude that $J^{G4^*} \setminus \{\theta, \theta^T\} = (J \setminus \{\theta, \theta^T\})^{G4^*}$. □

From Lemma 8, we conclude that the lonely triplets from a starting set are not involved in the derivation of any set N_k of newly constructed triplets in the Studený–Baioletti algorithm and, hence, can be excluded from the computations of the algorithm's main loop. The following lemma now further shows that the lonely triplets are g-maximal in any basis of the closure of the starting set.

Lemma 9. *Let $J \subseteq S^{(3)}$ be a triplet set and let $\overline{J}$ be its closure. Let $J^- \cup J^*$, with $J^- \cap J^* = \varnothing$, be a basis of $\overline{J}$, such that J^* contains all lonely triplets from J. Then,*

- *for all triplets $\theta \in J^*$, there do not exist any triplets $\theta' \in J^-$ such that $\theta \sqsubseteq \theta'$;*
- *for all triplets $\theta' \in J^-$, there do not exist any triplets $\theta \in J^*$ such that $\theta' \sqsubseteq \theta$.*

Proof. From each triplet $\theta \in J^*$ being a lonely triplet in the starting set J, we have that at least one of the conditions E1–E4 from Lemma 8 holds for all triplet pairs θ, θ' with $\theta' \in J^-$. It now suffices to show, for each of the four conditions, that if it holds, then at least one of the conditions for g-inclusion from Lemma 2 does not hold. We prove this property for the condition E3 from Lemma 8; the proofs for the other three conditions are analogous.

We assume that the condition E3 from Lemma 8 holds for each pair θ, θ' with $\theta \in J^*$, $\theta' \in J^-$. We then have that $(A \cup B) \subseteq C'$. Since the sets A', B', C' are disjoint, it follows that $A' \nsubseteq A$. Similarly, we find that $B' \nsubseteq A$. We conclude from Lemma 2 that the conditions for g-inclusion do not hold, and hence that $\theta \not\sqsubseteq \theta'$. Using the property $(A' \cup B') \subseteq C$, we equally find that $\theta' \not\sqsubseteq \theta$. □

From the properties stated in the two lemmas above, we have that the lonely triplets from a starting set can be set aside throughout the basis computations, only to be added again to the final basis in a post-processing step. We note that by doing so, the number of applications of the $G4^*$ derivation rule is effectively reduced for any triplet set which includes at least one lonely triplet; we will return to this observation in our examples in Section 5. We further note that for identifying the lonely triplets from a starting set at most $(n-1)^2/2$ triplet comparisons are required, where n is the size of the starting set at hand.

4.2. Maintaining a non-symmetric basis

We recall from Section 2 that Studený's original algorithm for basis construction included a pre-processing step which served to add all symmetric transposes to a starting set to allow all possible applications of the G4 derivation rule. In the reformulation of Studený's algorithm by Baioletti and his co-researchers, these applications are covered through the $G4^*$ derivation rule, as a consequence of which the original pre-processing step is no longer required. Now, upon constructing the sets $GC(\theta, \theta') \cup GC(\theta', \theta)$ of potentially new triplets by means of the gc-operator, symmetric transposes may arise and be included in a basis under construction. Upon taking a g-maximal subset of such an intermediate basis, both a triplet θ and its transpose θ^T may be included. Once introduced therefore, transposes may be carried throughout all further computations and end up in the final basis. Since a triplet θ is g-included in its symmetric transpose θ^T and vice versa, however, it is not necessary to include both triplets in the final basis. We now show that symmetric transposes also need not be retained in the intermediate bases constructed in the algorithm's successive iterations. In fact, by removing symmetric transposes from an intermediate g-maximal triplet set, a set of triplets results which shares the same information as the full closure of the starting set and hence still constitutes a basis. We begin by defining the notion of maximal non-symmetric triplet subset.

Definition 11. Let $J \subseteq S^{(3)}$ be a triplet set. A *maximal non-symmetric triplet subset* $J_{/\sqsubseteq n}$ of J is a g-maximal triplet subset of J which satisfies the following property:

$$\text{if } \theta \in J_{/\sqsubseteq n}, \text{ then } \theta^T \notin J_{/\sqsubseteq n}$$

From the definition, we note that a maximal non-symmetric subset of a starting set J can be constructed from a g-maximal subset of J by removing one triplet from each pair of symmetric transposes; the resulting maximal non-symmetric set then is a subset of this g-maximal triplet set. This property is stated more formally in the following lemma.

Lemma 10. *Let $J \subseteq S^{(3)}$ be a triplet set, and let $J_{/\sqsubseteq n}$ be a maximal non-symmetric triplet subset of J. Then, there exists a g-maximal triplet subset $J_{/\sqsubseteq}$ of J such that $J_{/\sqsubseteq n} \subseteq J_{/\sqsubseteq} \subseteq J$.*

Proof. The property follows immediately from Definition 11. □

In the following lemma, we further show that, given a starting set J, any maximal non-symmetric subset of J and any g-maximal subset of J share the same information.

Lemma 11. *Let $J \subseteq S^{(3)}$ be a triplet set. Let $J_{/\sqsubseteq}$ be a g-maximal triplet subset of J and let $J_{/\sqsubseteq n}$ be a maximal non-symmetric subset of J. Then, $J_{/\sqsubseteq n} \sqsubseteq J_{/\sqsubseteq}$ and $J_{/\sqsubseteq} \sqsubseteq J_{/\sqsubseteq n}$.*

Proof. We consider a triplet $\theta \in J_{/\sqsubseteq n}$. Since both $J_{/\sqsubseteq}$ and $J_{/\sqsubseteq n}$ are g-maximal subsets of J, at least one of θ and θ^T need be included in $J_{/\sqsubseteq}$. Since $\theta \sqsubseteq \theta$ and $\theta \sqsubseteq \theta^T$, we find that $J_{/\sqsubseteq n} \sqsubseteq J_{/\sqsubseteq}$. By a similar argument, we find that $J_{/\sqsubseteq} \sqsubseteq J_{/\sqsubseteq n}$. □

We recall that constructing a basis for a starting set J by the Studený–Baioletti algorithm in essence amounts to computing a g-maximal subset of the set J^{G4^*} of all triplets that are derived from J by the G4* derivation rule. The following lemma now relates the notion of maximal non-symmetric subset to the basis construction by showing that a maximal non-symmetric subset of the set J^{G4^*} carries the exact same information as a maximal non-symmetric triplet subset of the closure of the starting set J.

Lemma 12. *Let $J \subseteq S^{(3)}$ be a triplet set and let $\overline{J}$ be its closure. Let $J^{G4^*}_{/\sqsubseteq n}$ be a maximal non-symmetric subset of the set J^{G4^*} and let $\overline{J}_{/\sqsubseteq n}$ be a maximal non-symmetric subset of $\overline{J}$. Then, $J^{G4^*}_{/\sqsubseteq n} \sqsubseteq \overline{J}_{/\sqsubseteq n}$ and $\overline{J}_{/\sqsubseteq n} \sqsubseteq J^{G4^*}_{/\sqsubseteq n}$.*

Proof. From Lemma 11, we have that $J^{G4^*}_{/\sqsubseteq n} \sqsubseteq J^{G4^*}_{/\sqsubseteq}$ and $\overline{J}_{/\sqsubseteq} \sqsubseteq \overline{J}_{/\sqsubseteq n}$. Since $J^{G4^*}_{/\sqsubseteq} \sqsubseteq \overline{J}_{/\sqsubseteq}$, it follows that $J^{G4^*}_{/\sqsubseteq n} \sqsubseteq \overline{J}_{/\sqsubseteq n}$. By a similar argument, we find that $\overline{J}_{/\sqsubseteq n} \sqsubseteq J^{G4^*}_{/\sqsubseteq n}$. □

From the previous two lemmas we conclude that a maximal non-symmetric subset of the set J^{G4^*} of G4*-derived triplets constitutes an appropriate basis for the semi-graphoid model defined by the triplet set J. We now argue that the property that this basis will not include any symmetric triplet pairs, can be exploited already during its construction. Let $J_k = (J_{k-1} \cup N_k) \subseteq \overline{J}$ be an intermediate basis computed in the kth iteration of the Studený–Baioletti algorithm, from which we are about to remove any g-included triplets. If the set J_k includes triplets θ, θ^T and θ' with $\theta \sqsubseteq \theta'$, then both triplets θ, θ^T will be removed from J_k by the algorithm upon taking the g-maximal triplet subset of J_k. If J_k includes the two triplets θ, θ^T and no other triplet θ' with $\theta \sqsubseteq \theta'$, then both θ and θ^T are retained in the g-maximal triplet subset taken from J_k. Since the two triplets are mutually g-included however, one of them can safely be removed without losing any information. For the $(k+1)$st iteration of the algorithm therefore, it suffices to take the maximal non-symmetric subset of J_k.

4.3. Representing triplets for enhanced basis construction

Exploiting the properties described in the Sections 4.1 and 4.2, Section 5 will detail our enhanced algorithm for basis construction. Like the Studený–Baioletti algorithm, our algorithm builds in essence on application of the G4* derivation rule. To reduce the computational burden involved however, our algorithm employs a tailored representation of pairs of triplets and an accordingly adapted derivation rule. We begin by defining our representation of triplet pairs.

Definition 12. Let $\theta_i = \langle A_i,\ B_i \mid C_i \rangle \in S^{(3)}$, $i = 1, 2$. The triplet pair θ_1, θ_2 is in *normal form* if θ_1, θ_2 can be written as

$$\theta_1 = \langle A_A \cup A_B \cup A_C \cup A_X,\ B_A \cup B_B \cup B_C \cup B_X \mid C_A \cup C_B \cup C_C \cup C_X \rangle$$
$$\theta_2 = \langle A_A \cup B_A \cup C_A \cup A_Y,\ A_B \cup B_B \cup C_B \cup B_Y \mid A_C \cup B_C \cup C_C \cup C_Y \rangle$$

with $A_A = (A_i \cap A_j)$, $A_B = (A_i \cap B_j)$, $A_C = (A_i \cap C_j)$, $A_X = (A_i \setminus X_j)$, $A_Y = (A_j \setminus X_i)$ and with the other subsets defined analogously. The pair θ_1, θ_2 is in *strong normal form* if it is in normal form and $C_X = C_Y = \varnothing$.

We recall that the Studený–Baioletti algorithm computes the triplet set $GC(\theta_1, \theta_2) \cup GC(\theta_2, \theta_1)$ for all pairs of triplets θ_1, θ_2 from an intermediate basis. Upon doing so, the basic gc-operator is applied eight times, and for each application the five conditions from Lemma 4 are evaluated. We observe from our definition above that, if a pair of triplets θ_1, θ_2 is written in (strong) normal form, then each of the eight potential triplets for the set $GC(\theta_1, \theta_2) \cup GC(\theta_2, \theta_1)$ can be generated simply by manipulating the subsets identified in the representation; for example, the potential triplet $gc(\theta_1, \theta_2)$ is constructed to be $\langle A_A,\ A_B \cup B_B \cup B_Y \cup B_A \cup B_C \mid C_A \cup C_B \cup C_C \cup C_X \cup A_C \rangle$. We note that formulating a pair of triplets in (strong) normal form amounts to establishing the various subsets involved. Through this representation therefore, we forestall any duplicate evaluations of the conditions from Lemma 4.

We now further observe that a pair of triplets θ_1, θ_2 can be brought in strong normal form only if the conditions $C_1 \subseteq X_2$ and $C_2 \subseteq X_1$ are satisfied. As these conditions constitute also a necessary condition for generating a valid dominating triplet upon applying the gc-operator to the pair, we are guaranteed that if two triplets cannot be represented in strong normal form, they cannot yield any dominating triplets by using the operator. For constructing new triplets from an intermediate basis therefore, we need to consider pairs of triplets in strong normal form only. Still, for a pair of triplets in strong normal form, the conditions from Lemma 4 may not all hold. The following lemma shows that application of the gc-operator to such a pair would not result in a triplet which is not included in the closure of the basis at hand.

Lemma 13. *Let $J \subseteq S^{(3)}$ be a triplet set, and let $\theta_i = \langle A_i, B_i \mid C_i \rangle \in J$ with $X_i = A_i \cup B_i \cup C_i$, $i = 1, 2$, such that $C_1 \subseteq X_2$ and $C_2 \subseteq X_1$. Let the gc-operator be defined as in Definition 7. Then, if at least one of the conditions from Lemma 4 fails, the triplet $gc(\theta_1, \theta_2)$ is either undefined or g-included in θ_1 or θ_2.*

Proof. We consider the triplet $\langle A_1 \cap A_2, (B_2 \setminus C_1) \cup (B_1 \cap X_2) | C_1 \cup (A_1 \cap C_2) \rangle$ which results from applying the gc-operator to the triplet pair θ_1, θ_2. It now suffices to address each of the four conditions C1, C3, C4 and C5 of Lemma 4 separately,

and show that, if the condition fails, the triplet $gc(\theta_1, \theta_2)$ is either undefined or g-included in θ_1 or θ_2; we note that the condition C2 always holds by the precondition of the current lemma. We prove the stated property for the condition C5; the proofs for the other conditions are analogous.

We assume that the condition C5 from Lemma 4 fails and, hence, that $|(B_2 \setminus C_1) \cup (B_1 \cap X_2)| < 2$. We distinguish between the two cases where $|(B_2 \setminus C_1) \cup (B_1 \cap X_2)| = 0$ and $|(B_2 \setminus C_1) \cup (B_1 \cap X_2)| = 1$. In the first case, we observe that the generated 'triplet' $gc(\theta_1, \theta_2) = \langle A_1 \cap A_2, \varnothing \mid C_1 \cup (C_2 \cap A_1) \rangle$ is invalid because its second component is empty. In the second case, we find that $((B_2 \setminus C_1) \cup (B_1 \cap X_2)) = B_1 \cap B_2$, and that the generated triplet equals $g(\theta_1, \theta_2) = \langle A_1 \cap A_2, B_1 \cap B_2 \mid C_1 \cup (C_2 \cap A_1) \rangle$. As $C_1 \subseteq C_1 \cup (C_2 \cap A_1)$, $A_1 \cap A_2 \subseteq A_1$ and $B_1 \cap B_2 \subseteq B_1$, this triplet is g-included in θ_1 and therefore does not convey any new information. □

From the lemma we conclude that for a triplet pair in strong normal form it is not necessary to evaluate the four conditions C1, C3, C4 and C5 from Lemma 4 explicitly before applying the gc-operator: if we would apply the operator to the pair in case at least one of the conditions fails, the result would be either undefined or g-included. We note that verifying g-inclusion of the resulting triplet in the two defining triplets θ_1 and θ_2 would be no more demanding than verifying the conditions from Lemma 4.

5. Revisiting basis computation

Exploiting the properties detailed in the previous section, we formulate in Section 5.1 our enhanced algorithm for basis construction. We will show that our algorithm never results in a larger basis for a given triplet set, and in fact never constructs larger intermediate bases than the Studený–Baioletti algorithm. The practicability of our enhanced algorithm will be demonstrated in Section 5.2 by means of various example independency models.

5.1. The enhanced algorithm for basis construction

Like the Studený–Baioletti algorithm, our enhanced algorithm for basis construction builds in essence on iterated application of the $G4^*$ derivation rule. Building upon our new triplet representation however, we re-formulate the operator and define an associated derivation rule. Our enhanced operator GC^+ is defined as follows.

Definition 13. For all triplet pairs $\theta_i = \langle A_i, B_i \mid C_i \rangle \in S^{(3)}$ with $X_i = A_i \cup B_i \cup C_i$, $i = 1, 2$, in strong normal form, the GC^+-*operator* is defined through

$$GC^+(\theta_1, \theta_2) = \{\theta \mid \theta \in J^* \textit{ is a valid triplet with } \theta \not\sqsubseteq \theta_1 \textit{ and } \theta \not\sqsubseteq \theta_2\}$$

where the set J^* is composed of the following potential triplets:

$$\theta_1 = \langle A_A, A_B \cup B_A \cup B_B \cup B_C \cup B_{D'} \mid A_C \cup C_A \cup C_B \cup C_C \rangle$$
$$\theta_2 = \langle A_B, A_A \cup B_A \cup B_B \cup B_C \cup A_{D'} \mid A_C \cup C_A \cup C_B \cup C_C \rangle$$
$$\theta_3 = \langle B_A, A_A \cup A_B \cup A_C \cup B_B \cup B_{D'} \mid B_C \cup C_A \cup C_B \cup C_C \rangle$$
$$\theta_4 = \langle B_B, A_A \cup A_B \cup A_C \cup B_A \cup A_{D'} \mid B_C \cup C_A \cup C_B \cup C_C \rangle$$
$$\theta_5 = \langle A_A, A_B \cup B_A \cup B_B \cup B_D \cup C_B \mid A_C \cup B_C \cup C_A \cup C_C \rangle$$
$$\theta_6 = \langle B_A, A_A \cup A_B \cup A_D \cup B_B \cup C_B \mid A_C \cup B_C \cup C_A \cup C_C \rangle$$
$$\theta_7 = \langle A_B, A_A \cup B_A \cup B_B \cup B_D \cup C_A \mid A_C \cup B_C \cup C_B \cup C_C \rangle$$
$$\theta_8 = \langle B_B, A_A \cup A_B \cup A_D \cup B_A \cup C_A \mid A_C \cup B_C \cup C_B \cup C_C \rangle$$

We note that the set J^* introduced in the definition above includes all triplets which potentially result from applying the GC-operator to the triplet pair θ_1, θ_2. The GC^+-operator takes from this set all valid triplets which are not g-included in θ_1 or θ_2. To accommodate application of the GC^+-operator, the G4* derivation rule used in the Studený–Baioletti algorithm is enhanced to the generalised contraction rule $G4^+$ defined below.

Definition 14. Let $J \subseteq S^{(3)}$ be a semi-graphoid independency model. Then, $G4^+$ is the following derivation rule over J:

$$G4^+: \quad \text{if } \theta_1, \theta_2 \in J \text{ are in strong normal form, then } GC^+(\theta_1, \theta_2) \cup GC^+(\theta_2, \theta_1) \subseteq J$$

Our enhanced algorithm for basis construction now builds upon application of the $G4^+$ derivation rule introduced above and is summarised in Fig. 2. The algorithm takes a starting triplet set J for its input, from which it identifies, through a call to the function *Lonely* in a pre-processing step, all triplets which can be safely set aside during the basis computation, as described in Section 4.1. It then starts the basis construction with the (possibly reduced) initial basis J_0. In each iteration,

ENHANCED BASIS CONSTRUCTION

Input: $J \subseteq S^{(3)}$
Output: J_+ with $\overline{J_+} = \overline{J}$

1: **function** *Enhanced-Basis-Computation*(J)
2: $A \leftarrow Lonely(J)$
3: $J_0 \leftarrow J \setminus A$
4: $N_0 \leftarrow J \setminus A$
5: $k \leftarrow 0$
6: **repeat**
7: $k \leftarrow k + 1$
8: $N_k \leftarrow \bigcup_{\theta_1 \in J_{k-1}, \theta_2 \in N_{k-1}} (GC^+(\theta_1, \theta_2) \cup GC^+(\theta_1, \theta_2))$
9: $J_k \leftarrow (J_{k-1} \cup N_k)_{/\sqsubseteq n}$
10: **until** $J_k \sqsubseteq J_{k-1}$
11: **return** $J_+ = J_k \cup A$
12: **end function**

Fig. 2. Our enhanced algorithm for basis construction.

it computes the triplets which result from all possible applications of the G4$^+$ derivation rule (line 8) and adds these to the current triplet set. The algorithm establishes the intermediate basis for the next iteration by taking a maximal non-symmetric subset of the just constructed triplet set (line 9). The algorithm then continues iteratively, until the newly established basis is g-included in the intermediate basis from the previous iteration (line 10). We would like to note that for deciding upon stopping the main iterative loop it no longer suffices to verify equality as in the Studený–Baioletti algorithm: since taking a maximal non-symmetric subset involves a random choice among a triplet and its transpose, two successive intermediate bases J_{k-1} and J_k may differ by just symmetric transposes. In its final step, the algorithm includes the lonely triplets which were set aside before the actual basis construction commenced (line 11).

Building upon the correctness of the Studený–Baioletti algorithm for basis construction, we prove correctness of our enhanced algorithm through the following lemma.

Lemma 14. *Let $J \subseteq S^{(3)}$ be a starting triplet set and let $\overline{J}$ be its closure. Let $J^{G4^*}_{/\sqsubseteq}$ be the basis for $\overline{J}$ constructed by the Studený–Baioletti algorithm and let J_+ be the basis yielded by our algorithm. Then, $J_+ \sqsubseteq J^{G4^*}_{/\sqsubseteq}$ and $J^{G4^*}_{/\sqsubseteq} \sqsubseteq J_+$.*

Proof. From Lemmas 8 and 9 we know that the lonely triplets from a starting set can always be set aside from the iterative computations and be added after the basis construction has finished, regardless of the algorithm used. For the remainder of the proof, we can thus assume without loss of generality that the starting triplet set J does not include any lonely triplets.

Starting with the triplet sets J_0 and N_0, our enhanced algorithm constructs, in each iteration $k \geq 1$, a triplet set N_k of (potentially) new triplets through application of the G4$^+$ derivation rule; it then selects a maximal non-symmetric subset J_k from the triplet set $J_{k-1} \cup N_k$. Similarly, the Studený–Baioletti algorithm constructs, in each iteration, a triplet set N'_k through application of the G4* rule and selects a g-maximal subset J'_k from the triplet set $J'_{k-1} \cup N'_k$. We now first prove that $J^{G4^*}_{/\sqsubseteq} \sqsubseteq J_+$. To this end, we show by induction that, for any $i \in \mathbb{N}$, we have that $N'_i \sqsubseteq N_i$ and $J'_i \sqsubseteq J_i$. Clearly, the two stated properties hold for $i = 0$. For our induction hypothesis, we suppose that the two properties hold for all $i = 0, \ldots, k-1$. We show that the properties then also hold for $i = k$. By the induction hypothesis, we have that for all pairs $\theta'_1 \in J'_{k-1}$, $\theta'_2 \in N'_{k-1}$, there exists a pair $\theta_1 \in J_{k-1}$, $\theta_2 \in N_{k-1}$ such that $\theta'_1 \sqsubseteq \theta_1$ and $\theta'_2 \sqsubseteq \theta_2$. Now, let $\tau' = \mathrm{gc}(\theta'_1, \theta'_2)$ and $\tau = \mathrm{gc}(\theta_1, \theta_2)$. We then find from Lemma 5 that $\tau' \sqsubseteq \tau$. We thus have that $GC(\theta'_1, \theta'_2) \sqsubseteq GC(\theta_1, \theta_2)$. We similarly find that $GC(\theta'_2, \theta'_1) \sqsubseteq GC(\theta_2, \theta_1)$ from which we have that $N'_k \sqsubseteq N_k$. By the induction hypothesis, we also have that $J'_{k-1} \sqsubseteq J_{k-1}$, from which we find that $(J'_{k-1} \cup N'_k) \sqsubseteq (J_{k-1} \cup N_k)$ and, by Lemma 11, that

$$J'_k = (J'_{k-1} \cup N'_k)_{/\sqsubseteq} \sqsubseteq (J'_{k-1} \cup N'_k)_{/\sqsubseteq n} \sqsubseteq (J_{k-1} \cup N_k)_{/\sqsubseteq n} = J_k$$

From the induction, we conclude that, for any $i \in \mathbb{N}$, we have $N'_i \sqsubseteq N_i$ and $J'_i \sqsubseteq J_i$.

Now suppose that the iteration of the main loop of the two algorithms was to continue until both algorithms had halted. Since the starting set J is finite, there must be an integer n such that $J'_n = J'_{n-1} = J^{G4^*}_{/\sqsubseteq}$ and $J_n \sqsubseteq J_{n-1}$ with $J_n = J_+$. From $J'_{k-1} \sqsubseteq J_{k-1}$ for all iterations k, we conclude that $J^{G4^*}_{/\sqsubseteq} \sqsubseteq J_+$. To complete the proof, we recall from Lemma 7 that $\overline{J}_{/\sqsubseteq} \sqsubseteq J^{G4^*}_{/\sqsubseteq}$. Since $J_+ \sqsubseteq \overline{J}$, we find that $J_+ \sqsubseteq \overline{J} \sqsubseteq \overline{J}_{/\sqsubseteq} \sqsubseteq J^{G4^*}_{/\sqsubseteq}$. □

In Section 3 we briefly discussed the worst-case runtime complexity of the Studený–Baioletti algorithm for basis construction and argued that this algorithm can take exponential time in the size of the starting triplet set. As it builds upon essentially the same principles, our enhanced algorithm shares this high worst-case complexity with the Studený–Baioletti algorithm. We further argued that each iteration of the Studený–Baioletti algorithm takes polynomial time and that, in practical applications, the power of the polynomial involved is dependent on the sizes of the intermediate bases constructed in earlier iterations. By the following lemma, we will now show that all intermediate bases constructed by our enhanced algorithm cannot be larger in size than those constructed by the Studený–Baioletti algorithm. As a consequence, each iteration

of our algorithm takes at most the same runtime as the matching iteration of the Studený–Baioletti algorithm. Without any loss of generality, we consider in the following lemma only starting sets without any lonely triplets,

Lemma 15. *Let $J \subseteq S^{(3)}$ be a triplet set without lonely triplets. Let J'_k be the intermediate basis constructed by the Studený–Baioletti algorithm and J_k be the intermediate basis constructed by our algorithm, in the kth iteration of the respective algorithm. Then, $J_k \subseteq J'_k$ for all $k \geq 0$.*

Proof. For the first iteration, our algorithm starts with the triplet sets J_0 and $N_0 = J_0$; similarly, the Studený–Baioletti algorithm starts with the sets $J'_0 = J_0$ and $N'_0 = N_0$. We now consider the triplet sets

$$N_1 = \bigcup_{\theta_1,\theta_2 \in J_0} (GC^+(\theta_1,\theta_2) \cup GC^+(\theta_2,\theta_1))$$

$$N'_1 = \bigcup_{\theta_1,\theta_2 \in J'_0} (GC(\theta_1,\theta_2) \cup GC(\theta_2,\theta_1))$$

constructed by the two algorithms, respectively. For any valid triplet $\theta \in N_1 \cup N'_1$ constructed by applying the *GC*-operator or the GC^+-operator to the triplet pair $\theta_1, \theta_2 \in J_0$, either one of the following two properties holds:

- if θ is not g-included in θ_1 or θ_2, then $\theta \in N_1$ and $\theta \in N'_1$;
- if θ is g-included in either θ_1 or θ_2, then $\theta \notin N_1$ and $\theta \in N'_1$.

We now distinguish between the following three subsets of the set N'_1 constructed by the Studený–Baioletti algorithm:

$$E_1^g = \bigcup_{\theta_1,\theta_2 \in J'_0} \{\theta \mid \theta \in GC(\theta_1,\theta_2) \cup GC(\theta_2,\theta_1) \text{ and } \theta \sqsubseteq \theta_1 \text{ or } \theta \sqsubseteq \theta_2\}$$

$$E_1^t = \{\theta \mid \exists \theta' \in (J'_0 \cup N'_1) \setminus \{\theta\} \text{ such that } \theta = \theta'^T\}$$

$$M'_1 = N'_1 \setminus (E_1^g \cup E_1^t)$$

We note that the two sets E_1^g and E_1^t may be empty and may overlap. We further note that the set E_1^g includes only triplets θ which are constructed through application of the *GC*-operator to a triplet pair θ_1, θ_2 and are g-included in either one of these triplets. Such triplets θ are not generated by our algorithm, by the definition of the GC^+-operator. All other triplets constructed through application of the *GC*-operator to a pair of triplets from the starting set $J'_0 = J_0$ are also constructed by the GC^+-operator. We conclude that $N_1 = M'_1 \cup E_1^t \subseteq M'_1 \cup E_1^g \cup E_1^t = N'_1$.

We now observe that the Studený–Baioletti algorithm constructs an intermediate basis J'_1 by taking a g-maximal subset of the set $J'_0 \cup N'_1$ as follows:

$$J'_1 \;=\; (J'_0 \cup N'_1)_{/\sqsubseteq} \;=\; (J'_0 \cup M'_1 \cup E_1^g \cup E_1^t)_{/\sqsubseteq} \;=\; (J'_0 \cup M'_1 \cup E_1^t)_{/\sqsubseteq}$$

that is, by taking a g-maximal subset, the Studený–Baioletti effectively removes the g-included extra triplets of the set E_1^g from the intermediate basis being constructed. We note that the extra symmetric transposes of the set E_1^t may not be all removed from the intermediate basis under construction by taking a g-maximal subset of $J'_0 \cup N'_1$.

If the triplet set E_1^t is non-empty, then we know by its definition that each triplet θ in the set is one of a pair of transposed triplets θ, θ^T with $\theta^T \in J'_0 \cup N'_1$. We now distinguish between two types of such triplets:

- for each triplet $\theta \in E_1^t$ which is g-included in a triplet $\theta' \in J'_0 \cup N'_1$ with $\theta' \notin \{\theta, \theta^T\}$, we have that both θ and θ^T are removed from the intermediate basis under construction by taking the g-maximal subset $(J'_0 \cup N'_1)_{/\sqsubseteq}$;
- for each triplet $\theta \in E_1^t$ which is g-maximal in the set $J'_0 \cup N'_1$, we find that taking a g-maximal subset of $J'_0 \cup N'_1$ will retain both θ and θ^T.

Based upon these observations, we now write the set E_1^t of extra transposes as $E_1^t = E_1^{ti} \cup E_1^{tm}$, where E_1^{ti} consists of the g-included triplets mentioned above and E_1^{tm} consists of the g-maximal triplets from E_1^t. We then find that

$$J'_1 \;=\; (J'_0 \cup M'_1 \cup E_1^t)_{/\sqsubseteq} \;=\; (J'_0 \cup M'_1 \cup E_1^{tm})_{/\sqsubseteq}$$

for the intermediate basis with which the next iteration of the Studený–Baioletti algorithm will commence.

Our enhanced algorithm constructs its intermediate basis J_1 by taking a maximal non-symmetric subset of the set $J_0 \cup N_1$ as follows:

$$J_1 \;=\; (J_0 \cup N_1)_{/\sqsubseteq n} \;=\; (J_0 \cup M'_1 \cup E_1^{tm})_{/\sqsubseteq n}$$

The Studený–Baioletti algorithm

k	J_{k-1}	N_k	J_k
1	3	1	3
2	3	3	2
3	2	3	2

Our enhanced algorithm

k	J_{k-1}	N_k	J_k
1	3	1	3
2	3	1	2
3	2	0	2

Fig. 3. Some statistics from running the Studený–Baioletti algorithm and our enhanced algorithm for basis construction, for the starting set from Example 1.

By taking a maximal non-symmetric subset of the set $J_0 \cup M'_1 \cup E_1^{tm}$, our algorithm effectively removes one triplet from each pair of transposed triplets θ, θ^T. As it randomly chooses the triplet to be removed from each pair, we may assume without loss of generality that it removes all triplets from the set E_1^{tm}. We can thus conclude that

$$J_1 \;=\; (J_0 \cup N_1)_{/\sqsubseteq n} \;=\; (J_0 \cup M'_1)_{/\sqsubseteq n}$$

From the observation that $J_1 \subseteq J'_1$ we have that the intermediate basis J_1 with which our algorithm will commence the next iteration, cannot be larger in size than the intermediate basis J'_1 used in the Studený–Baioletti algorithm.

For the next iteration, our algorithm starts with the triplet sets $J_1 = (J_0 \cup M'_1)_{/\sqsubseteq n}$ and $N_1 = M'_1 \cup E_1^t$; similarly, the Studený–Baioletti algorithm starts with the sets $J'_1 = (J'_0 \cup M'_1 \cup E_1^{tm})_{/\sqsubseteq}$ and $N'_1 = M'_1 \cup E_1^g \cup E_1^t$. The Studený–Baioletti algorithm thus (possibly) includes extra symmetric transposes in its set J'_1 when compared to the set J_1 used by our algorithm; it further (possibly) includes extra g-included triplets in its set N'_1. Since these extra triplets are all g-included in the set J_1, we know that the extra applications of the GC-operator by the Studený–Baioletti algorithm cannot produce any new triplets which are not yet covered by application of the GC^+-operator in our enhanced algorithm. Since the same argument holds for all subsequent iterations by the two algorithms, we conclude that the property stated in the lemma holds. □

From the previous lemma, we have that all intermediate bases constructed by our enhanced algorithm cannot be larger in size than those constructed by the Studený–Baioletti algorithm. More specifically, the proof of the lemma provides the necessary ingredients to demonstrate that the basis resulting from our enhanced algorithm is minimal in the sense that it no longer includes any redundant triplets. The basis constructed by our algorithm not necessarily is a basis of minimum size, however; we will return to this latter observation in our examples in the next section.

5.2. Examples

We detail three examples to illustrate the differences and similarities between the two algorithms for basis construction discussed in the present paper. For ease of presentation, the examples are kept small and simple. From the illustrated differences between the two algorithms however, the potential of our enhancements is readily envisioned for examples of more realistic size.

Our first example serves to illustrate that the two algorithms can yield the exact same basis for a starting triplet set.

Example 1

We consider a starting set $J = J_0$ composed of the following three triplets:

$$\langle\{1,2\},\{4\} \mid \{3,5\}\rangle, \quad \langle\{2\},\{4\} \mid \{3\}\rangle, \quad \langle\{1\},\{5\} \mid \{2,3\}\rangle$$

and compute a basis for the semi-graphoid independency model $\overline{J}$ of J, by means of both algorithms.

From the starting set, the Studený–Baioletti algorithm constructs the set N_1 of newly generated triplets, which is comprised of just a single element:

$$\langle\{1\},\{4,5\} \mid \{2,3\}\rangle$$

Since this triplet g-includes the triplet $\langle\{1\},\{5\} \mid \{2,3\}\rangle$ from the starting set, the algorithm constructs the intermediate basis $J_1 = (J_0 \cup N_1)_{/\sqsubseteq}$ to include the following three triplets:

$$\langle\{1,2\},\{4\} \mid \{3,5\}\rangle, \quad \langle\{2\},\{4\} \mid \{3\}\rangle, \quad \langle\{1\},\{4,5\} \mid \{2,3\}\rangle$$

In the next iteration, the algorithm generates the triplet set N_2 with the following three elements:

$$\langle\{4\},\{1,2\} \mid \{3,5\}\rangle, \quad \langle\{1\},\{4,5\} \mid \{2,3\}\rangle, \quad \langle\{4\},\{1,2\} \mid \{3\}\rangle$$

The algorithm continues and, after the third iteration, returns a basis for $\overline{J}$ consisting of the following two triplets:

$$\langle\{4\},\{1,2\} \mid \{3\}\rangle, \quad \langle\{1\},\{4,5\} \mid \{2,3\}\rangle$$

Fig. 3 summarises, on the left, the numbers of triplets included in the sets J_{k-1}, N_k and J_k after each iteration k of the Studený–Baioletti algorithm.

The Studený–Baioletti algorithm

k	J_{k-1}	N_k	J_k
1	3	2	5
2	5	4	6
3	6	4	6

Our enhanced algorithm

k	J_{k-1}	N_k	J_k
1	3	2	5
2	5	3	5

Fig. 4. Some statistics from running the Studený–Baioletti algorithm and our enhanced algorithm for basis construction, for the starting set from Example 2.

Our enhanced algorithm returns the same basis for the model $\overline{J}$ defined by J, and in fact constructs the same intermediate basis in each iteration. The sets N_2 and N_3 constructed by our algorithm are smaller than those constructed by the Studený–Baioletti algorithm, however. In the second iteration for example, the Studený–Baioletti algorithm constructs a set N_2 of three triplets, while the set N_2 constructed by our algorithm includes just a single triplet. We note that the triplet $\theta = \langle\{4\}, \{1, 2\} \mid \{3, 5\}\rangle$, for example, from the set N_2 constructed by the Studený–Baioletti algorithm is the result of applying the *GC*-operator to the two triplets θ_1^T and θ_2^T where $\theta_1 = \langle\{1, 2\}, \{4\} \mid \{3, 5\}\rangle \in J_1$ and where $\theta_2 = \langle\{1\}, \{4, 5\} \mid \{2, 3\}\rangle \in N_1$; as the triplet θ is g-included in the triplet θ_1^T, it is not added to the set N_2 by our algorithm. Fig. 3 summarises, on the right, the numbers of triplets included in the sets J_{k-1}, N_k and J_k after each iteration k of our enhanced algorithm. We note that since the starting triplet set does not include any lonely triplets, our algorithm involved all starting triplets throughout the computations. □

Our second example serves to show that our enhanced algorithm can in fact return a smaller basis than the Studený–Baioletti algorithm for basis construction.

Example 2

We now consider a starting set $J = J_0$ composed of the following triplets:

$$\langle\{1\}, \{2, 5\} \mid \{3, 4\}\rangle, \quad \langle\{1\}, \{4\} \mid \{5\}\rangle, \quad \langle\{1, 2\}, \{4\} \mid \{3, 5\}\rangle$$

and again compute a basis for the model $\overline{J}$ of J, by means of both algorithms.

From the starting triplet set, the Studený–Baioletti algorithm constructs the set N_1 with the following two triplets:

$$\langle\{1\}, \{2, 4\} \mid \{3, 5\}\rangle, \quad \langle\{2\}, \{1, 4\} \mid \{3, 5\}\rangle$$

Since the set $J_0 \cup N_1$ does not include any g-included triplets, the new basis under construction equals $J_1 = J_0 \cup N_1$ with $|J_1| = 5$. In the next iteration, the algorithm constructs the set N_2 to include the four triplets:

$$\langle\{1\}, \{2, 5\} \mid \{3, 4\}\rangle, \quad \langle\{1\}, \{2, 4\} \mid \{3, 5\}\rangle,$$
$$\langle\{2\}, \{1, 4\} \mid \{3, 5\}\rangle, \quad \langle\{4\}, \{1, 2\} \mid \{3, 5\}\rangle$$

from which it finds a basis of size 6 which is returned after the third iteration:

$$\langle\{1\}, \{4\} \mid \{5\}\rangle, \quad \langle\{1, 2\}, \{4\} \mid \{3, 5\}\rangle, \quad \langle\{1\}, \{2, 5\} \mid \{3, 4\}\rangle,$$
$$\langle\{2\}, \{1, 4\} \mid \{3, 5\}\rangle, \quad \langle\{4\}, \{1, 2\} \mid \{3, 5\}\rangle, \quad \langle\{1\}, \{2, 4\} \mid \{3, 5\}\rangle$$

In Fig. 4, we report, on the left, the numbers of triplets included in the sets J_{k-1}, N_k and J_k for each iteration k of the Studený–Baioletti algorithm.

Our enhanced algorithm finds a smaller basis, of size 5, which it returns after the second iteration. Close inspection of the result returned by the Studený–Baioletti algorithm shows that the final basis includes a triplet and its symmetric transpose, that is, it includes both $\langle\{1, 2\}, \{4\} \mid \{3, 5\}\rangle$ and $\langle\{4\}, \{1, 2\} \mid \{3, 5\}\rangle$. Our algorithm forestalls the generation of the transpose in the second iteration by constructing a set N_1 of size 3: compared to the set generated by the Studený–Baioletti algorithm, the g-included triplet $\langle\{1\}, \{2, 5\} \mid \{3, 4\}\rangle$ is not added to N_1. Since now no additional g-maximal triplets are generated, our algorithm terminates after the second iteration. In Fig. 4, we report, on the right, the numbers of triplets included in the sets J_{k-1}, N_k and J_k for each iteration k of our enhanced algorithm. We note that since the starting triplet set does not include any lonely triplets, the algorithm could not set aside any triplets during the computations. We further note that both algorithms return a basis for the model $\overline{J}$ which is larger in size than the original starting set J_0 defining the model. The example thereby demonstrates that also the basis returned by our enhanced algorithm may not be of minimum size. □

Our final example demonstrates the effect of lonely triplets in a starting triplet set.

Example 3

We consider a starting set $J = J_0$ composed of the following four triplets:

$$\langle\{5\}, \{6\} \mid \varnothing\rangle, \quad \langle\{2, 3\}, \{1, 4\} \mid \{5\}\rangle,$$
$$\langle\{1, 2\}, \{3, 4\} \mid \{5\}\rangle, \quad \langle\{3\}, \{1, 4\} \mid \{2, 5\}\rangle$$

and compute a basis for the semi-graphoid model $\overline{J}$ of J, by means of both algorithms.

The Studený–Baioletti algorithm

k	J_{k-1}	N_k	J_k
1	4	8	8
2	8	13	11
3	11	13	11

Our enhanced algorithm

k	J_{k-1}	N_k	J_k
1	3	5	6
2	6	12	7
3	7	13	7

Fig. 5. Some statistics from running the Studený–Baioletti algorithm and our enhanced algorithm for basis construction, for the starting set from Example 3.

From the starting triplet set, the Studený–Baioletti algorithm constructs the set N_1 with the following eight triplets:

$$\langle\{2\},\{1,3,4\} \mid \{5\}\rangle,\quad \langle\{1\},\{2,3,4\} \mid \{5\}\rangle,\quad \langle\{1,4\},\{2,3\} \mid \{5\}\rangle,$$
$$\langle\{3\},\{1,4\} \mid \{2,5\}\rangle,\quad \langle\{4\},\{1,3\} \mid \{2,5\}\rangle,\quad \langle\{1\},\{3,4\} \mid \{2,5\}\rangle,$$
$$\langle\{3\},\{1,2,4\} \mid \{5\}\rangle,\quad \langle\{4\},\{1,2,3\} \mid \{5\}\rangle$$

The set $J_0 \cup N_1$ now includes several g-included triplets; more specifically, the triplets $\langle\{3\},\{1,4\} \mid \{2,5\}\rangle$ (with two occurrences), $\langle\{1\},\{3,4\} \mid \{2,5\}\rangle$ and $\langle\{4\},\{1,3\} \mid \{2,5\}\rangle$ are g-included in the triplets $\langle\{3\},\{1,2,4\} \mid \{5\}\rangle$, $\langle\{1\},\{2,3,4\} \mid \{5\}\rangle$ and $\langle\{4\},\{1,2,3\} \mid \{5\}\rangle$, respectively. The intermediate basis J_1 is constructed to include eight triplets as a consequence. In the next iteration, the algorithm constructs the set N_2 to include the following 13 triplets:

$$\langle\{3,4\},\{1,2\} \mid \{5\}\rangle,\quad \langle\{2,4\},\{1,3\} \mid \{5\}\rangle,\quad \langle\{1,4\},\{2,3\} \mid \{5\}\rangle,$$
$$\langle\{2\},\{1,3,4\} \mid \{5\}\rangle,\quad \langle\{1,3\},\{2,4\} \mid \{5\}\rangle,\quad \langle\{1\},\{2,3,4\} \mid \{5\}\rangle,$$
$$\langle\{2,3\},\{1,4\} \mid \{5\}\rangle,\quad \langle\{1\},\{3,4\} \mid \{2,5\}\rangle,\quad \langle\{4\},\{1,3\} \mid \{2,5\}\rangle,$$
$$\langle\{3\},\{1,4\} \mid \{2,5\}\rangle,\quad \langle\{3\},\{1,2,4\} \mid \{5\}\rangle,\quad \langle\{4\},\{1,2,3\} \mid \{5\}\rangle,$$
$$\langle\{1,2\},\{3,4\} \mid \{5\}\rangle$$

from which it finds a basis of size 11, which is returned after the third iteration:

$$\langle\{5\},\{6\} \mid \varnothing\rangle,\quad \langle\{2,4\},\{1,3\} \mid \{5\}\rangle,\quad \langle\{1,4\},\{2,3\} \mid \{5\}\rangle,$$
$$\langle\{2\},\{1,3,4\} \mid \{5\}\rangle,\quad \langle\{1,3\},\{2,4\} \mid \{5\}\rangle,\quad \langle\{1\},\{2,3,4\} \mid \{5\}\rangle,$$
$$\langle\{2,3\},\{1,4\} \mid \{5\}\rangle,\quad \langle\{3\},\{1,2,4\} \mid \{5\}\rangle,\quad \langle\{3,4\},\{1,2\} \mid \{5\}\rangle,$$
$$\langle\{1,2\},\{3,4\} \mid \{5\}\rangle,\quad \langle\{4\},\{1,2,3\} \mid \{5\}\rangle$$

In Fig. 5, we report, on the left, the numbers of triplets included in the sets J_{k-1}, N_k and J_k for each iteration k of the Studený–Baioletti algorithm.

Our enhanced algorithm starts with identifying the lonely triplet $\langle\{5\},\{6\} \mid \varnothing\rangle$ and setting it aside from its iterations. It then computes a basis of size 7 from the reduced starting set and returns a final basis of size 8 after including the lonely triplet. Closer inspection of the final basis returned by the Studený–Baioletti algorithm shows that it includes several pairs of symmetric transposes; in fact, the triplets $\langle\{1,4\},\{2,3\} \mid \{5\}\rangle$, $\langle\{2,4\},\{1,3\} \mid \{5\}\rangle$ and $\langle\{3,4\},\{1,2\} \mid \{5\}\rangle$ are symmetric transposes of the triplets $\langle\{2,3\},\{1,4\} \mid \{5\}\rangle$, $\langle\{1,3\},\{2,4\} \mid \{5\}\rangle$ and $\langle\{1,2\},\{3,4\} \mid \{5\}\rangle$, respectively. Our algorithm identifies these transposed pairs during its computations and removes one triplet from each such pair. As a consequence of the incorporated enhancements, our algorithm constructs and employs smaller triplet sets in the various iterations of its main loop than the Studený–Baioletti algorithm. These smaller set sizes translate into a smaller computational burden for our algorithm. Focusing on just the computation of the three sets N_i, $i = 1, 2, 3$, for example, the Studený–Baioletti algorithm investigates as many as 223 pairs of triplets, while 123 triplet pairs are investigated by our enhanced algorithm for basis construction. Fig. 5 reports, on the right, the numbers of triplets included in the sets J_{k-1}, N_k and J_k, after removal of the single lonely triplet from J_0, for each iteration k of our enhanced algorithm. □

6. Conclusions and further research

We revisited the representation of semi-graphoid independency models and identified properties which we exploited for enhancing currently existing algorithms for basis construction. Our first enhancement is based on the observation that particular elements from a starting triplet set can be excluded from consideration during basis computation; we stated necessary conditions for identifying such lonely triplets. We further introduced the notion of maximal non-symmetric triplet subset, which allowed the removal of symmetric transposes of triplets from a basis under construction. Building upon these properties, we presented an enhanced algorithm for computing a non-symmetric basis for a closure defined by a starting set of triplets. By means of several examples, we illustrated the potential of our enhancements by comparing the bases constructed by our algorithm with the results obtained from currently existing algorithms.

In our future research, we will study the representation of other established families of independency models, such as the family of graphoid models and the family of models with identified stable independencies. We will also investigate the potential of our notions of lonely triplet and non-symmetric basis for the construction of graphical representations of independency models.

References

[1] M. Baioletti, G. Busanello, B. Vantaggi, Conditional independence structure and its closure: inferential rules and algorithms, Int. J. Approx. Reason. 50 (7) (2009) 1097–1114.
[2] A.P. Dawid, Conditional independence in statistical theory (with discussion), J. R. Stat. Soc. B 41 (1979) 1–31.
[3] A.P. Dawid, Some misleading arguments involving conditional independence, J. R. Stat. Soc. B 41 (1979) 249–252.
[4] D. Geiger, J. Pearl, Logical and algorithmic properties of conditional independence and graphical models, Ann. Stat. 21 (1993) 2001–2021.
[5] F.V. Jensen, T.D. Nielsen, Bayesian Networks and Decision Graphs, Information Science and Statistics, Springer, 2007.
[6] S.L. Lauritzen, Graphical Models, vol. 17, Oxford University Press, USA, 1996.
[7] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann, 1988.
[8] J. Pearl, A. Paz, Graphoids: a graph-based logic for reasoning about relevance relations, in: Advances in Artificial Intelligence 2, North-Holland, 1985.
[9] W. Spohn, Stochastic independence, causal independence, and shieldability, J. Philos. Log. 9 (1980) 73–99.
[10] W. Spohn, On the properties of conditional independence, in: P. Humphreys (Ed.), Patrick Suppes: Scientific Philosopher. Vol. 1: Probability and Probabilistic Causality, Kluwer, Dordecht, 1994, pp. 173–194.
[11] M. Studeny, Conditional independence relations have no finite complete characterization, in: S. Kubik, J.A. Visek (Eds.), Information Theory, Statistical Decision Functions and Random Processes. Transactions of the 11th Prague Conference, Kluwer, Dordecht, 1992, pp. 377–396.
[12] M. Studenỳ, Semigraphoids and structures of probabilistic conditional independence, Ann. Math. Artif. Intell. 21 (1) (1997) 71–98.
[13] M. Studenỳ, Complexity of structural models, in: Prague Conference on Information Theory, Statistical Decision Functions and Random Processes, 1998.
[14] S.K.M. Wong, C.J. Butz, D. Wu, On the implication problem for probabilistic conditional independency, IEEE Trans. Syst. Man Cybern., Part A, Syst. Hum. 30 (6) (2000) 785–805.