

Алгебраические байесовские сети: проверка магистральной связности*

А. Г. Максимов¹, А. Л. Тулупьев^{1,2}

¹ Санкт-Петербургский федеральный исследовательский центр РАН,
Российская Федерация, 199178, 14-я линия В. О., 39

² Санкт-Петербургский государственный университет,
Российская Федерация, 199034, Санкт-Петербург, Университетская наб., 7–9

Для цитирования: Максимов А. Г., Тулупьев А. Л. Алгебраические байесовские сети: проверка магистральной связности // Вестник Санкт-Петербургского университета. Математика. Механика. Астрономия. 2021. Т. 8 (66). Вып. 2. С. 305–316.
<https://doi.org/10.21638/spbu01.2021.210>

В работе исследуется одна из задач, возникающих при машинном обучении баз фрагментов знаний с неопределенностью, представленных в виде алгебраических байесовских сетей — построение графа смежности как глобальной структуры сети по ее первичной структуре. Цель исследования заключается в предложении методов решения обратной задачи. В качестве результатов предложены алгоритмы проверки графа на принадлежность семейству графов смежности и семейству минимальных графов смежности, сделаны оценки их вычислительной сложности. Для алгоритма проверки принадлежности семейству графов смежности также предложена улучшенная версия для частного случая и улучшение для общего случая в среднем. Вопрос распознавания графов смежности ранее не исследовался, в текущей формулировке ставится и решается впервые. Теоретическая значимость заключается в возможностях для применения результатов в дальнейших исследованиях теоретико-графовых инвариантов в глобальных структурах алгебраических байесовских сетей.

Ключевые слова: алгебраические байесовские сети, граф смежности, минимальный граф смежности, алгоритмы на графах, сложность алгоритмов.

1. Введение. Одной из задач, решаемых в рамках работы с моделями машинного обучения, естественным образом является задача непосредственного обучения модели на имеющихся данных. В случае алгебраических байесовских сетей (АБС) [1], относящихся к классу логико-вероятностных графических моделей баз фрагментов знаний с неопределенностью, среди прочих выделяется задача обучения глобальной (или вторичной) структуры сети по известной первичной структуре [1, 2]. Первичная структура АБС — это набор фрагментов знаний, где каждый фрагмент знаний является идеалом конъюнктов со скалярными или интервальными оценками вероятности истинности [3]. Вторичная структура алгебраической байесовской сети — это совокупность связей между фрагментами знаний, представленная в виде графа смежности.

Одной и той же первичной структуре может соответствовать несколько графов смежности [4]. Более того, одной и той же первичной структуре может соответство-

*Работа выполнена в рамках государственного задания СПб ФИЦ РАН (проект № 0073-2019-0003).

© Санкт-Петербургский государственный университет, 2021

вать несколько минимальных графов смежности [4]. Отсюда возникает, в частности, потребность изучить семейства графов смежности, сформированных над одним и тем же набором фрагментов знаний. Однако сейчас известен лишь один инвариант этого семейства — число ребер [5].

Таким образом, актуальна серия исследований, направленных на изучение семейства графов смежности как подмножества всех графов, а также нередуцируемых и минимальных графов смежности, и выявление в них инвариантов. Предельной целью является построение такой альтернативной структуры, которая хранила бы те же сведения, что и минимальные графы смежности, но была бы однозначно определена (единственна). Кроме того, такой целью является построение критериев магистральности и минимальности, которые бы вычислительно просто проверялись.

Цель данной работы — построение и анализ алгоритмов проверки принадлежности графа семейству графов смежности (проверки магистральной связности графа), магистральной связности дерева, минимальности графа смежности. Вопрос распознавания графов смежности ранее не исследовался, в текущей формулировке ставится и решается впервые. Теоретическая значимость работы состоит в возможности дальнейшего применения результатов в исследованиях теоретико-графовых свойств минимальных графов смежности.

Поясним также необходимость магистральной связности в теории АБС. После формирования сети мы можем получить какую-то дополнительную информацию, называемую *свидетельством* (например, пошел дождь — тогда вероятность истинности переменной «пошел дождь» должна теперь оцениваться единицей). Изменение оценки вероятности истинности одной из переменных влечет за собой изменение оценок истинности всех остальных. Процесс обновления оценок при поступлении свидетельства называется апостериорным выводом. При апостериорном выводе влияние свидетельства, поступившего в один из фрагментов знаний, распространяется по графу смежности, и именно магистральная связность гарантирует, что влияние распространится корректно.

2. Релевантные работы. Алгебраические байесовские сети относятся к классу вероятностных графических моделей [6], которые также включают в себя байесовские сети доверия и марковские сети [7] и позволяют работать с неточными, неполными и нечисловыми данными [8]. Исследования вероятностных графических моделей широко встречаются в современной математике. Например, в недавней работе Ф. Йенсена была предложена эффективная версия ПК-алгоритма [9], в статье [10] байесовские сети рассматриваются как перспективный подход для мультимодального обучения, а в [11] марковские сети применяются в задаче распознавания образов. Также байесовские сети очень хорошо подходят для задач оценки надежности [12]. Подробный обзор применения вероятностных моделей в задачах искусственного интеллекта можно найти в [13]. За исследования частного случая марковских сетей — модели Изинга — в 2010 г. С. Смирнов получил медаль Филдса [14].

Напомним, что одной и той же первичной структуре АБС может соответствовать несколько графов смежности. Для реализации некоторых алгоритмов логико-вероятностного вывода (проверка и поддержание непротиворечивости АБС, апостериорного вывода) выбор графа смежности существенен. Причем наиболее удачным выбором графа смежности для формирования вторичной структуры АБС является ациклический граф смежности или, иными словами, дерево смежности [3]. В работах [1, 2] подробно описаны алгоритмы синтеза одного минимального графа

смежности и всех минимальных графов смежности, однако обратная задача — проверки, что заданный заранее граф является графом смежности, рассматривается впервые. В целом подобный подход рассмотрения обратной идентификационной задачи встречается достаточно часто [15].

В работе также используются некоторые базовые алгоритмы, такие как система непересекающихся множеств и обход в глубину. Их подробное описание можно найти в монографии [16].

3. Постановка задачи. Прежде чем приступать к описанию разработанных алгоритмов, необходимо дать формальное описание графов смежности и решаемых задач. Содержательно, вторичной структурой алгебраической байесовской сети является граф смежности, то есть неориентированный нагруженный (у каждой вершины есть нагрузка — некоторое множество) граф, удовлетворяющий следующим условиям:

- любые две вершины, имеющие непустое пересечение нагрузок, соединены путем;
- в нагрузки всех вершин этого пути входят все элементы, общие для его начальной и конечной вершин.

Эти условия называют свойством *магистральной связности*, а соответствующий путь между двумя вершинами — *магистральным путем* [5].

Для обеспечения дополнительной формализации будем считать, что заданы неориентированный граф $G = \langle V, E \rangle$ и конечный алфавит \mathcal{A} , а также определена и известна функция $\omega : V \rightarrow 2^{\mathcal{A}}$, сопоставляющая каждой вершине графа ее нагрузку — множество элементов \mathcal{A} . Для простоты записи под $v \cap u$ будем понимать $\omega(v) \cap \omega(u)$ — пересечение нагрузок вершин v и u . Нагрузкой ребра по определению считают пересечение нагрузок его концов. Граф будем считать связным. В противном случае разобьем его на компоненты связности и рассмотрим каждую из них отдельно. Необходимо проверить, что данный граф с данными нагрузками вершин является графом смежности, а также является ли он минимальным.

4. Проверка магистральной связности графа. Опишем разработанный алгоритм проверки магистральной связности произвольного графа и докажем его корректность.

4.1. Описание. Для решения поставленной задачи воспользуемся структурой данных disjoint set union (DSU) [16, 17]. Она представляет собой набор непересекающихся множеств и позволяет обрабатывать запросы двух типов: объединить два множества и определить, в каком из множеств лежит данный элемент. Робертом Тарьяном и Яном ван Лейвеном [18], а позже М. Фредманом и М. Саксом [19] было показано, что при совместном использовании ранговой эвристики со сжатием путей такие запросы можно обрабатывать в среднем за $O(\alpha(n))$ на запрос, где $\alpha(n)$ — обратная функция Аккермана, то есть примерно за $O(1)$ на запрос. Для удобства в дальнейшем будем обозначать эту структуру DSU.

Дадим обозначения для ряда функций, определенных в DSU.

`make_set(u)` — создает множество, содержащее элемент u .

`find_set(u)` — определяет, в каком множестве лежит элемент u .

$\text{union_sets}(u, v)$ — объединяет множество, содержащее u , с множеством, содержащим v .

Опишем алгоритм, листинг которого приведен ниже (алгоритм 1).

Алгоритм 1. Алгоритм проверки магистральной связности

Require: V, E, ω

Ensure: \mathbb{T}/\mathbb{F}

```

 $W = \emptyset$ 
for all  $u, v \in V, u \neq v$  do
  if  $\omega_u \cap \omega_v \neq \emptyset$  &  $\omega_u \cap \omega_v \notin W$  then
     $\omega_u \cap \omega_v \rightarrow W$ 
  end if
end for
for all  $w \in W$  do
   $V_w = \emptyset$ 
   $DSU = \emptyset$ 
  for all  $v \in V$  do
    if  $w \subseteq \omega_v$  then
      if  $w = \omega_v$  then
        return  $\mathbb{F}$ 
      else
         $v \rightarrow V_w$ 
         $\text{make\_set}(v)$ 
      end if
    end if
  end for
  for all  $e \in E$  do
    if  $\text{find\_set}(e.\text{first}) = \text{find\_set}(e.\text{second})$  then
       $\text{union\_sets}(e.\text{first}, e.\text{second})$ 
    end if
  end for
  if  $|DSU| \neq 1$  then
    return  $\mathbb{F}$ 
  end if
end for
return  $\mathbb{T}$ 

```

Определим множество сепараторов $W = \{v_i \cap v_j | v_i, v_j \in V, i \neq j\} \setminus \{\emptyset\}$ — множество всех имеющихся попарных пересечений нагрузок вершин, за исключением пустого. Отметим, что нагрузка любого ребра является сепаратором, но не наоборот.

Теперь переберем все элементы множества W . Для каждого $w \in W$ сделаем следующее:

- 1) найдем множество $V_w = \{v \in V | w \subset \omega(v)\}$ — множество вершин, содержащих текущий сепаратор w ;
- 2) создадим DSU и заведем в ней отдельное множество для каждой из вершин из пункта 1, состоящее из этой вершины;
- 3) поочередно переберем ребра графа и, если ребро соединяет две вершины из разных множеств в DSU, будем объединять эти два множества.

Утверждение. *Граф является графом смежности, если и только если на каждом шаге алгоритма (то есть для каждого $w \in W$) все вершины V_w оказались в DSU в одном множестве.*

Лемма. Для любых двух $V_1, V_2 \in V$ и любого $x \subseteq V_1 \cap V_2$ существует путь из V_1 в V_2 , такой что нагрузка любой вершины на этом пути содержит x .

ДОКАЗАТЕЛЬСТВО ЛЕММЫ. $x \subseteq V_1 \cap V_2$. По определению в графе смежности существует путь из V_1 в V_2 , нагрузка любой вершины на котором содержит $V_1 \cap V_2$, но тогда она содержит и x .

ДОКАЗАТЕЛЬСТВО УТВЕРЖДЕНИЯ. Заметим, что на шаге, на котором рассматривается сепаратор w , две вершины лежат в одном множестве в DSU, если и только если между ними есть путь, нагрузка каждой вершины на котором содержит w . Из этого факта и леммы немедленно следует наше утверждение. \square

4.2. Оценка. Пусть $n = |V|$, $m = |E|$. Множество W всех сепараторов в худшем случае имеет размер порядка n^2 . Шаги 1 и 2 можно выполнить за $O(n)$ операций. Шаг 3 за $O(m) = O(n^2)$, поскольку, как уже было сказано, DSU позволяет отвечать на запросы примерно за $O(1)$. Таким образом, итоговая асимптотика $O(n^2(n+n^2)) = O(n^4)$.

На практике многие алгоритмы в среднем работают значительно лучше своей асимптотики [20, 21]. В данном случае для улучшения ситуации предлагается сделать следующее. Будем выходить из шага 3, если все вершины уже лежат в одном множестве в DSU. Докажем следующее утверждение.

Утверждение. Если ребра расположены в случайном порядке (с равномерным распределением на множестве всех перестановок ребер), в среднем это улучшает асимптотику до $O(n^3)$.

Лемма. Пусть даны числа n и k . Средняя длина минимального префикса перестановки n элементов, содержащего все числа от 1 до k , равна $\frac{k}{k+1}(n+1)$.

ДОКАЗАТЕЛЬСТВО ЛЕММЫ. Посчитаем количество перестановок n элементов, в которых искомый префикс имеет длину p . Оно равно $\binom{p-1}{k-1} k!(n-k)!$. Тогда искомое среднее значение равно

$$\sum_{p=k}^n \binom{p-1}{k-1} k!(n-k)! \frac{p}{n!} = \sum_{p=k}^n \frac{(p-1)!}{(k-1)!(p-k)!} \frac{k!(n-k)!}{n!} p = \frac{k(n-k)!}{n!} \sum_{p \geq k} \frac{p!}{(p-k)!}.$$

Приведем это к более удобному виду:

$$\begin{aligned} \frac{k(n-k)!}{n!} \sum_{p \geq k} \frac{p!}{(p-k)!} &= \frac{kk!(n-k)!}{n!} \sum_{p \geq k} \frac{p!}{(p-k)!k!} = \\ &= \frac{kk!(n-k)!}{n!} \sum_{p \geq k} \binom{p}{k} = \frac{kk!(n-k)!}{n!} \sum_{p=0}^{n-k} \binom{k+p}{k}. \end{aligned}$$

Воспользуемся известным тождеством о сумме чисел на диагонали треугольника Паскаля:

$$\sum_{p=0}^r \binom{k+p}{k} = \binom{k+r+1}{k+1}.$$

Получим

$$\begin{aligned} \frac{kk!(n-k)!}{n!} \sum_{p=0}^{n-k} \binom{k+p}{k} &= \frac{kk!(n-k)!}{n!} \binom{k+n-k+1}{k+1} = \\ &= \frac{kk!(n-k)!}{n!} \frac{(n+1)!}{(k+1)!(n-k)!} = \frac{k}{k+1} (n+1). \end{aligned}$$

Лемма. При k , не превосходящем $\sqrt{2n}$, искомая величина не превосходит n .

ДОКАЗАТЕЛЬСТВО ЛЕММЫ. Пусть $k = \sqrt{2n}$. Проверим неравенство $\frac{\sqrt{2n}}{\sqrt{2n+1}}(n+1) \leq n$:

$$\begin{aligned} \frac{\sqrt{2n}}{\sqrt{2n+1}}(n+1) \leq n &\Rightarrow \sqrt{2n}(n+1) \leq n\sqrt{2n} + n \Rightarrow \\ &\Rightarrow 2n(n+1)^2 \leq 2nn^2 + n^2 + 2n^2\sqrt{2n} \Rightarrow \\ &\Rightarrow 2n\sqrt{2n} \geq 2(n+1)^2 - 2n^2 - n \Rightarrow 2n\sqrt{2n} \geq 3n + 2 \Rightarrow 8n^3 \geq (3n+2)^2. \end{aligned}$$

Это неравенство обращается в равенство при $n = 2$, затем левая часть растет быстрее. \square

ДОКАЗАТЕЛЬСТВО УТВЕРЖДЕНИЯ. Заметим, что если наш граф действительно граф смежности, то существует набор из $k-1$ ребра, где $k = |V_w|$, который гарантирует, что все вершины попали в одно множество в DSU, потому что в любом связном графе есть остовное дерево. Максимальное число ребер в графе $m = \frac{n(n-1)}{2}$, тогда $k-1 \leq \sqrt{2m}$. Следовательно, по двум изложенным выше леммам, при расположении ребер в случайном порядке набор, обеспечивающий условие выхода, встречается в среднем не позже, чем через n итераций цикла. Тогда вместо квадратичного времени выполнения цикла 3 мы можем надеяться на примерно линейное, что улучшает асимптотику алгоритма до $O(n^3)$ в среднем. \square

5. Проверка магистральной связности дерева. Опишем разработанный алгоритм проверки магистральной связности дерева и докажем его корректность.

5.1. Описание. Опишем алгоритм, который работает только на деревьях, зато имеет лучшую асимптотику. Алгоритм опирается на тот факт, что в дереве существует ровно один путь между двумя вершинами. Проведем серию несколько измененных поисков в глубину (dfs). В качестве параметра будем передавать в dfs не только текущую вершину, но и нагрузку w ее предка. Для текущей вершины s проверим, содержится ли пересечение ее нагрузки с нагрузкой корня в нагрузке w . Утверждается, что дерево будет деревом смежности, если и только если каждая такая проверка вернет результат true. Докажем это.

\Rightarrow Рассмотрим произвольную проверку. Между текущей вершиной s и корнем s существует путь, все вершины в котором содержат $\omega_c \cap \omega_s$. Но тогда и нагрузка предка w текущей вершины содержит это пересечение, так как в дереве между двумя вершинами существует только один путь.

\Leftarrow Рассмотрим произвольную пару вершин s и t . Пусть $u := \omega_u \cap \omega_v$. Пусть $P = s, p_1, \dots, p_k, t$ — путь между s и t . Рассмотрим dfs с корнем s . Тогда $\omega(p_k) \supset u$, так как соответствующая проверка вернула true. Но тогда и $\omega(p_{k-1}) \supset u$, и так далее. Условие магистральной связности выполнено.

Листинг алгоритма приведен ниже (алгоритм 2).

Алгоритм 2. Алгоритм проверки магистральной связности в дереве

Require: V, E, ω

Ensure: \mathbb{T}/\mathbb{F}

```
used(v) = false  $\forall v \in V$ 
function dfs( $v \in V, w \in 2^A$ )
if  $\omega_v \cap \omega_s \not\subseteq w$  then
    return  $\mathbb{F}$ 
end if
used(v) = true
for all  $t : (t, v) \in E$  do
    if !used(t) then
        dfs( $t, \omega_v$ )
    end if
end for
end function
for all  $s \in V$  do
    used = false  $\forall v \in V$ 
    dfs( $s, \omega_s$ )
end for
return  $\mathbb{T}$ 
```

5.2. Оценка. Время работы dfs составляет $O(n + m)$, но так как наш граф является деревом, то есть в нем $m = n - 1$, асимптотика улучшается до $O(n)$. Так как делается целая серия dfs, общее время работы оценивается как $O(n^2)$.

6. Проверка графа смежности на минимальность. Опишем разработанный алгоритм проверки графа смежности на минимальность и докажем его корректность.

6.1. Описание. Граф смежности называется минимальным по включению, если из него нельзя удалить ребро так, чтобы он остался графом смежности. Граф смежности называется минимальным по числу ребер, если не существует графа смежности над той же первичной структурой, который содержал бы меньше ребер. В работе [5] было показано, что эти два свойства минимальности достигаются одновременно.

Опишем алгоритм проверки графа смежности на минимальность, псевдокод которого приведен в алгоритме 3.

Поочередно переберем все ребра. Для каждого ребра $e \in E$ найдем множество $V_e = \{v \in V, \omega_e \subset \omega_v\}$ — множество вершин, содержащих нагрузку ребра e . Затем рассмотрим граф G_e — порожденный подграф G над множеством вершин V_e (за это отвечает функция induction()). Найдем в G_e все мосты (функция find_briges()).

Утверждение. *Граф смежности минимален, если и только если на каждом шаге алгоритма текущее ребро e является мостом в графе G_e .*

Напомним, что мостом называется ребро, при удалении которого увеличивается число компонент связности [22].

ДОКАЗАТЕЛЬСТВО УТВЕРЖДЕНИЯ.

\Rightarrow Так как G — минимальный по включению граф смежности, найдется пара вершин $u, v \in G$, все магистральные пути между которыми проходят через ребро e , так как в противном случае его можно было бы исключить. Но тогда в графе G_e ребро e будет мостом по определению.

Алгоритм 3. Алгоритм проверки графа смежности на минимальность

Require: V, E, ω

Ensure: \mathbb{T}/\mathbb{F}

```
for all  $e \in E$  do
   $V_e = \emptyset$ 
  for all  $v \in V$  do
    if  $\omega_e \subseteq \omega_v$  then
       $V_e \leftarrow v$ 
    end if
  end for
  induction( $V_e$ )  $\rightarrow G_e$ 
  find_briges()  $\rightarrow$  briges
  if  $e \notin$  briges then
    return  $\mathbb{F}$ 
  end if
end for
return  $\mathbb{T}$ 
```

\Leftarrow Если в графе G_e ребро e является мостом, то существует пара вершин $u, v \in G_e$, все пути между которыми проходят через ребро e . Тогда в графе G все магистральные пути между u и v проходят через e , и, следовательно, ребро e нельзя исключить из графа. Так как алгоритм проверяет все ребра, никакое ребро нельзя исключить из G , то есть G — минимальный по включению граф смежности.

6.2. Оценка. Наконец, оценим асимптотику предложенного алгоритма. Делается m итераций, на каждой из них необходимо выполнить поиск мостов с препроцессингом $\text{induction}()$ и построением V_e . Множество V_e можно построить за $O(n)$, а препроцессинг выполнить за $O(m)$. Построение мостов занимает $O(n + m)$, например, алгоритмом Шмидта [23] на основе ушной декомпозиции [24] и теоремы Уитни [25]. Итоговая асимптотика — $O(n(n + m + n + m)) = O(n(n + m))$.

7. Несвязные графы. Рассмотрим теперь случай несвязных графов. Как уже упоминалось выше, мы можем рассматривать каждую компоненту связности как отдельный граф. За n и m по-прежнему будем обозначать число вершин и число ребер в графе соответственно, за n_1, \dots, n_k и m_1, \dots, m_k — число вершин и ребер в компонентах связности соответственно.

Для алгоритма проверки магистральной связности дерева (в данном случае магистральной связности леса) асимптотика получится следующей: $\sum_1^k O(n_i^2)$. В то же время, если бы граф был связным, но имел такое же число вершин и ребер, асимптотика была бы $O(n^2)$, то есть $O((\sum_1^k n_i)^2)$. Поскольку $(\sum_1^k n_i)^2 = \sum_1^k n_i^2 + t, t \geq 0$, в несвязном графе алгоритм работает даже быстрее. Абсолютно аналогичная ситуация будет и для алгоритма проверки магистральной связности произвольного графа.

Для алгоритма проверки графа смежности на минимальность асимптотика получится $\sum_1^k O(n_i(n_i + m_i))$. В то же время, если бы граф был связным, но имел такое же число вершин и ребер, асимптотика была бы $O(n(n + m))$. Если в выражении $\sum_1^k O(n_i(n_i + m_i))$ заменить первое вхождение n_i на n и вынести его за скобку, то получится $\sum_1^k O(n(n_i + m_i)) = O(n(\sum_1^k n_i + \sum_1^k m_i)) = O(n(n + m))$. Таким образом, этот алгоритм в несвязном случае также работает быстрее.

8. Заключение. Предложены алгоритмы проверки принадлежности графа семейству графов смежности в общем случае и для деревьев. Проведены теоретические оценки сложности. Предложен и доказан подход к ускорению в среднем алгоритма проверки принадлежности графа семейству графов смежности в общем случае. Наконец, показан и обоснован алгоритм проверки графа смежности на минимальность. В будущем предполагается рассмотреть вопрос построения таких графов (или, возможно, модификации сети) с помощью генетических алгоритмов [26] или других эвристик [27], где предложенные в работе решения будут полезны и применимы, а также более пристально изучить вопросы, связанные с природой графов смежности и минимальных графов смежности. Кроме того, предложенные алгоритмы могут быть полезны в реализациях программных библиотек, чтобы предотвратить ошибки пользователей при ручном построении или модификации вторичной структуры, а также при проведении вычислительных экспериментов, когда испытываются алгоритмы-кандидаты, синтезирующие графы смежности по первичной структуре сети. Одним из направлений применения алгебраических байесовских сетей в будущем являются исследования социоинженерных атак [28–30].

Авторы выражают благодарность и глубокую признательность лаборанту-исследователю Санкт-Петербургского международного математического института имени Леонарда Эйлера Татьяне Дмитриевне Мосеевой за помощь со сверткой комбинаторных формул.

Литература

1. Тулупьев А. Л., Столяров Д. М., Ментюков М. В. Представление локальной и глобальной структуры алгебраической байесовской сети в Java-приложениях. *Труды СПИИРАН*, (5), 71–99 (2007).
2. Опарин В. В., Тулупьев А. Л. Синтез графа смежности с минимальным числом ребер: формализация алгоритма и анализ его корректности. *Труды СПИИРАН*, (11), 142–157 (2009).
3. Тулупьев А. Л., Сироткин А. В., Николенко С. И. *Байесовские сети доверия: логикопроверяемый вывод в ациклических направленных графах*. Санкт-Петербург, Изд-во С.-Петерб. ун-та (2009).
4. Фильченков А. А. Субоптимальная звездчатая структура алгебраической байесовской сети. *Информационно-управляющие системы*, (2), 13–17 (2013).
5. Опарин В. В., Фильченков А. А., Сироткин А. В., Тулупьев А. Л. Матроидное представление семейства графов смежности над набором фрагментов знаний. *Научно-технический вестник информационных технологий, механики и оптики*, (4), 73–76 (2010).
6. Pearl J. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Elsevier (2014).
7. Rue H., Held L. *Gaussian Markov random fields: theory and applications*. CRC Press (2005).
8. Kharitonov N. A., Maximov A. G., Tulupjev A. L. Algebraic Bayesian Networks: Naïve Frequentist Approach to Local Machine Learning Based on Imperfect Information from Social Media and Expert Estimates. *Russian Conference on Artificial Intelligence*. Springer, Cham, 234–244 (2019).
9. Madsen A. L., Jensen F., Salmerón A., Langseth H., Nielsen T. D. A parallel algorithm for Bayesian network structure learning from large data sets. *Knowledge-Based Systems* **117**, 46–55 (2017).
10. Baltrušaitis T., Ahuja C., Morency L. P. Multimodal machine learning: A survey and taxonomy. *IEEE transactions on pattern analysis and machine intelligence* **41** (2), 423–443 (2018).
11. Sawada K., Hashimoto K., Nankaku Y., Tokuda K. A Bayesian framework for image recognition based on hidden Markov eigen-image models. *IEEJ Transactions on Electrical and Electronic Engineering* **13** (9), 1335–1347 (2018).
12. Cai B., Kong X., Liu Y., Lin J., Yuan X., Xu H., Ji R. Application of Bayesian networks in reliability evaluation. *IEEE Transactions on Industrial Informatics* **15** (4), 2146–2157 (2018).
13. Ghahramani Z. Probabilistic machine learning and artificial intelligence. *Nature* **521** (7553), 452–459 (2015).
14. Smirnov S. Towards conformal invariance of 2D lattice models. *Proceedings of the International Congress of Mathematicians (ICM)*, Madrid, Spain, August 22–30, 2006. Vol. II, 1421–1451. Zürich, European Mathematical Society (2006).

15. Schmidt J. M. The Mondschein sequence. In: *Lecture Notes in Computer Science*, 967–978 (2014).
16. Cormen T. H., Leiserson C. E., Rivest R. L., Stein C. *Introduction to algorithms*. MIT Press (2009).
17. Tarjan R. E. Efficiency of a good but not linear set union algorithm. *Journal of the ACM (JACM)* **22** (2), 215–225 (1975).
18. Tarjan R. E., Van Leeuwen J. Worst-case analysis of set union algorithms. *Journal of the ACM (JACM)* **31** (2), 245–281 (1984).
19. Fredman M., Saks M. The cell probe complexity of dynamic data structures. *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, 345–354 (1989).
20. Lancia G., Vidoni P. Finding the largest triangle in a graph in expected quadratic time. *European Journal of Operational Research* **286** (2), 458–467 (2020).
21. Spielman D. A., Shang-Hua T. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *Journal of the ACM (JACM)* **51** (3), 385–463 (2004).
22. Bollobás B. *Modern graph theory*. Vol. 184. Springer Science & Business Media (2013).
23. Schmidt J. M. A simple test on 2-vertex-and 2-edge-connectivity. *Information Processing Letters* **113** (7), 241–244 (2013).
24. Szegedy B., Szegedy C. Symplectic spaces and ear-decomposition of matroids. *Combinatorica* **26** (3), 353–377 (2006).
25. Whitney H. Non-separable and planar graphs. In: *Classic Papers in Combinatorics*. Modern Birkhäuser Classics. Boston, Birkhäuser, 25–48 (2009). https://doi.org/10.1007/978-0-8176-4842-8_2
26. Shang X., Chao T., Ma P., Yang M. An efficient local search-based genetic algorithm for constructing optimal Latin hypercube design. *Engineering Optimization* **52** (2), 271–287 (2020).
27. Ghoshal S., Sundar S. Two heuristics for the rainbow spanning forest problem. *European Journal of Operational Research* **285** (3), 853–864 (2020).
28. Корепанова А. А., Абрамов М. В., Тулупьева Т. В. Идентификация аккаунтов пользователей в социальных сетях «ВКонтакте» и «Одноклассники». *Семнадцатая Национальная конференция по искусственному интеллекту с международным участием (КИИ-2019)*. Ульяновск, 21–25 октября 2019, Ульяновск, УлГТУ. Т. 2, 153–163 (2019).
29. Shindarev N., Bagretsov G., Abramov M., Tulupyeva T., Suvorova A. Approach to identifying of employees' profiles in websites of social networks aimed to analyze social engineering vulnerabilities. *Advances in Intelligent Systems and Computing* **679**, 441–447 (2018). https://doi.org/10.1007/978-3-319-68321-8_45
30. Khlobystova A. O., Abramov M. V., Tulupyev A. L. An approach to estimating of criticality of social engineering attacks traces. In: *Studies in Systems, Decision and Control* **199**, 446–456 (2019).

Статья поступила в редакцию 20 июля 2020 г.;
 после доработки 3 сентября 2020 г.;
 рекомендована в печать 17 декабря 2020 г.

Контактная информация:

Максимов Анатолий Григорьевич — мл. науч. сотр.; agm@dscs.pro
 Тулупьев Александр Львович — д-р физ.-мат. наук, проф.; alt@dscs.pro

Algebraic Bayesian networks: Checking backbone connectivity*

A. G. Maksimov¹, A. L. Tulupyev^{1,2}

¹ St. Petersburg Federal Research Center of the Russian Academy of Sciences,
 39, 14-ia liniia V. O., St. Petersburg, 199178, Russian Federation

² St. Petersburg State University, 7–9, Universitetskaya nab., St. Petersburg, 199034, Russian Federation

For citation: Maksimov A. G., Tulupyev A. L. Algebraic Bayesian networks: Checking backbone connectivity. *Vestnik of Saint Petersburg University. Mathematics. Mechanics. Astronomy*, 2021, vol. 8 (66), issue 2, pp. 305–316. <https://doi.org/10.21638/spbu01.2021.210> (In Russian)

*The work was performed on state assignment of St. Petersburg Federal Research Center of the Russian Academy of Sciences (project no. 0073-2019-0003).

The paper investigates the construction of a joint graph as a global structure of network based on its primary structure, one of the problems arising in machine learning of bases of knowledge patterns with uncertainty, presented in the form of algebraic Bayesian networks. The aim of the research is to propose methods for solving the inverse problem. As the results, algorithms for checking a graph for belonging to a family of joint graphs and a family of minimal joint graphs are proposed, and estimates of their computational complexity are made. An improved version for the special case and an improvement for the general case on average are also proposed for the algorithm for checking membership in a family of joint graphs. The problem of recognition of joint graphs has not been previously researched; issue is being addressed for the first time as currently drafted. The theoretical significance lies in the possibilities for applying the results in further researches of graph-theoretic invariants in the global structures of algebraic Bayesian networks.

Keywords: algebraic Bayesian networks, joint graph, minimal joint graph, algorithms, complexity of algorithms.

References

1. Tulupyev A. L., Stolyarov D. M., Mentyukov M. V. Representation of the local and global structure of a Bayesian algebraic network in Java applications. *SPIIRAS Proceedings*, (5), 71–99 (2007). (In Russian)
2. Oparin V. V., Tulupyev A. L. Synthesis of a joint graph with a minimum number of edges: formalization of the algorithm and analysis of its correctness. *SPIIRAS Proceedings*, (11), 142–157 (2009). (In Russian)
3. Tulupyev A. L., Sirotkin A. V., Nikolenko S. I. *Bayesian beliefs networks: logical probabilistic inference in acyclic directed graphs*. St. Petersburg, St. Petersburg University Press (2009). (In Russian)
4. Filchenkov A. A., Tulupyev A. L., Sirotkin A. V. The power of the set of minimal joint graphs. *Information and Control Systems* **15**, 136–161 (2010). (In Russian)
5. Oparin V. V., Filchenkov A. A., Sirotkin A. V., Tulupyev A. L. Matroid representation of a family of joint graphs over a set of knowledge fragments. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, (4), 73–76 (2010). (In Russian)
6. Pearl J. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Elsevier (2014).
7. Rue H., Held L. *Gaussian Markov random fields: theory and applications*. CRC Press (2005).
8. Kharitonov N. A., Maximov A. G., Tulupyev A. L. Algebraic Bayesian Networks: Naïve Frequentist Approach to Local Machine Learning Based on Imperfect Information from Social Media and Expert Estimates. *Russian Conference on Artificial Intelligence*. Springer, Cham, 234–244 (2019).
9. Madsen A. L., Jensen F., Salmerón A., Langseth H., Nielsen T. D. A parallel algorithm for Bayesian network structure learning from large data sets. *Knowledge-Based Systems* **117**, 46–55 (2017).
10. Baltrušaitis T., Ahuja C., Morency L. P. Multimodal machine learning: A survey and taxonomy. *IEEE transactions on pattern analysis and machine intelligence* **41** (2), 423–443 (2018).
11. Sawada K., Hashimoto K., Nankaku Y., Tokuda K. A Bayesian framework for image recognition based on hidden Markov eigen-image models. *IEEE Transactions on Electrical and Electronic Engineering* **13** (9), 1335–1347 (2018).
12. Cai B., Kong X., Liu Y., Lin J., Yuan X., Xu H., Ji R. Application of Bayesian networks in reliability evaluation. *IEEE Transactions on Industrial Informatics* **15** (4), 2146–2157 (2018).
13. Ghahramani Z. Probabilistic machine learning and artificial intelligence. *Nature* **521** (7553), 452–459 (2015).
14. Smirnov S. Towards conformal invariance of 2D lattice models. *Proceedings of the International Congress of Mathematicians (ICM)*, Madrid, Spain, August 22–30, 2006. Vol. II, 1421–1451. Zürich, European Mathematical Society (2006).
15. Schmidt J. M. The Mondschein sequence. In: *Lecture Notes in Computer Science*, 967–978 (2014).
16. Cormen T. H., Leiserson C. E., Rivest R. L., Stein C. *Introduction to algorithms*. MIT Press (2009).
17. Tarjan R. E. Efficiency of a good but not linear set union algorithm. *Journal of the ACM* **22** (2), 215–225 (1975).

18. Tarjan R. E., Van Leeuwen J. Worst-case analysis of set union algorithms. *Journal of the ACM (JACM)* **31** (2), 245–281 (1984).
19. Fredman M., Saks M. The cell probe complexity of dynamic data structures. *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, 345–354 (1989).
20. Lancia G., Vidoni P. Finding the largest triangle in a graph in expected quadratic time. *European Journal of Operational Research* **286** (2), 458–467 (2020).
21. Spielman D. A., Shang-Hua T. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *Journal of the ACM (JACM)* **51** (3), 385–463 (2004).
22. Bollobás B. *Modern graph theory*. Vol. 184. Springer Science & Business Media (2013).
23. Schmidt J. M. A simple test on 2-vertex-and 2-edge-connectivity. *Information Processing Letters* **113** (7), 241–244 (2013).
24. Szegedy B., Szegedy C. Symplectic spaces and ear-decomposition of matroids. *Combinatorica* **26** (3), 353–377 (2006).
25. Whitney H. Non-separable and planar graphs. In: *Classic Papers in Combinatorics*. Modern Birkhäuser Classics. Boston, Birkhäuser, 25–48 (2009). https://doi.org/10.1007/978-0-8176-4842-8_2
26. Shang X., Chao T., Ma P., Yang M. An efficient local search-based genetic algorithm for constructing optimal Latin hypercube design. *Engineering Optimization* **52** (2), 271–287 (2020).
27. Ghoshal S., Sundar S. Two heuristics for the rainbow spanning forest problem. *European Journal of Operational Research* **285** (3), 853–864 (2020).
28. Korepanova A. A., Oliseenko V. D., Abramov M. V., Tulupyev A. L. Application of machine learning methods in the task of identifying user accounts in two social networks. *Computer tools in education*, (3), 29–43 (2019). <https://doi.org/10.32603/2071-2340-2019-3-29-43>
29. Shindarev N., Bagretsov G., Abramov M., Tulupyeva T., Suvorova A. Approach to identifying of employees' profiles in websites of social networks aimed to analyze social engineering vulnerabilities. *Advances in Intelligent Systems and Computing* **679**, 441–447 (2018). https://doi.org/10.1007/978-3-319-68321-8_45
30. Khlobystova A. O., Abramov M. V., Tulupyev A. L. An approach to estimating of criticality of social engineering attacks traces. In: *Studies in Systems, Decision and Control* **199**, 446–456 (2019).

Received: July 20, 2020
 Revised: September 3, 2020
 Accepted: December 17, 2020

Authors' information:

Anatolii G. Maksimov — agm@dscs.pro
 Aleksandr L. Tulupyev — alt@dscs.pro