УДК 004.8

А.А. Фильченков

АЛГОРИТМ ПОСТРОЕНИЯ МНОЖЕСТВА МИНИМАЛЬНЫХ ГРАФОВ СМЕЖНОСТИ ПРИ ПОМОЩИ КЛИК ВЛАДЕНИЙ

Фильченков А.А. **Алгоритм построения множества минимальных графов смежности при помощи клик владений.**
Аннотация. Известен эффективный алгоритм построения множества минимальных графов смежности по заданному набору максимальных фрагментов знаний (при помощи самоуправляемых клик), однако этот алгоритм может быть улучшен путем привлечения разработанной теории глобальной структуры алгебраической байесовской сети. Цель работы — улучшить работу этого алгоритма за счет усовершенствованного построения *владений* (компонент связности строгих сужений) — ключевых объектов в построении данного множество: строить их не прямым поиском, а путем анализа пересечений множеств вершин детей соответствующих клик. Был предложен алгоритм, реализующий предложенные улучшения, и доказана его корректность.
Ключевые слова: алгебраические байесовские сети, вторичная структура, машинное обучение, вероятностно-графические модели систем знаний, глобальная структура.

Filchenkov A.A. **Minimal join graph set synthesis possession cliques algorithm.**
Abstract. The effective minimal join graph set synthesis algorithm (of self-managing cliques) from a given maximal knowledge pattern set is known, but it can be improved by engaging the developed theory of algebraic Bayesian network global structure. The goal of the work is to improve the known minimal join graph set algorithm by perfecting designing *possessions* (connection components of strong narrowings) — the key objects for the set synthesis: to design them not with the straight search but with the corresponding cliques children's intersections analysis. A new algorithm implementing suggested improvements is designed.
Keywords: algebraic Bayesian networks, secondary structure, machine learning, probabilistic graphical models, global structure.

1. Введение. Дефицит знаний и дефицит систем знаний с неопределенностью, на наличие которых было указано в работе [24], явились предпосылкой для создания в начале 80–х годов алгебраических байесовских сетей (АБС) — логико-вероятностной модели систем знаний с неопределенностью, которая позволяет работать как с точечными, так и с интервальными вероятностями [1, 2, 15, 23].

Как их ближайшие родственники — байесовские сети доверия (БСД), алгебраические байесовские сети позволяют осуществлять пропагацию поступивших свидетельств и осуществлять согласованный синтез оценок внутри сети. Так же как и в БСД, в АБС ключевую роль играет вторичная структура, которая обычно представляется в виде графа смежности и характеризует связи между максимальными фрагментами знаний сети, представляющими ее первичную структуру [7, 15, 17]. Следует отметить, что поскольку БСД оперируют точечными оценками, а АБС — как точечными, так и интервальными, любая байесовская сеть доверия может быть представлена через алгебраическую байесовскую сеть [13, 15].

Выбор графа смежности (т.е. вторичной структуры) чрезвычайно важен для работы АБС [11–14, 16]. Лучше всего для этого подходят деревья смежности, соответствующие ациклическим вторичным структурам, которые обеспечивают наиболее эффективную работу сети [10], однако лишь для некоторых первичных структур возможно построение подобных деревьев. Именно поэтому актуальной задачей является исследование множества минимальных графов смежности, которое в случае существования деревьев смежности содержит их, а в общем случае содержит самые эффективные (с точки зрения времени и даже возможности работы АБС) вторичные структуры [4–8].

К сожалению, минимальные графы смежности до сих пор изучены недостаточно. Основной работой, в которой сформулирована система терминов, а также изучен широкий спектр свойств минимальных графов смежности, в том числе сформулирована и доказана теорема о множестве минимальных графов смежности, утверждающая, что указанное множество является декартовым произведением множеств особых подграфов минимальных графов смежности, является статья [18]. Кроме того, в ней была предложена схема базового алгоритма построения множества минимальных графов смежности, а также предпринята попытка улучшить время его работы. Ряд теоретических результатов был получен в работах [3, 20–22]. Особенно важна работа [22], в которой была предложена классификация владений — компонент связности строгих сужений, а также доказана теорема о том, что данная классификация является исчерпывающей. В статье [19] был предложен улучшенный алгоритм построения множества минимальных графов смежности (при помощи самоуправляемых клик), однако этот алгоритм может быть улучшен.

Цель данной работы — предложить улучшение существующего алгоритма построения множества минимальных графов смежности на основе учета специфики владений и реализовать данное улучшение, разработав новый алгоритм построения этого множества.

2. Основные определения и обозначения. В данном разделе приводятся определения так, как они были введены в работе [20]. Данная статья содержится в этом же выпуске, поэтому определения приведены максимально сжато, опущен формализм, иллюстрации и пояснения.

Граф — пара $\langle V, E\rangle$, где V — множество вершин графа, а E — множество ребер, каждое из которых является неупорядоченной парой $(v_i, v_j),\ i \neq j, v_i,\ v_j \in V$.

Алфавит — множество атомарных пропозициональных формул $A = \{x_1, \dots, x_n\}$, над которым будут заданы максимальные фрагменты знаний. *Слово* V — подмножество алфавита.

Множество главных конъюнктов максимальных фрагментов знаний (МФЗ), вошедших в АБС, — это такое множество слов $V^* = \{V_i \subseteq A\}_{i=1}^{i=m}$, что:

1) оно не содержит несобственное подмножество алфавита;
2) никакое слово полностью не содержит никакое другое слово.

Граф максимальных фрагментов знаний — ненаправленный граф G, вершины которого соответствуют элементам множества главных конъюнктов МФЗ, вошедших в алгебраическую байесовскую сеть, а ребра удовлетворяют условию

$$(V_i, V_j) \in E(G) \Rightarrow V_i \cap V_j \neq \emptyset.$$

Вес $W(V_i)$ *вершины* $V_i \in V(G)$ — множество атомов алфавита, вошедших в V_i. *Вес* $W(\{V_i, V_j\})$ *ребра* $\{V_i, V_j\} \in E(G)$ графа G определяется как пересечение весов тех вершин, которые соединены этим ребром. *Вес* $W(H)$ *подграфа* $H \subseteq G$ — наибольшее по включению слово, которое входит в веса всех его вершин.

Магистральный путь $B: V_b \dashrightarrow V_e$ от вершины V_b до вершины V_e, пересечение весов которых непусто, — это такой путь от вершины V_b до вершины V_e, что вес любой принадлежащей ему вершины содержит пересечение весов начальной и конечной вершин:

$B: V_b \dashrightarrow V_e = P\colon V_b \rightsquigarrow V_e$, такой, что

$$\forall V_i \in B W(V_b) \cap W(V_e) \subset W(V_i).$$

Граф *магистрально связен*, если между каждой парой несовпадающих вершин, веса которых содержат общие элементы, существует магистральный путь. Обозначим множество магистрально связных графов через $\mathbf{BCG}$.

Граф смежности — магистрально связный граф МФЗ. *Минимальный граф смежности* — граф смежности, число ребер в котором минимально. Обозначим множество минимальных графов смежности через $\mathbf{MJG}$. *Максимальный граф смежности* $G_{\max}$ — наибольший по числу ребер граф смежности.

Сужение $G \downarrow U$ ненаправленного графа G на слово U — это ненаправленный граф, в который входят только те вершины и ребра исходного графа G, веса которых содержат или равны U:

Любое слово, являющееся весом какого-либо ребра графа G, будем называть *значимым словом графа* G, а сужение графа G на такое слово — *значимым сужением*.

Клика U — значимое сужение максимального графа смежности на вес U. Любая клика является полным подграфом графа $G_{\max}$. Множество всехтаких клик будем обозначать как Clique.

Граф клик — направленный граф, вершинами которого являются клики из множества Clique. Ребро из вершины P в вершину Q проведено, если клика P содержит клику Q, и не существует клики R, такой, что клика P содержит клику R и клика R содержит клику Q.

Сильное сужение $G \downarrow\!\!\!\!\!\!\;- U$ — значимое сужение $G \downarrow U$, из которого удалили все ребра веса U:

$$G \downarrow\!\!\!\!\!\!\;- U = \langle\{V_i | V_i \in V(G), U \subseteq W(V_i)\}, \{E_i | E_i \in E(G), U \subset W(E_i)\}\rangle.$$

Сильное сужение графа $G_{\max} \downarrow\!\!\!\!\!\!\;- U$ представляет собой компоненты связности, на которые разбивается сужение $G_{\max} \downarrow U$ путем удаления ребер веса U. *Владение* P_U^i — множество вершин i-й компоненты связности сильного сужения $G_{\max} \downarrow\!\!\!\!\!\!\;- U$.

Доменная вершина D_U клики U — вершина, принадлежащая клике U и не принадлежащая ни одному из ее сыновей.

Вассал V_U клики U — множество вершин, входящих в какого-либо сына клики U.

Два вассала V_U^i и V_U^j называются *братьями*, если их пересечение непусто:

$$V_U^i \leftrightarrow V_U^j \Leftrightarrow V_U^i \cap V_U^j \neq \emptyset.$$

Два вассала V_U^i и V_U^j называются *полусиблингами*, если существует такой упорядоченный набор вассалов $\{V_U^{w_1}, \dots, V_U^{w_n}\}$, что V_U^i — брат $V_U^{w_1}$, $V_U^{w_1}$ — брат $V_U^{w_2}$, …, $V_U^{w_{n-1}}$ — брат $V_U^{w_n}$, а $V_U^{w_n}$ — брат V_U^j:

$$V_U^i \leftrightarrow V_U^j \Leftrightarrow \exists\{V_U^{w_1}, \dots, V_U^{w_n}\}: V_U^b \leftrightarrow V_U^{w_1}, V_U^{w_n} \leftrightarrow V_U^e$$
$$\text{и } \forall i < n \;\; V_U^{w_i} \leftrightarrow V_U^{w_{i+1}}.$$

Полусиблинговый путь между двумя родственными вассалами V_U^i и V_U^j — такой набор вассалов $\{V_U^{w_1}, \dots, V_U^{w_n}\}$ из определения 6’, что $V_U^b \leftrightarrow V_U^{w_1}$; $V_U^{w_n} \leftrightarrow V_U^e$ и $\forall i < n \; V_U^{w_i} \leftrightarrow V_U^{w_{i+1}}$.

Братство B_U клики U — непустой набор вассалов $\{V_U^1, \dots, V_U^l\}$ клики U, такой, что с каждым вассалом в братство входят все его полусиблинги и только они:

$$B_U = \{V_U^i | (V_U^i \in B_U) \& (V_U^i \leftrightarrow V_U^j) \Rightarrow V_U^j \in B_U; V_U^i, V_U^j \in B_U \Rightarrow$$
$$\Rightarrow V_U^i \leftrightarrow V_U^j\}.$$

Теорема о классификации владений [22]. Любое владение любого сильного сужения $G \mathrel{\not\downarrow} U$ является либо доменной вершиной, либо вассалом, либо братством U.

Сжатие σ_U компоненты связности $P_U^i \subseteq G \mathrel{\not\downarrow} U$ в вершину f_i — отображение на множестве подмножеств вершин, сопоставляющее множеству вершин P_U^i вершину f_i.

Сжатие σ_U множества ребер

$$E_{i,j} \subset (G \downarrow U) \backslash (G \mathrel{\not\downarrow} U),$$

соединяющих владения P_U^iи P_U^j в ребро $e_{i,j}$, — отображение на множестве подмножеств ребер, сопоставляющее множеству ребер $E_{i,j}$ ребро $e_{i,j}$, соединяющее вершины $f_i = \sigma_U(P_U^i)$ и $f_j = \sigma_U(P_U^j)$ и имеющее кратность, равную $|E_{i,j}|$.

Сжатие σ_U графа смежности G в граф K_U — отображение на множестве графов, сопоставляющее графу G, являющемуся графом смежности, граф K_U, вершинами которого являются владения сильного сужения $G \mathrel{\not\downarrow} U$, а ребро между двумя вершинами f_1 и f_2 графа K_U существует, если существует ребро в графе G между вершинами f_1 и f_2, принадлежащими соответствующим владениям P_U^1 и P_U^2. Кратность такого ребра (f_1, f_2) равна числу всех ребер, соединяющих вершины из P_U^1 и P_U^2.

Феод — f_i — вершина, получившаяся путем сжатия какого-то владения.

Курия веса U — K_U — ненаправленный граф с кратными ребрами, полученный путем сжатием значимого сужения $G \downarrow U$.

Оммаж H_U — курия K_U, являющаяся деревом, все ребра которой имеют кратность, равную единице. Любое сжатие минимального графа смежности является оммажем.

Жила S_U — множество ребер графа смежности G, такое, что $E_u = \{\sigma_U(e) | e \in S_U\}$ является множеством ребер оммажа сжатия σ_U. В жилу S_U входят те и только те ребра минимального графа смежности M, вес которых равен U.

Пучок — граф, построенный на исходном наборе вершин, множество ребер которого равно объединению жил, выбранных по одной для каждого значимого слова.

Теорема (о множестве минимальных графов смежности) [18, 20]. Множество минимальных графов смежности совпадает с множеством пучков.

Следствие 1 [18]. Число ребер в минимальных графах смежности одинаково.

Следствие 2 [18]. Множество минимальных графов смежности совпадает с множеством пучков, которое равно декартовому произведению множеств жил каждой клики.

Следствие 3 [18]. Мощность множества графов смежности равна произведению мощностей множеств жил каждой клики.

Следствие 4 [18]. Согласно следствию 2, для того, чтобы построить множество графов смежности, достаточно для каждой клики построить множество соответствующей ей жил.

3. Классификация клик и алгоритм построения множества минимальных графов смежности при помощи самоуправляемых клик. *Собственное ребро клики* — ребро, принадлежащее клике, вес которого совпадает с весом клики.

По числу собственных ребер множество клик можно поделить на:

- *безреберные клики* — C_0 — клики (сужения), у которых нет собственных ребер;
- *однореберные клики* — C_1 — клики, у которых ровно одно собственное ребро;
- *многореберные клики* — C_n — клики, у которых более одного собственного ребра.

По наличию детей множество клик можно поделить на:

- *бездетные* — C^- — клики, у которых нет детей;
- *родительские* — C^+ —клики, у которых есть дети.

Таблица 1. **Классификация клик**

По ребрам \ По детям	Бездетные	Родительские
Безреберные	C_0^-	C_0^+
Однореберные	C_1^-	C_1^+
Многореберные	C_n^-	C_n^+

По числу феодов, в которые сжимаются клики, они делятся на:

- *моноклики* — клики, сжимающиеся до одного феода.
- *стереоклики* — клики, сжимающиеся до более чем одного феода.

Псевдоклика — C_0^- — сужение, не имеющее собственных ребер и не имеющее детей.

Моноклика-0 — C_0^+ — сужение, не имеющее собственных ребер, но имеющее детей.

Биклика — C_1^- — клика, имеющее ровно одно собственное ребро и не имеющее детей.

Моноклика-1 — C_1^+ — клика, имеющая ровно одно собственное ребро и имеющая детей.

Бездетная поликлика (возможно также название *бездетная стереоклика*) — C_n^- —клика, имеющая более одного собственного ребра, но не имеющая детей.

Родительская поликлика — C_n^+ — клика, имеющая более одного собственного ребра и имеющая детей.

Моноклика-n — mC_n^+ — клика, имеющая более одного собственного ребра и имеющая детей, но состоящая ровно из одного феода.

Родительская стереоклика — pC_n^+ — клика, имеющая более одного собственного ребра, имеющая детей, и состоящая более чем из одного феода.

Полученные в [20] результаты удобно расположить в таблице (табл. 2).

Таблица 2. **Характеристики различных клик**

Сужение	Обозначение	Является ли кликой	Есть ли дети	Число собственных ребер	Число вершин	Число феодов	Число жил
Псевдоклика	C_0^-	Нет	Нет	0	1	0	0
Моноклика-0	C_0^+	Нет	Да	0	> 2	1	1
Биклика	C_1^-	Да	Нет	1	2	2	1
Моноклика-1	C_1^+	Да	Да	1	> 2	1	1
Бездетная стереоклика	C_n^-	Да	Нет	> 1	> 2	> 2	> 1
Родительская поликлика	C_n^+	Да	Да	> 1	> 2	$\forall$	$\forall$
Моноклика-n	mC_n^+	Да	Да	> 1	> 2	1	1
Родительская стереоклика	pC_n^+	Да	Да	> 1	> 2	> 1	> 1

Базовая схема алгоритма построения множества минимальных графов смежности [18].

Над множеством вершин V строится максимальный граф смежности $G_{\max}$. По этому графу строится граф клик Clique. Каждой клике сопоставляется множество вершин, которые в нее попали. На графе клик задается порядок.

По заданному порядку перебираются все клики из упорядоченного дерева клик. Для каждой клики при помощи алгоритма Прюфера перебираются все оммажи, поскольку оммаж является деревом — и для каждого оммажа перебираются все соответствующие ему жилы. Таким образом, для каждой клики перебираются все жилы, соответствующие ее весу. Все такие жилы записываются в массив жил для данной клики.

В работе [19] было предложено три улучшения базового алгоритма, а также сам алгоритм построения множества минимальных графов смежности при помощи самоуправляемых клик.

Улучшение 1 (исключение незначимых сужений) [19]. Вместо того чтобы рассматривать все возможные клики, мы можем исключить из рассмотрения моноклики-0, так как они не создают никаких жил, и перебор оставшихся клик будет сведен к перебору весов ребер максимального графа смежности.

Улучшение 2 (исключение клик с единственным владением) [19]. Вместо того чтобы строить жилы для моноклик-1 и моноклик-n, мы можем исключить их из множества клик, для которых строятся жилы, что ускорит вывод минимальных графов смежности, так как нужно будет перебирать меньшее число клик.

Улучшение 3 (априорный учет ребер однореберных бездетных клик) [19]. Вместо того, чтобы строить жилы для биклик, мы можем исключить их из множества клик, для которых строятся жилы, добавив их ребра в граф обязательных ребер, что ускорит вывод минимальных графов смежности, так как нужно будет перебирать меньшее число клик.

Алгоритм построения минимальных графов смежности при помощи самоуправляемых клик (self-managing cliques algorithm) строит по заданному набору вершин V, соответствующих множеству главных конъюнктов максимальных ФЗ, все возможные минимальные графы смежности.

Require: V
Ensure: $MJG = \{\langle V, E_i \rangle\}$
1: $Weights = \emptyset$
2: **for all** $u, v \in V, (u \neq v) \& (W(u) \cap W(v) \neq \emptyset)$ **do**
3: $Weighs \leftarrow W(u) \cap W(v)$
4: **end for**
5: $Cliques = \emptyset$

for all $v \in V$ **do**
  for all $w \in Weights$
    if $w \subset W(v)$ **do**
      $Cliques \rightarrow C_w : W(C_w) = w$
      $C_w \longleftarrow u$
      $Cliques \longleftarrow C_w$
    end if
  end for
end for
$NecessaryEdges = \emptyset$
$StereoIndex = \emptyset$
for all $C_w \in \mathrm{Cliques}$ **do**
  if $|C_w| = 2$ **then**
    $C_w \rightarrow v$
    $C_w \rightarrow u : u \neq v$
    $NecessaryEdges \longleftarrow (v, u)$
  end if
  else
    $P =$ компоненты связности строго сужения клики C_w
    if $|P| > 1$ **then**
      $StereoIndex \longleftarrow w$
      $S_w = \emptyset$
      for all t **is_a** дерево на P **do**
        for all s **is_a** жила для t **do**
          $S_w \longleftarrow s$
        end for
      end for
    end if
  end else
end for
$MJG = \emptyset$
for all $\{s_w^i\} : \{s_w^i\}$ —индексирующая последовательность жил стереоклик $w \in StereoIndex, s_w^i \in S_w$ **do**
  $E = NecessaryEdges$
  for all $s \in \{s_w^i\}$ **do**
    $E = E \cup s.edges$

```
    end for
    MJG ⟵ ⟨V, E⟩
end for
return MJG
```

Листинг 1. Алгоритм построения минимального графа смежности при помощи самоуправляемых клик

Здесь и далее $S \leftarrow e$ обозначает добавление элемента e к множеству S, а $S \rightarrow e: Cond(e)$ обозначает извлечение элемента, удовлетворяющего условиям $Cond$, из множества S в переменную e.

4. Алгоритм построения множества минимальных графов смежности при помощи самоуправляемых клик.

Улучшение 4 (построение владений через сыновей). Вместо того, чтобы строить владения клик путем поиска компонент связности соответствующего строго сужения, мы можем строить владения как компоненты связности относительно родства вассалов (см. теорему о классификации владений), то есть компоненты связности будут либо вершинами, принадлежащими клике, но не принадлежащими ее сыновьям (доменными вершинами); либо вершинами, принадлежащими одному из сыновей клики и не принадлежащими больше никому (вассалами); либо максимальными по включению наборами вассалов, такие, что любые два вассала в одном наборе являются полусиблингами (братствами). Для этого нужно будет построить граф клик.

Пояснение. Вообще говоря, полезность улучшения 4 не столь очевидна, как у улучшений 1–3. Тогда как алгоритм ПМГС при помощи самоуправляемых клик в качестве результата возвращает только множество минимальных графов смежности и совершенно не раскрывает внутреннего устройства клик, улучшение 4 требует построение дополнительных сведений о кликах, таких, как множество ее сыновей и разделение ее вершин на доменные, принадлежащие вассалам и входящие в братства. Все это требует времени работы программы. Однако конечной целью построения множества минимальных графов смежности является в том числе и изучение свойств сети, поэтому получение и вывод этой информации будут полезны.

Если не учитывать время на получение дополнительной информации, построение компонент связности на множестве вершин происходит дольше, чем построение компонент связности в смысле пересечения на множестве подмножеств тех же вершин. Именно ради этого и было предложено улучшение 4. Однако, поскольку оно не является абсолютным, алгоритм ПМГС при помощи самоуправляемых клик остается вполне функциональным.

Алгоритм построения минимальных графов смежности при помощи клик владений (possessing cliques algorithm) строит по заданному набору вершин V, соответствующих множеству главных конъюнктов максимальных ФЗ, все возможные минимальные деревья смежности графа МФЗ.

```
Require: V
Ensure: MJG = {⟨V, E_i⟩}
Weights = ∅
for all u,v ∈ V,(u ≠ v)&(W(u)⋂W(v) ≠ ∅) do
  Weighs ← W(u)⋂W(v)
end for
Cliques = ∅
for all v ∈ V do
  for all w ∈ Weights
    if w ⊂ W(v) do
      Cliques[w] ⟵ u
    end if
  end for
end for
ICl = множество индексов Cliques
NecessaryEdges = ∅
StereoIndex = ∅
for w = последний по порядку вес из ICl
to первый по порядку вес из ICl do
  C = Cliques[w]
  Domains[w] = Cliques[w]
  if |Domains[w]| > 2 then
    C.Sons = ∅
    for w* = последний по порядку вес из ICl
    to идущий по порядку после w вес
    из ICl do
      C* = Cliques[w*]
      if W(C) ⊂ W(C*) then
        C.Sons = C.Sons\C*.Sons
        C.Sons ← C*
        Domains[w] = Domains[w]\Domains[w*]
```

```
      end if
      w = предыдущий по порядку вес из ICl
    end for
    IntersEdges = ∅
    for all S_1, S_2 ∈ C.Sons, (S_1 ≠ S_2)&(S_1 ∩ S_2) ≠ ∅
    do
      IntersEdges ← (S_1, S_2)
    end for
    P = компоненты связности
    ⟨C.Sons, IntersEdges⟩
    P ⟵ Domains[w]
    if |P| > 1 then
      StereoIndex ← w
      Sinews[w] = ∅
      for all t is_a дерево на P do
        for all s is_a жила для t do
          Sinews[w] ← s
        end for
      end for
    end if
  end if
  else
    Domains[w] → v
    Domains[w] → u: u ≠ v
    NecessaryEdges ← (v, u)
  end else
  w = предыдущий по порядку вес из ICl
end for
MJG = ∅
for all {s_w^i}: {s_w^i} — индексирующая
последовательность жил стереоклик
w ∈ StereoIndex, s_w^i ∈ S_w do
  E = NecessaryEdges
  for all s ∈ {s_w^i} do
    E = E ∪ s.edges
  end for
  MJG ← ⟨V, E⟩
end for
```

61: **return** MJG

Листинг 2. Алгоритм построения минимального графа смежности при помощи клик владений

В цикле (2–4) строится множество $Weights$ значимых весов.

В цикле (6–12) строится множество вершин для каждой клики путем перебора всех вершин.

В цикле (7–11) перебираются все веса из множеств $Weights$ и рассматриваемая вершина добавляется в каждую клику, которая должна ее содержать.

Условный оператор (8–10) в случае, если вес вершины содержит вес клики, добавляет эту вершину к этой клике.

В строке 13 инициализируется множество индексов коллекции $Clique$, которое представляет собой множество значимых слов (весов).

В цикле (16–52) последовательно (в обратном лексикографическом порядке, заданном на весах) перебираются все клики, исключаются из рассмотрения моноклики, особым образом обрабатываются биклики, строится множество обязательных ребер и для всех стереоклик строится множество их жил.

Условный оператор (19–45) в случае, если число вершин, принадлежащих клики, больше двух, строит множество сыновей и множество собственных вершин для данной клики, а также все ее домены, оммажи и в конечном итоге жилы.

В цикле (21–29) перебираются все клики, имеющий вес, следующий за весом рассматриваемой клики и строится множество сыновей данной клики, а также множество ее собственных вершин.

Условный оператор (23–27) в случае, если вес рассматриваемой клики C содержится в весе перебираемых в цикле (21–29) клик C^*, вычитает из множества сыновей клики C множество сыновей клики C^*, добавляет C^* в множество сыновей клики C и вычитает из множества собственных вершин клики C собственные вершины клики C^*.

В цикле (31–33) строится множество ребер на сыновьях рассматриваемой клики: ребро между двумя сыновьями существует тогда, когда множество их вершин пересекается.

Условный оператор (36–44) в случае, если число владений у клики больше одного, строит множество всех возможных жил для данной клики.

В цикле (39–43) перебираются все деревья, построенные на феодах клики (т.е. оммажи). Для этого, как правило, используется алгоритм Прюфера.

В цикле (40-–42) в массив жил для данной клики добавляются все жилы, соответствующие данному оммажу.

Условный оператор (46–50) вызывается в случае, когда число вершин в клике равно двум. Он добавляет две эти вершины к множеству обязательных ребер $NecessaryEdges$.

В цикле (54–60) алгоритм перебирает все кортежи, индексирующие жилы для каждой клики. Каждый кортеж индексирует набор жил, выбранных по одной для каждой клики. В цикле происходит объединение всех жил одного кортежа с ребрами из множества обязательных ребер. Граф, содержащий эти ребра, добавляется к множеству минимальных графов смежности.

Утверждение 1. Алгоритм построения множества минимальных графов смежности при помощи клик владений строит множество минимальных графов смежности.

Доказательство. В цикле (2–12) строится массив клик (т.е. вершин, этим кликам принадлежащих) по определению клик.

В цикле (21–29) будут построены все сыновья и доменные вершины рассматриваемой клики, потому что вес сыновей клики идет после веса самой клики в лексико-графическом порядке, ее доменные вершины — это ее вершины за исключением тех, которые входят в сыновей (строка 26), а сыновья определяются как клики, чей вес содержит вес рассматриваемой клики (условие в строке 23) и они не являются ее прямыми потомками (т.е. в первом колене строка 24 исключает из множества сыновей всех «внуков», то есть сыновей предполагаемых сыновей).

Результатом выполнения строк 34 и 35 станет множество всех владений рассматриваемой клики, это следует из теоремы о классификации владений.

В цикле (36–44) будет построено множество всех жил для данной клики, если у нее более двух владений (т.е. она является стереокликой), что следует из определения жилы и оммажа. Моноклики игнорируются согласно улучшениям 1–2.

Таким образом в условном операторе (19–45) будут построены только стереоклики.

В условном операторе (46–50) будут обработаны все биклики.

В цикле (16–52) будут рассмотрены все клики, для стереоклик будет построено множество жил, собственные ребра биклик будут добавлены в граф обязательных ребер, а моноклики будут проигнорированы.

Цикл (54–60) перебирает все наборы жил по одной для каждой клики стереоклики и объединяет их с множеством обязательных ребер. Согласно теореме о минимальных графах смежности, мы перебираем таким образом все возможные множества ребер минимальных графов смежности.

Оценка времени работы алгоритма остается открытой задачей, поскольку оно чрезвычайно сильно зависит от структуры АБС, для которой строится множество минимальных графов смежности. Однако можно утверждать, что если не учитывать время на построение графа клик (который интересен не только с точки зрения построения рассматриваемого множества, но также и с точки зрения целого ряда проблем глобальной структуры АБС), то алгоритм построения множества минимальных графов смежности при помощи клик владений работает быстрее алгоритма построения множества минимальных графов смежности при помощи самоуправляемых клик, поскольку поиск компонент связности путем построения пересечений работает очевидно быстрее, чем непосредственный поиск самих компонентов связности.

Тем не менее, если учитывать время, затрачиваемое на построение графа клик, то преимущества алгоритма клик владений становятся спорными, поэтому оба алгоритма имеют право на существование, как представляется на данный момент.

5. Заключение. В статье были рассмотрены результаты анализа клик и вторичных структур, проведенные в работах [20–22]. На основании классификации владений, предложенной в статье [20], было сформулировано улучшение для алгоритма синтеза множества минимальных графов смежности, представляющих вторичную структуру алгебраической байесовской сети, предложенного в работе [18] и улучшенного в работе [19].

Улучшение состоит в оптимизации построения компонент связности строго сужения (*владений*) — сужения (клики), у которого удалили все ребра его веса. Как было доказано в статье [20], любой такой компонент связности, называемый владением, является либо отдельной вершиной, вес которой совпадает с весом клики (*доменная вершина*), либо множеством вершин, принадлежащих сыну клики *(вассал)*, либо максимальным по включению набором таких вассалов, который является компонентом связности в смысле пересечения множеств вершин двух сыновей: два сына соединены (являются *братьями*), если множество их вершин пересекается, два сына соединены (являются *полусиблингами*), если существует путь между ними.

Именно это рассуждение и легло в основу улучшения: владения можно строить не путем поиска компонент связности строго сужения, а путем поиска компонент связности в описанном выше смысле на множестве сыновей клики.

Это предполагает построения графа клик, что является достаточно трудоемкой операцией, однако если этот граф уже построен (сейчас он представляется важным и для других вопросов, связанных и исследованием и построением вторичной структуры), то алгоритм будет работать более эффективно.

Предложен алгоритм построения множества минимальных графов смежности, реализующий указанное улучшение, а также три улучшения, предложенные в [19], который будет выполняться быстрее, чем базовый алгоритм, предложенный в статье [18]. Доказана корректность алгоритма.

Дальнейший анализ глобальной структуры алгебраических байесовских сетей позволит обособить и структурировать новые элементы, специальная работа с которыми ляжет в основу дальнейших улучшений алгоритма построения множества минимальных графов смежности.

Литература

1. *Городецкий В.И.* Алгебраические байесовские сети — новая парадигма экспертно-вычислительных систем // Юбилейный сборник трудов институтов Отделения информатики, вычислительной техники и автоматизации РАН. Т. 2. М.: РАН, 1993, С. 120–141.
2. *Городецкий В.И., Тулупьев А.Л.* Формирование непротиворечивых баз знаний с неопределенностью // Известия РАН. Сер. Теория и системы управления. 1997. №5. С. 33–42.
3. *Опарин В.В., Фильченков А.А., Тулупьев А.Л., Сироткин А.В.* Матроидное представление семейства графов смежности над набором фрагментов знаний // Научно-технический вестник Санкт-Петербургского государственного университета информационных технологий, механики и оптики. 2010. Вып. 4. С. 73–76.
4. *Павельчук А.В., Тулупьев А.Л., Тотмянина С.А.* Подход к объектно-ориентированному представлению данных алгебраических байесовских сетей в java-коде и реляционных СУБД // Региональная информатика-2008 (РИ-2008). XI Санкт-Петербургская международная конференция. Санкт-Петербург, 22–24 октября, 2008 г.: Материалы конференции / СПОИСУ. СПб.: 2009. С. 68–76.
5. *Сироткин А.В.* Модели, алгоритмы и вычислительная сложность синтеза согласованных оценок истинности в алгебраических байесовских сетях // Информационно-измерительные и управляющие системы. 2009. №11. С. 32–37.
6. *Тотмянина С.А., Павельчук А.В., Тулупьев А.Л.* Алгебраические байесовские сети: структуры данных в СУБД и Java- коде // Интегрированные модели, мягкие вычисления, вероятностные системы и комплексы программ в искусственном интеллекте. Научно-практическая конференция студентов, аспирантов, молодых ученых

и специалистов (Коломна, 26–27 мая 2009 г.). Научные доклады. В 2-х т. Т. 2. М.: Физматлит, 2009, С. 123–131.
7. *Тулупьев А.Л.* Алгебраические байесовские сети: глобальный логико-вероятностный вывод в деревьях смежности: Учеб. пособие. СПб.: СПбГУ; ООО Издательство «Анатолия», 2007. 40 с. (Сер. Элементы мягких вычислений).
8. *Тулупьев А.Л.* Алгебраические байесовские сети: реализация логико-вероятностного вывода в комплексе java-программ // Труды СПИИРАН. СПб.: Наука, 2009. Вып. 8. С. 191–232.
9. *Тулупьев А.Л.* Алгебраические байесовские сети: система операций глобального логико-вероятностного вывода // Информационно-измерительные и управляющие системы. 2010. № 11. С 65–72.
10. *Тулупьев А.Л.* Ациклические алгебраические байесовские сети: логико-вероятностный вывод // Нечеткие системы и мягкие вычисления: Научный журнал Российской ассоциации нечетких систем и мягких вычислений. 2006. Том 1, № 1. С. 57–93.
11. *Тулупьев А.Л.* Байесовские сети: логико-вероятностный вывод в циклах. СПб.: Изд-во С.-Петербургского ун-та, 2008. 140 с. (Элементы мягких вычислений.)
12. *Тулупьев А.Л.* Непротиворечивость оценок вероятностей в алгебраических байесовских сетях. Вестник СПбГУ. Сер. 10. 2009. Вып. 3. С. 144–151.
13. *Тулупьев А.Л.* Преобразование ациклических байесовских сетей доверия в алгебраические байесовские сети // Известия высших учебных заведений: Приборостроение. 2009. № 3. С. 21–23.
14. *Тулупьев А.Л.* Согласованность данных и оценка вероятности альтернатив в цикле стохастических предпочтений // Известия высших учебных заведений: Приборостроение. 2009. № 7. С. 3–8.
15. *Тулупьев А.Л., Николенко С.И., Сироткин А.В.* Байесовские сети: логико-вероятностный подход. СПб.: Наука, 2006. 607 с.
16. *Тулупьев А.Л., Сироткин А.В.* Алгебраические байесовские сети: принцип декомпозиции и логико-вероятностный вывод в условиях неопределенности // Информационно-измерительные и управляющие системы. 2008. № 10. т. 6. С. 85–87.
17. *Тулупьев А.Л., Сироткин А.В., Николенко С.И.* Байесовские сети доверия: логико-вероятностный вывод в ациклических направленных графах. СПб.: Изд-во С.-Петерб. ун-та, 2009, 400 с.
18. *Фильченков А.А., Тулупьев А.Л.* Структурный анализ систем минимальных графов смежности Труды СПИИРАН. 2009. Вып. 11. С. 104–127.
19. *Фильченков А.А.* Алгоритм построения множества минимальных графов смежности при помощи самоуправляемых клик // Труды СПИИРАН. 2010. Вып. 1 (12). [в печати]
20. *Фильченков А.А., Тулупьев А.Л., Сироткин А.В.* Компаративный анализ клик минимальных графов смежности алгебраических байесовских сетей // Труды СПИИРАН. 2010. Вып. 2 (13). [в печати]
21. *Фильченков А.А., Тулупьев А.Л., Сироткин А.В.* Особенности анализа вторичной структуры алгебраической байесовской сети // Труды СПИИРАН. 2010. Вып. 1 (12). [в печати]
22. *Фильченков А.А., Тулупьев А.Л., Сироткин А.В.* Структурный анализ клик минимальных графов смежности // Вестник Тверского государственного университета. Сер. Прикладная математика. 2010. [в печати].
23. *Gorodetsky V.I., Drozdgin V.V., Jusupov R.M.* Application of Attributed Grammar and Algorithmic Sensitivity Model for Knowledge Representation and Estimation // Artifi-

cial Intelligence and Information, Control Systems of ROBOTSA. Amsterdam: Elsevier Science Publishers B. V., 1984, P. 232–237.

24. *Korb K.B., Nicholson A.E.* Bayesian Artificial Intelligence. NY.: Chapman and Hall/CRC. 2004. 364 p.

Фильченков Андрей Александрович — аспирант кафедры информатики математико-механического факультета С.-Петербургского государственного университета (СПбГУ), младший научный сотрудник лаборатории теоретических и междисциплинарных проблем информатики СПИИРАН. Область научных интересов: автоматическое обучение вероятностных графических моделей. Число научных публикаций — 6. aaafil@mail.ru, СПИИРАН, 14-я линия В.О., д. 39, г. Санкт-Петербург, 199178, РФ; р.т. +7(812)328-3337, факс +7(812)328-4450. Научный руководитель — А.Л. Тулупьев.

Filchenkov Andrey Alexandrovich — PhD student of Computer Science Department, SPbGU, junior researcher, Theoretical and Interdisciplinary Computer Science Laboratory, SPIIRAS Research area: machine learning of probabilistic graphical models. The number of publications — 6. aaafil@mail.ru, SPIIRAS, 14-th line V.O., 39, St. Petersburg, 199178, Russia; office phone +7(812)328-3337, fax +7(812)328-4450. Scientific advisor — A.L. Tulupyev.

Поддержка исследования. Работа выполнена при финансовой поддержке РФФИ, проект № **09-01-00861**-а «Методология построения интеллектуальных систем поддержки принятия решений на основе баз фрагментов знаний с вероятностной неопределенностью».

Рекомендовано лабораторией ТиМПИ, заведующий лабораторией д-р физ.-мат. наук, доцент А.Л. Тулупьев.
Статья поступила в редакцию 10.12.2010.

РЕФЕРАТ

Фильченков А.А. **Алгоритм построения множества минимальных графов смежности при помощи клик владений.**

Цель работы — предложить улучшения для существующего алгоритма построения множества минимальных графов смежности и реализовать их в новом алгоритме.

Рассмотрена теория глобальной структуры алгебраических байесовских сетей, в частности, классификация владений клик и классификация клик графов смежности. Приведена базовая схема алгоритма построения множества минимальных графов смежности, три ее улучшения и известный улучшенный алгоритм построения множества минимальных графов смежности (при помощи самоуправляемых клик).

Над множеством вершин строится максимальный граф смежности. По этому графу строится граф клик. Каждой клике сопоставляется множество вершин, которые в нее попали. На графе клик задается порядок. По заданному порядку перебираются все клики из упорядоченного дерева клик. Для каждой клики при помощи алгоритма Прюфера перебираются все оммажи, поскольку оммаж является деревом — и для каждого оммажа перебираются все соответствующие ему жилы. Таким образом, для каждой клики перебираются все жилы, соответствующие ее весу. Все такие жилы записываются в массив жил для данной клики. Согласно теореме о множестве минимальных графов смежности, нам достаточно построить все возможные пучки, чтобы получить все минимальные графы. Это можно сделать, перебрав всевозможные комбинации жил для каждой клики, объединяя такие жилы в единый граф.

Первое улучшение состояло в том, чтобы рассматривать при построении графа клик только клики, то есть только те подграфы максимального графа смежности, вес которых совпадает с весом какого-либо из ребер этого графа.

Второе улучшение состояло в том, чтобы рассматривать при переборе клик из графа клик только те клики, которые имеют более одной компоненты связности при удалении из них ребер веса, равного весу клики (т.е. имеющих ровно одно владение).

Третье улучшение состояло в том, чтобы клики, состоящие ровно из двух вершин, не использовать для построения особых множеств ребер — жил, а выделять из них единственное ребро, которое добавлять к специальному графу — графу обязательных ребер.

На основании понятия владения предложено улучшение, которое строит владения не путем поиска компонент связанности на строгом сужении, а путем поиска компонент связанности на множестве сыновей клики, где под связью между двумя сыновьями понимается наличие общих вершин. Полученные компоненты связности, объединенные с доменными вершинами будут являться владениями клики.

Предложен алгоритм построения множества минимальных графов смежности, реализующий описанные улучшения, доказана его корректность.

Дальнейший анализ глобальной структуры алгебраической байесовской сети может быть полезен в целях создания новых эффективных алгоритмов построения семейства минимальных графов смежности.

SUMMARY

Filchenkov A.A. **Minimal join graph set synthesis possession cliques algorithm.**

The goal of the work is to suggest improvement for the known minimal join graph set synthesis algorithm and actualize them in a new algorithm.

Algebraic Bayesian network global structure in particular theory, clique possessions classification and join graph cliques classification are overviewed. Minimal join graph set synthesis basic algorithm scheme is represented with its three known improvements and the known minimal join graph set synthesis (self-managing cliques) algorithm.

Maximal join graph should be built over a given vertex set. A clique graph should be built by that graph. Every clique should be collated to the set of vertexes that belong to the clique. Cliques order should be specified. With the order all the cliques should be sorted out. All homages should be sorted out for every clique by the means of the Pruefer algorithm, because an homage is a tree. Than all sinews should be sorted out for every homage. So that, all the sinews would be sorted out for each clique. All such sinews should be reordered to sinew array for the clique. According to the minimal join graph set theorem, it's enough to design all branches in the purpose to design all minimal join graphs. It can be done by sorting out all sinew combinations chosen one for each clique and uniting them into the one graph.

On the base of observed classification three basic algorithm improvements that base some cliques specifies, are suggested.

The first improvement is to use only the cliques (e.g. only the maximal join graph subgraphs), the weight of which equals to weigh of any maximal join graph edge.

The second improvement is to use only the cliques that have more than one connection components having lost all the edges of the same weight. (e.g. the cliques that has more than one possession).

The third improvement is to use cliques that have only two vertices not in designing sinews, but to add their only edge to a special graph — necessary edges graph.

New algorithm improvement designing possessions not by searching connection components on strong narrowing but by searching connection components on clique son set (where connection between two sons is understood as containing at least the one same vertex) based on a concept of possession is suggested. Designed components being united with domains are clique possessions.

Minimal join graph set synthesis algorithm using the improvement is designed and proven to be correct.

Further algebraic Bayesian network global structure analysis could be useful in the purpose to design new effective minimal join graph set synthesis algorithms.