

УДК 519.179

DOI: 10.15827/2311-6749.19.179

## ДЕКОМПОЗИЦИОННЫЙ ПОДХОД ПРИ РЕШЕНИИ ОПТИМИЗАЦИОННЫХ ЗАДАЧ НА БОЛЬШИХ РАЗРЕЖЕННЫХ ГРАФАХ

*В.В. Быкова, д.ф.-м.н., профессор, bykvalen@mail.ru; Р.Е. Илларионов, магистрант, illroman@yandex.ru; Ю.И. Кириллов, аспирант, hellsingyura@mail.ru  
(Сибирский федеральный университет, просп. Свободный, 79, г. Красноярск, 660041, Россия)*

**Аннотация.** Рассматривается декомпозиционный подход, приводящий к ускорению работы классических алгоритмов решения оптимизационных задач на разреженных графах большой размерности. Подход основан на разложении исходного графа на атомы кликовыми минимальными сепараторами. Разреженность графа выражается ограничением на древовидную ширину графа. Представлены алгоритмы и программы, реализующие атомарное разложение графа. Демонстрируется практическое применение данного подхода к NP-трудной задаче о клике.

**Ключевые слова:** двухфазные алгоритмы, предобработка, атомарное разложение, древовидная ширина, оптимизация на графах.

Отличительными особенностями графов, возникающих в современных приложениях, являются их большая размерность и разреженность. Ускорение работы любого из известных алгоритмов решения оптимизационных задач на графах может быть достигнуто за счет предобработки исходного графа. Технология предобработки реализуется, как правило, в виде двухфазной процедуры. На первой фазе производится предобработка графа путем его редуцирования или декомпозиции. Выбор вида предобработки определяется особенностями решаемой задачи и структурой графа. На второй фазе обычно применяется известный алгоритм решения задачи к преобразованному графу и формируется решение для исходного графа. При этом требуется, чтобы предобработка существенно снижала размерность задачи, обеспечивала корректность формирования решения задачи исходя из одного или нескольких решений, полученных для преобразованного графа, вычислительная сложность предобработки не превышала сложности решения рассматриваемой задачи на исходном графе.

Реализация декомпозиционного подхода, то есть процесса выделения частей исходного графа, во многом зависит от структурных особенностей этого графа, в том числе от ограничений на параметры, характеризующие его разреженность. Один из параметров разреженности графа – ограничение на его древовидную ширину [1, 2]. Для графов с ограниченной древовидной шириной используется разложение графа кликовыми минимальными сепараторами. Идея такого разложения была предложена Р. Тарьяном [3] как средство реализации принципа «разделяй и властвуй» для решения NP-трудных задач на графах, базирующихся на отношениях смежности вершин графа. Полученные в результате разложения части графа были названы атомами. Тарьяном установлено, что разложение графа на атомы не разрушает клики этого графа, не порождает новые клики и в этом смысле сохраняет его структуру.

В данной работе предлагаются алгоритмы и программы, практически реализующие идею Тарьяна. Демонстрируется применение данного подхода к NP-трудной задаче о клике.

### Основные понятия и обозначения

Введем основные понятия и обозначения из теории графов, необходимые для изложения результатов работы. Используем терминологию и обозначения, принятые в работе [4].

Пусть  $G = (V, E)$  – неориентированный конечный граф с множеством вершин  $V$  и множеством ребер  $E$ . Считается, что вершины  $v_1$  и  $v_2$  графа  $G$  смежные, если  $\{v_1, v_2\} \in E$ , и несмежные в противном случае. Ребро  $e$  инцидентно вершине  $v_1$ , если  $e = \{v_1, v_2\} \in E$ , и не инцидентно иначе. Если  $e = \{v_1, v_2\} \in E$ , то вершины  $v_1$  и  $v_2$  называют концами ребра  $e$ . В этом случае говорят также, что ребро  $e$  соединяет вершины  $v_1$  и  $v_2$ . Два ребра считаются кратными, если их концы совпадают. Ребро, соединяющее вершину саму с собой, называется петлей. Простой граф – неориентированный конечный граф  $G = (V, E)$  без петель и кратных ребер. Везде далее будут рассматриваться только простые графы, для которых  $|V| \geq 2$  и  $|E| \geq 1$ .

Множество всех вершин графа  $G$ , смежных с некоторой вершиной  $v \in V$ , образует в  $G$  окрестность (или открытую окрестность) вершины  $v$ , которая обозначается через  $N(v)$ . Замкнутая окрестность вершины  $v$  обозначается  $N[v]$  и определяется равенством:  $N[v] = N(v) \cup \{v\}$ . Число  $\deg v = |N(v)|$  называется степенью вершины  $v$ .

Граф  $G' = (V', E')$  называется подграфом графа  $G = (V, E)$  при условии, что  $V' \subseteq V$ ,  $E' \subseteq E$ . Подграф  $G' = (V', E')$  является остовным, когда  $V' = V$ . Если множество вершин подграфа  $G'$  есть  $V'$ , а множество его ребер совпадает с множеством всех ребер графа  $G$ , оба конца которых принадлежат  $V'$ , то  $G'$  называется подграфом, индуцированным множеством  $V' \subseteq V$ , и обозначается  $G(V')$ . Всякий максимальный по включению связный подграф графа  $G$  образует его компоненту связности. Множество вершин некоторой компоненты графа  $G$  называется областью связности этого графа.

Множество вершин  $V$  называется кликой графа  $G$ , если в графе  $G(V)$  всякие две вершины смежные, и максимальной кликой, если она не содержится в клике с большим количеством вершин. Размер наибольшей (по числу вершин) клики графа  $G$  обозначается  $\phi(G)$  и называется плотностью графа  $G$ . Граф  $G = (V, E)$ , являющийся кликой, называется полным графом и обозначается через  $K_n$ , где  $n = |V|$ . Для него  $\phi(K_n) = |V|$ , а в общем случае для  $G = (V, E)$  всегда  $1 \leq \phi(G) \leq |V|$ .

Говорят, что множество вершин  $S \subset V$  разделяет несмежные вершины  $u$  и  $v$  графа  $G = (V, E)$ , если в графе  $G - S = G(V \setminus S)$  вершины  $u$  и  $v$  принадлежат различным компонентам связности. Множество  $S$  при этом называется  $(u, v)$ -сепаратором и минимальным  $(u, v)$ -сепаратором, если  $S$  есть  $(u, v)$ -сепаратор и в нем нет собственного подмножества, являющегося  $(u, v)$ -сепаратором. Сепаратор  $S$  считают минимальным, если в графе  $G$  существует такая пара вершин  $u$  и  $v$ , что  $S$  – минимальный  $(u, v)$ -сепаратор. Наименьший по мощности сепаратор графа  $G$  – наименьший сепаратор. Одновершинный сепаратор называется точкой сочленения графа. Кликовый минимальный сепаратор – это минимальный сепаратор, являющийся кликой.

Граф  $G$  называется хордальным, если ни один из его индуцированных подграфов не является простым циклом длины  $l \geq 4$ . Любой граф можно превратить в хордальный, добавив в него некоторое множество ребер. Триангуляцией графа  $G = (V, E)$  называется хордальный граф  $H = (V, E')$ ,  $E \subseteq E'$ , который содержит  $G$  в качестве остовного подграфа. Триангуляция  $H$  – минимальная триангуляция графа  $G$ , если для  $G$  не существует другой триангуляции, которая является собственным подграфом графа  $H$ . Наименьшей считается триангуляция с наименьшим числом ребер.

Древовидная ширина (treewidth) – числовой параметр, характеризующий меру древовидности графа, то есть близости его к дереву. Древовидная ширина вычисляется через дерево декомпозиции. Под деревом декомпозиции графа  $G = (V, E)$  понимают пару  $(V, T)$ , задающую определенное разбиение вершин и ребер графа  $G$ , где  $V = \{V_i : i \in I\}$  – семейство непустых подмножеств множества  $V$ , называемых «мешками»,  $T = (I, W)$  – дерево, узлам которого сопоставлены эти «мешки». Данное разбиение удовлетворяет следующим свойствам [1, 2]:

- 1) множество «мешков» покрывает все вершины графа;
- 2) для всякого ребра всегда существует хотя бы один «мешок», содержащий обе вершины этого ребра;
- 3) для любой вершины  $v \in V$  множество узлов, «мешки» которых содержат  $v$ , индуцирует поддерево дерева  $T$ .

Ширина дерева декомпозиции  $(M, T)$  – наибольшая вместимость его «мешка», уменьшенная на единицу:  $\max\{|X_i| - 1 : i \in I\}$ . Древовидная ширина графа  $G$  определяется как наименьшая ширина всех допустимых его деревьев декомпозиции и обозначается  $tw(G)$ .

### Характеризация разреженных и хордальных графов

Известно несколько формальных определений разреженного графа. Связный граф  $G = (V, E)$ ,  $n = |V| \geq 2$ ,  $|E| \geq 1$ , называют (реберно) разреженным, если число его ребер удовлетворяет условию

$$|E| \leq an^b, \quad (1)$$

где  $a > 0$ ,  $1 \leq b < 2$  – положительные вещественные константы. Считают, чем меньше значение  $b$ , тем более разреженным является граф  $G$ . Для сравнения: в каждом дереве число ребер равно  $n - 1$ , то есть  $b = 1$ , что отвечает нижней границе значения  $b$ , а для любого полного  $n$ -вершинного графа всегда  $|E| = n(n - 1) / 2$ , то есть  $b = 2$ , что соответствует верхней границе значения  $b$ .

Существует другое определение разреженного графа  $G$ , которое выражается через древовидную ширину  $tw(G)$  этого графа. Для  $tw(G)$  верны границы:  $1 \leq tw(G) \leq n - 1$ . Так, всякое  $n$ -вершинное дерево ( $n \geq 2$ ) имеет единичную древовидную ширину, что отвечает нижней границе для  $tw(G)$ , а полному  $n$ -вершинному графу свойственна древовидная ширина, равная  $n - 1$ , что соответствует верхней границе для  $tw(G)$ . Пусть  $k$  есть некоторая заданная положительная целая константа. Если  $tw(G) \leq k$ , то говорят, что граф  $G = (V, E)$  обладает ограниченной (значением  $k$ ) древовидной шириной. Считается, чем меньше значение  $k$ , тем более разреженным является граф  $G$ . Известно [2], если  $tw(G) \leq k$ , то для числа ребер графа  $G = (V, E)$  справедливо неравенство

$$|E| \leq kn - k(k + 1) / 2. \quad (2)$$

При  $k = 1$  и  $k = n - 1$  соотношение (2) приводит к неравенствам (1), соответствующим деревьям и полным графам. Следовательно, ограничение  $tw(G) \leq k$  не противоречит (1) и задает естественную меру разреженности графа  $G$ . Учитывая, что всегда  $\varphi(G) - 1 \leq tw(G)$ , можно говорить о том, что значение  $tw(G)$  ограничивает не только число ребер графа, но и размеры его клик. В дальнейшем разреженность графа  $G$  понимается в смысле ограничения  $tw(G) \leq k$ .

Процесс разложения графа на атомы базируется на свойствах хордальных графов [3]. Сформулируем наиболее важные из них в виде лемм 1–4.

**Лемма 1.** Граф  $G = (V, E)$  является хордальным тогда и только тогда, когда его любой минимальный сепаратор образует в  $G$ -клик.

**Лемма 2.** Связный граф  $G = (V, E)$  хордальный тогда и только тогда, когда для него существует дерево клик, определяемое так: множество узлов дерева – множество  $\{C_i : i \in J\}$  всех максимальных клик графа  $G$ ; два узла  $C_i, C_j$  соединены ребром, если соответствующие им клики графа  $G$  имеют непустое пересечение, то есть  $C_i \cap C_j \neq \emptyset$ ; для любой вершины  $v \in V$  графа  $G$  множество максимальных клик, содержащих эту вершину, индуцирует связный подграф, являющийся поддеревом дерева клик.

Поскольку всякий связный хордальный граф  $G = (V, E)$  имеет не более  $n - 1$  максимальных клик, то, согласно лемме 2, его дерево клик содержит не более  $n - 1$  узлов, где  $n = |V|$ .

**Лемма 3.** Связный хордальный граф  $G = (V, E)$  может иметь несколько различных деревьев клик, но каждое из них – остовное дерево наибольшего веса графа пересечений всех максимальных клик исходного графа, где вес ребра – число вершин, образующих пересечение надлежащих максимальных клик графа  $G$ .

**Лемма 4.** Пусть  $T$  – некоторое дерево клик связного хордального графа  $G$ . Для произвольного ребра дерева клик  $T$ , связывающего узлы  $C_i$  и  $C_j$ , пересечение  $C_i \cap C_j$  образует минимальный сепаратор графа  $G$ . Верно и обратное утверждение: если  $S$  – минимальный сепаратор графа  $G$ , то в  $T$  всегда найдется ребро, соединяющее узлы  $C_i$  и  $C_j$ , такое, что  $C_i \cap C_j = S$ .

Согласно леммам 2 и 4, в хордальном графе  $G = (V, E)$  минимальных сепараторов не более  $n - 2$ . По лемме 1 все эти сепараторы кликовые.

### Формулировка задачи разложения графа на атомы

Задачу разложения графа на атомы (Decomposition of Graph into Atoms Problem, DGAP) можно сформулировать следующим образом:

#### Decomposition of Graph into Atoms Problem.

**Заданы:** связный граф  $G = (V, E)$ , для которого  $|V| \geq 2$ ,  $|E| \geq 1$ ,  $tw(G) \leq k$ , то есть древовидная ширина ограничена некоторой малой по значению константой  $k > 0$ , а также множество всех кликовых минимальных сепараторов  $\Delta(G)$  этого графа.

**Требуется:** найти для  $G$  множество атомов  $\Omega(G) = \{G_1, G_2, \dots, G_p\}$ , где каждый атом  $G_i$ ,  $i = 1, \dots, p$ , определяется как максимальный связный подграф графа  $G$ , не имеющий кликовых минимальных сепараторов графа  $G$ .

Множество  $\Omega(G)$  принято называть атомарным представлением (или разложением на атомы) графа  $G$ .

Доказано [3, 5], что задача DGAP полиномиально разрешима и имеет единственное решение. Последнее означает, что множество атомов  $\Omega(G)$  уникально для  $G$ , если разложение осуществлять на основе  $\Delta(G)$  – множества всех кликовых минимальных сепараторов графа  $G$ . Если в  $\Delta(G)$  имеются минимальные сепараторы, не являющиеся кликами в  $G$ , то  $\Omega(G)$  может зависеть как от состава  $\Delta(G)$ , так и от порядка просмотра сепараторов в процессе разложения. Следует заметить, что число минимальных сепараторов графа, как правило, больше числа его кликовых минимальных сепараторов, и только для хордовых графов имеет место равенство. В общем случае граф  $G = (V, E)$  может иметь экспоненциальное число минимальных сепараторов относительно  $n = |V|$ , тогда как в хордовом графе их всегда не более  $n - 2$ . Заметим, что кликовые минимальные сепараторы могут пересекаться как множества вершин. Возможна вложенность кликовых минимальных сепараторов один в другой, поскольку каждый из них может быть минимальным относительно различных пар вершин графа. Примечательно, что не каждый граф обладает сепараторами, в том числе и кликовыми минимальными сепараторами: например, сепараторов нет в полном графе.

Следует отметить, что атомарное представление  $\Omega(G)$  графа  $G = (V, E)$  является обобщением разложения его на блоки: когда множество кликовых минимальных сепараторов  $\Delta(G)$  состоит лишь из точек сочленения графа  $G$ , то каждый атом из  $\Omega(G)$  представляет собой блок этого графа. Напомним, что блок графа – максимальный относительно включения связный подграф графа  $G$ , не имеющий точек сочленения.

### Способы вычисления кликовых минимальных сепараторов

Для определения уникального набора атомов  $\Omega(G)$  заданного графа  $G = (V, E)$  надо предварительно найти множество  $\Delta(G)$  всех кликовых минимальных сепараторов графа  $G$ . Такой набор сепараторов легко вычисляется для хордального графа с использованием лемм 1–4. Однако в общем случае существуют два способа нахождения  $\Delta(G)$ :

- найти все минимальные сепараторы, а затем среди них выделить те, которые образуют клики в  $G$ ;
- извлечь кликовые минимальные сепараторы из минимальной триангуляции  $H$  графа  $G$ .

Рассмотрим вначале первый способ. Известно [6], что множество  $S \subseteq V$  является минимальным сепаратором графа  $G = (V, E)$ , если и только если существуют две различные области связности  $D_1$  и  $D_2$  графа  $G - S$ , такие, что

$$N(D_1) = N(D_2) = S, \quad (3)$$

$$N(D_i) = (\cup_{v \in D_i} N(v)) \setminus D_i, \quad i = 1, 2.$$

Напомним, что граф  $G - S$  получается из  $G$  удалением всех вершин из  $S \subseteq V$ . Необходимое и достаточное условие (3) позволяет находить в графе  $G$  все минимальные сепараторы через близкие сепараторы. Минимальный  $(u, v)$ -сепаратор  $S$ , для которого  $S \subseteq N(v)$ , называется минимальным сепаратором, близким к вершине  $v$ . Для произвольной вершины  $v \in V$  графа  $G = (V, E)$  можно вычислить близкие к ней минимальные сепараторы следующим образом:

- определить множество вершин  $N[v]$ ;
- построить граф  $G - N[v]$  и выявить в нем все области связности;
- установить для всякой выявленной области связности  $D$  графа  $G - N[v]$  множество вершин  $N(D)$ , образующих окрестность для  $D$  в графе  $G$ . Каждое такое множество определяет в  $G$  минимальный сепаратор, близкий к вершине  $v$ .

Исходя из определения минимальных сепараторов, можно доказать следующие свойства отношения их близости к вершинам графа. Если вершина  $v \in V$  смежна со всеми другими вершинами графа  $G = (V, E)$ , то в  $G$  не существует минимального сепаратора, близкого к  $v$ . В этом случае  $N[v] = V$  и граф  $G - N[v]$  пустой. Во всех других случаях могут существовать несколько минимальных сепараторов, близких к  $v$ . Возможны случаи, когда один и тот же минимальный сепаратор близок к различным несмежным вершинам графа. При этом если  $u$  и  $v$  не являются смежными вершинами, то существует ровно один минимальный  $(u, v)$ -сепаратор, близкий к  $v$ . Кроме того, для всякого минимального сепаратора графа  $G = (V, E)$  всегда найдется хотя бы одна вершина  $v \in V$ , к которой он близок. Эти свойства минимальных сепараторов указывают стратегию их перебора: начиная с некоторого множества минимальных сепараторов, близких к вершинам, входящим в ранее обнаруженные минимальные сепараторы. Подробно алгоритм генерации всех минимальных сепараторов графа представлен в работе [6]. Ясно: если граф имеет экспоненциальное число сепараторов, то данный способ вычисления  $\Delta(G)$  может быть чрезмерно затратным по времени для графов большой размерности.

Второй способ – извлечение всех кликовых минимальных сепараторов из минимальной триангуляции  $H$  графа  $G$  – единственный на сегодняшний день эффективный (полиномиальный по времени) способ построения  $\Delta(G)$  для графа  $G$ . Заметим, что задача нахождения наименьшей триангуляции является NP-трудной [7]. Между тем для произвольного графа  $G$  всегда можно построить за полиномиальное время некоторую минимальную (не обязательно наименьшую) триангуляцию  $H$ , при этом  $tw(G) \leq tw(H)$  [1, 2]. Знание хотя бы одной минимальной триангуляции  $H$  для исходного графа  $G$  позволяет найти для  $G$  набор всех его кликовых минимальных сепараторов. Об этом свидетельствует следующее утверждение.

**Утверждение** [3, 5]. Пусть заданы граф  $G = (V, E)$  и некоторая его минимальная триангуляция  $H = (V, E')$ ,  $E \subseteq E'$ . Все минимальные сепараторы графа  $H$  являются минимальными сепараторами графа  $G$ .

Из данного утверждения следует, что кликовые минимальные сепараторы графа  $G$  можно определять путем анализа минимальных сепараторов хордального графа  $H$  и выделения среди них тех, которые образуют клики в  $G$ .

### Алгоритм разложения графа на атомы и свойства атомов

Алгоритм разложения графа на атомы (далее алгоритм MakeAtoms), включая вычисление множества  $\Delta(G)$  всех кликовых минимальных сепараторов графа  $G$ , сводится к выполнению следующей последовательности шагов.

*Шаг 1.* Нахождение для исходного графа  $G = (V, E)$  минимальной триангуляции  $H = (V, E')$ ,  $E \subseteq E'$ .

*Шаг 2.* Поиск всех максимальных кликов в триангуляции  $H$ .

*Шаг 3.* Построение дерева кликов для триангуляции  $H$  (с использованием лемм 2 и 3).

*Шаг 4.* Установление множества минимальных сепараторов  $\Delta(H)$  для триангуляции  $H$  (по леммам 1 и 4). Вычисление для графа  $G$  множества кликовых минимальных сепараторов  $\Delta(G)$  путем выделения из  $\Delta(H)$  тех сепараторов, которые образуют клики в  $G$  (согласно утверждению).

*Шаг 5.* Непосредственное формирование множества атомов  $\Omega(G)$  на основе  $\Delta(G)$ .

Шаги 1 и 2 алгоритма MakeAtoms реализуются с помощью известного алгоритма MCS-M, который находит для  $G$  минимальную триангуляцию  $H$  и все максимальные клики в  $H$ , затрачивая на это время  $O(n^3)$  [3]. На шаге 3 осуществляется построение дерева клик для связного хордального графа  $H$ , исходя из графа пересечения всех его максимальных клик, с применением классического алгоритма построения оптимального остова за время  $O(n^2)$  [4]. Шаги 3, 4 основаны на свойствах хордальных графов и утверждении. Согласно свойствам хордальных графов, минимальные сепараторы для  $H$  находятся через точки сочленения дерева клик и их число не превышает  $n - 2$ . Кроме того, каждый минимальный сепаратор минимальной триангуляции  $H$  является минимальным сепаратором для входного графа  $G$ . Значит,  $\Delta(G) \subseteq \Delta(H)$ . Поскольку граф  $H$  хордальный, любой сепаратор из  $\Delta(H)$  образует клику в  $H$ , но необязательно клику в  $G$ . Между тем всякий кликовый минимальный сепаратор графа  $G$  всегда входит в  $\Delta(H)$  независимо от выбранной на шаге 1 минимальной триангуляции  $H$ . В целом для нахождения  $\Delta(G)$  требуется время  $O(n^3)$ . На шаге 5 выполняется непосредственное формирование множества  $\Omega(G)$ . Процесс выполнения этого шага сводится к многократному разделению графа  $G$  на части одним из кликовых минимальных сепараторов  $S \in \Delta(G)$ , выделению компонент связности графа  $G - S$  и копированию  $S$  в эти компоненты. Этот процесс продолжается до тех пор, пока в полученных частях не окажется кликовых минимальных сепараторов из  $\Delta(G)$ . Время выполнения шага 5 составляет  $O(n^3)$ . Значит, исполнение алгоритма MakeAtoms требует  $O(n^3)$  времени.

Для приложений существенны следующие особенности атомов, формируемых алгоритмом MakeAtoms:

- всякий атом из  $\Omega(G)$  является порожденным подграфом графа  $G$ ;
- количество атомов в  $\Omega(G)$  не превосходит числа вершин графа  $G$ ;
- если  $G$  является хордальным графом, то каждый атом из  $\Omega(G)$  образует клику в  $G$ ;
- множество  $\Omega(G)$  сохраняет все клики графа  $G$ , то есть всякая клика графа  $G$  становится кликой одного из его атомов. Верно и обратное: все клики атомов являются кликами графа  $G$ , новых клик при разложении не возникает;
- атомы перекрываются, если они содержат один и тот же кликовый минимальный сепаратор, но не всякое непустое пересечение областей связности двух различных атомов  $\Omega(G)$  является сепаратором;
- на основе множеств  $\Omega(G)$  и  $\Delta(G)$  возможно полное восстановление графа  $G$  путем «склеивания» атомов с помощью соответствующих кликовых минимальных сепараторов.

Пару множеств  $\Omega(G)$  и  $\Delta(G)$  можно рассматривать как форму представления графа большой размерности. Такое представление может быть предварительно создано за полиномиальное время и храниться во внешней памяти компьютера, а необходимые для обработки атомы последовательно загружаться в оперативную память. Основной недостаток атомарного представления графа: не все графы имеют кликовые минимальные сепараторы. Между тем, если граф  $G$  отличен от полного графа, то в нем всегда имеются минимальные сепараторы. Если при разложении воспользоваться ими, уникальность  $\Omega(G)$  не гарантируется, однако при  $tw(G) \leq k$  размеры получаемых частей графа составят  $O(k)$ . Будут сохранены также все клики входного графа.

### Применение декомпозиционного подхода

Изложенный выше декомпозиционный подход применим для решения задач на графах, базирующихся на отношениях смежности вершин графа. Он может служить основой разработки двухфазных алгоритмов, на первой фазе которых производится предобработка входного графа путем его разложения на атомы. Как показали эксперименты, такая предобработка дает ощутимое ускорение классических алгоритмов решения NP-трудных и полиномиально разрешимых оптимизационных задач на больших разреженных графах.

Двухфазные алгоритмы были реализованы в виде программ для NP-трудной задачи о клике (Maximum-Clique-Problem, MCP), которая формулируется следующим образом [7]:

#### Maximum-Clique-Problem.

**Заданы:** граф  $G = (V, E)$  и положительное целое число  $l \leq n = |V|$ .

**Вопрос.** Верно ли, что  $G$  содержит клику размера не менее  $l$ ?

Для точного решения MCP использовался алгоритм Уилфа [8]. Данный алгоритм является рекурсивным и находит точное решение MCP за время  $O(\text{poly}(n) \cdot 1,39^n)$ , где  $\text{poly}(n)$  – некоторый полином от  $n$ . При двухфазной процедуре решения MCP первая фаза (алгоритм MakeAtoms – разложение графа на атомы) выполняется за время  $O(n^3)$ , а вторая фаза (алгоритм Уилфа для каждого атома и связывание реше-

ний) – за время  $O(n \cdot \text{poly}(k) \cdot 1,39^k)$ . Таким образом, время работы двухфазного алгоритма линейно зависит от  $n = |V|$  и экспоненциально от значения  $k$ ,  $tw(G) \leq k$ . Таким образом, чем более разреженным является граф  $G$ , тем больший эффект достигается от его предобработки при нахождении точного решения МСР с помощью алгоритма Уилфа.

Для решения задачи о клике также существует большое количество приближенных алгоритмов решения, некоторые из которых построены на жадной стратегии. Один из таких алгоритмов приведен рисунке 1. Время работы алгоритма GREEDY составляет  $O(n^2)$ . Применение данного алгоритма при двухфазном подходе к решению задачи МСР для разреженных графов дает оценку времени работы  $O(n \cdot k^2)$ .

#### Алгоритм GREEDY

**Вход:** граф  $G = (V, E)$ .

**Выход:** множество вершин, составляющих максимальную клику графа  $G$ .

**Begin**

Найти в  $G$  вершину  $v$  наименьшей степени:

$\text{deg}(v) = \min \{ \text{deg}(u) : u \in V \}$

**While**  $\text{deg}(v) \neq |V| - 1$  **do**

Удалить вершину  $v$  из графа  $G$

Положить  $G = G - v$  и пересчитать в  $G$  значение  $|V|$   
и степени всех вершин

**End**

Рис. 1. Жадный алгоритм для МСР

Вычислительные эксперименты, проведенные на случайных графах, подтвердили указанные выше теоретические оценки времени работы двухфазных алгоритмов и показали, что они работают значительно быстрее, чем соответствующие однофазные алгоритмы, и порядок ускорения существенно зависит от размерности и разреженности входного графа (рис. 2).

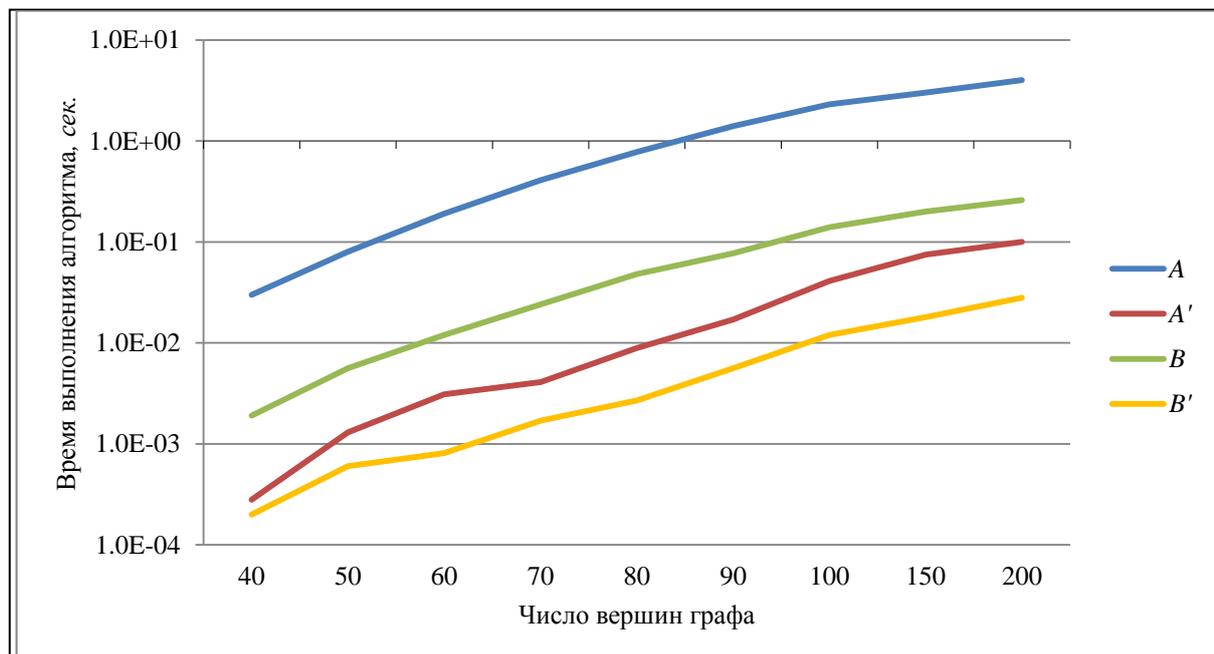


Рис. 2. Экспериментальное сравнение алгоритмов по времени работы:

A – алгоритм Уилфа, A' – алгоритм Уилфа с предобработкой,

B – алгоритм GREEDY, B' – алгоритм GREEDY с предобработкой

На сегодняшний день, когда объемы исследуемых данных достигают огромных размеров, востребованы алгоритмы и программы, способные обрабатывать эти данные за реальное время. В статье рассмотрен декомпозиционный подход, приводящий к ускорению работы классических алгоритмов решения оптимизационных задач на разреженных графах большой размерности. Подход основан на разложении исходного графа на атомы кликовыми минимальными сепараторами. Атомарное представление графа может быть предварительно создано за полиномиальное время и храниться во внешней памяти компью-

тера, а необходимые для обработки атомы последовательно загружаются в оперативную память. В статье представлены алгоритмы и программы, реализующие построение такого представления. Перспективны дальнейшие исследования по совершенствованию рассмотренного декомпозиционного подхода, направленные на учет возможных изменений в течение времени состава вершин и ребер исходного графа, и по его применению к задачам маршрутизации в коммуникационных сетях.

### *Литература*

1. Быкова В.В. Вычислительные аспекты древовидной ширины графа // Прикладная дискретная математика. 2011. № 3. С. 65–79.
2. Bodlaender L. Discovering treewidth. Proc. of Conf. SOFSEM, vol. 3381 of LNCS. Springer, 2005, pp. 1–16.
3. Tarjan R.E. Decomposition by clique separators. J. Discrete Math. 1985, no. 55, pp. 221–232.
4. Емеличев В.А., Мельников О.И., Сарванов В.И., Тышкевич Р.И. Лекции по теории графов. М.: Либроком, 2012. 392 с.
5. Leimer H.G. Optimal decomposition by clique separators. J. Discrete Mathematics. 1993, no. 113, pp. 99–123.
6. Быкова В.В., Кириллов Ю.И. Программные средства анализа отказоустойчивости технических систем на основе вершинной целостности графа // Программные продукты и системы. 2015. № 3. С. 161–165.
7. Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи. М: Мир, 1982. 416 с.
8. Wilf H.S. Algorithms and complexity. Taylor & Francis, 2002, 219 p.