DOI 10.37882/2223–2966.2025.04-2.08

ОЦЕНКА СЛОЖНОСТИ АЛГОРИТМОВ, РЕАЛИЗУЮЩИХ ОПЕРАЦИИ НАД МЕТАГРАФАМИ

Горячкин Борис Сергеевич
кандидат технических наук, доцент,
Московский государственный технический университет им. Н.Э. Баумана
bsgor@mail.ru
Гапанюк Юрий Евгеньевич
кандидат технических наук, доцент,
Московский государственный технический университет им. Н.Э. Баумана
gapyu@bmstu.ru
Винников Степан Сергеевич
Московский государственный технический университет им. Н.Э. Баумана
acesolo@mail.ru

Аннотация. Постановка проблемы. На сегодняшний день не описаны и не реализованы алгоритмы для проведения операций над метаграфами, что ограничивает возможности для обработки таких структур.
Цель. Дать возможность работать с метаграфами на математическом и алгоритмическом уровне.
Результаты. Рассмотрено альтернативное описание метаграфовой модели, указаны её существенные отличия от классической. Описаны математические операции над метаграфами. Приведены алгоритмы, реализующие данные операции. Дана теоретическая оценка сложности таких алгоритмов. Написана программа, реализующая данные алгоритмы, проверена её работоспособность. Проведены эксперименты с расчётами над метаграфами разных размеров. Проведено сравнение теоретических и экспериментальных данных, подтверждена теоретическая оценка.
Практическая значимость. Переход от классического описания метаграфов к новому позволил удобно и лаконично описывать операции над метаграфами, а также упростил работу с данной структурой. Описание операций над метаграфами и реализация алгоритмов, позволяющих проводить такие операции, расширило возможности для использования метаграфовой модели данных.

Ключевые слова: метаграфы, метаграфовая модель данных, алгоритмы, асимптотическая оценка сложности, дискретная математика.

EVALUATION OF THE COMPLEXITY OF ALGORITHMS IMPLEMENTING OPERATIONS ON METAGRAPHS

B. Goryachkin
Yu. Gapanyuk
S. Vinnikov

Summary. *Problem statement*. Currently, algorithms for performing operations on metagraphs have not been described or implemented, which limits the possibilities for processing such structures.
Goal. Provide an opportunity to work with metagraphs on a mathematical and algorithmic level.
Results. An alternative description of the metagraphic model has been considered, and its significant differences from the classical model have been indicated. Mathematical operations on metagraphs have been described. Algorithms that implement these operations have been given. A theoretical assessment of the complexity of such algorithms has been given. A program that implements these algorithms has been written, and its operability has been tested. Experiments with calculations on metagraphs of different sizes have been carried out. A comparison of theoretical and experimental data has been carried out, and the theoretical assessment has been confirmed.
Practical significance. The transition from the classical description of metagraphs to the new one made it possible to conveniently and concisely describe operations on metagraphs, and also simplified work with this structure. The description of operations on metagraphs and the implementation of algorithms that allow such operations have expanded the possibilities for using the metagraphic data model.

Keywords: metagraphs, metagraph data model, algorithms, asymptotic complexity estimation, discrete mathematics.

Введение

В настоящее время все более востребованными становятся системы интенсивной обработки графов. Интенсивность обработки графов выражается как в большом объеме обрабатываемой информации (большие графы), так и в необходимости её потоковой обработки.

В то же время для задач обработки и обогащения графов все чаще используется не модель плоского графа, а модели, основанная на сложных графах или сложных сетях. В [1] отмечается, что термин «сложная сеть» обычно используется для обозначения реальной исследуемой системы, в то время как термин «сложный граф» обычно используется для обозначения математической модели такой системы. Сложными называются графы, в которых используются сложные описания вершин, ребер и/или их расположения. Часто в таких моделях используется не плоский, а пространственный вариант расположения вершин и ребер. Именно такой подход может быть наиболее полезен при описании сложных моделей данных и знаний. На сегодняшний день наиболее широко известны три такие модели: гиперграфы [2], гиперсети [3] и метаграфы.

В данной статье мы будем использовать модель метаграфа, которая будет рассмотрена далее.

Формальное определение метаграфа

Согласно [4][5], метаграфы, и их внутренние структуры описываются следующим образом:

Метаграф — это кортеж $MG = (V, MV, E)$, где MG — метаграф; V — множество вершин метаграфа; MV — множество метавершин метаграфа; E — множество рёбер метаграфа.

Ребро метаграфа — это кортеж $e_i = (id, (id_{begin}, id_{end}))$, где e_i — ребро, принадлежащее множеству E; id — идентификатор ребра; id_{begin} — идентификатор вершины (метавершины), из которой ребро исходит; id_{begin} — идентификатор вершины (метавершины), в которую ребро входит.

Фрагмент метаграфа — это множество $MG_f = \{ev_j\}, ev_j \in (V \cup MV \cup E)$, где MG_f — фрагмент метаграфа; ev_i — элемент фрагмента метаграфа. Элемент фрагмента метаграфа — это объект, принадлежащий результату объединения множеств вершин, метавершин и рёбер метаграфа.

Метавершина — это кортеж $mv_i = (id, MG_f), mv_i \in MV$, где mv_i — метавершина, принадлежащая множеству метавершин MV метаграфа; id — идентификатор метавершины; MG_f — вложенный в метавершину фрагмент метаграфа.

По сути, метаграф — это граф, вершины которого находятся в некоторой произвольной иерархической связи. Произвольность такой связи заключается в том, что вершина или мета-вершина может быть вложена в несколько метавершин, в отличие, например, от древовидных структур.

Кроме того, можно заметить, что ребра метаграфа могут соединять любые метавершины и вершины, независимо от их положения в иерархии метаграфа. Таким образом, метаграф представляет собой обобщенный иерархический граф.

Альтернативное определение метаграфа

Классическое определение метаграфов несколько затрудняет работу с данной структурой. В частности, трудности возникают при описании операций над метаграфами, их хранении, обработке и эмбеддинге.

Это происходит по следующим причинам:
1. Метавершины содержат слишком много информации. Такая информация сама по себе иерархична, глубина вложенности метавершин может быть бесконечно большой.
2. Иерархическая структура метаграфа неизвестна заранее. Чтобы определить ее, необходимо просмотреть все метавершины, выполнить поиск в глубину и в какой-либо форме сохранить информацию об иерархии метаграфа.

Данные проблемы побуждают нас перейти к другому, улучшенному описанию структуры метаграфа. Для такого описания предлагаются следующие действия:
1. Удаление свойства иерархической вложенности из метавершин. Метавершины не должны содержать информацию о том, что в них вложено. Таким образом, метавершины преобразуются в обычные вершины.
2. Поместить описание иерархии метаграфа в отдельное множество рёбер, которое будет описывать вложенность метаграфа.

Применяя указанные действия, метаграф может быть описан следующим образом:

Метаграф — это кортеж $MG = (V, E_1, E_2)$, где MG — метаграф; V — множество вершин метаграфа; E_1 — множество рёбер метаграфа; E_2 — множество дуг, характеризующее иерархию метаграфа.

Опишем метаграф, представленный на рисунке 1, с помощью данного выше определения. Данный метаграф — это кортеж $mg_1 = (V, E_1, E_2)$, где

$$V = \{v_1, v_2, v_3, v_4, v_5, mv_1, mv_2, mv_3\}$$

$$E_1 = \{e_1, (v_1, v_2), e_2, (v_3, v_2), e_3, (v_1, v_3), e_4, (v_4, v_2), e_5, (v_3, v_5), e_6, (v_4, v_5), e_7, (mv_1, mv_2), e_8, (v_2, mv_2)\}$$

$$E_2 = \{mv_1, v_1, mv_1, v_2, mv_1, v_3, mv_2, v_4, mv_2, v_5, mv_3, v_2, mv_3, v_3, mv_3, mv_2\}$$

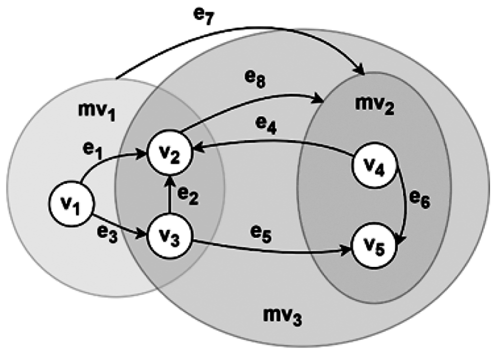

Рис. 1. Пример метаграфа

Опишем метаграф, представленный на рисунке 1, с помощью данного выше определения. Для наглядности изобразим метаграф на рисунке 1 в виде двух графов — светло-серого и серого. Эти графы представлены на рисунке 2.

Множество вершин каждого из этих графов является множеством вершин и метавершин исходного метаграфа. Множество ребер первого (светло-серого) графа описывает реберные связи вершин (метавершин) метаграфа. Если ребро соединяет две вершины в красном графе, то эти вершины (метавершины) соединяются ребром в метаграфе, при этом направление ребра сохраняется.

Набор ребер второго (серого) графа описывает вложенность вершин (метавершин) метаграфа друг в друга. Если ребро соединяет две вершины в сером графе, то вершина (метавершина) метаграфа, которая находится в конце этого ребра, расположена внутри вершины (метавершины), из которой исходит это ребро.

Отметим, что приведенное выше описание метаграфов не ограничивается двумя наборами ребер. Мы можем описать метаграф с произвольным количеством наборов ребер и даже предоставить метаграф с набором гиперрёбер и прочих математических структур.

В таком случае, метаграф может быть описан следующим образом:

Метаграф — это кортеж $MG = \left(V, \{E_i\}\right)$, где MG — метаграф; V — множество вершин метаграфа; E_i — i-е множество рёбер метаграфа.

В сущности, каждое из таких множеств рёбер представляет из себя некоторую “надстройку” над исходным множеством элементов V. Для каждого из таких множеств мы можем описать собственные свойства, ограничения и операции.

Операции над метаграфами

Определим базовые операции над метаграфами: добавление вершины; удаление вершины; добавление ребра; удаление ребра; вложение одной вершины в другую; вытягивание одной вершины из другой; объединение двух вершин.

Альтернативное определение метаграфа позволяет достаточно просто описать более сложные операции над метаграфами:

Объединение метаграфов mg_1 и mg_2 — это метаграф mg_3, заданный следующим образом:

$$\mathrm{mg}_3 = \mathrm{mg}_1 \cup \mathrm{mg}_2 = \mathrm{X}_1 \cup \mathrm{X}_2, \mathrm{E}_{1,\mathrm{mg}_1} \cup_1 \mathrm{E}_{1,\mathrm{mg}_2}, \mathrm{E}_{2,\mathrm{mg}_1} \cup_2 \mathrm{E}_{2,\mathrm{mg}_2}, \ldots,$$

для $\forall \mathrm{mg}_1, \mathrm{mg}_2 \in \mathrm{MG}$, где $\mathrm{E}_{\mathrm{i},\mathrm{mg}_1}, \mathrm{E}_{\mathrm{i},\mathrm{mg}_2}$ — это i-е множества рёбер первого и второго метаграфов; $\cup_\mathrm{i}$ — это операция объединения, определенная на i-м множестве рёбер.

Пересечение метаграфов mg_1 и mg_2 — это метаграф mg_3, заданный следующим образом:

$$mg_3 = mg_1 \cap mg_2 = X_1 \cap X_2, E_{1,mg_1} \cap_1 E_{1,mg_2}, E_{2,mg_1} \cap_2 E_{2,mg_2}, \ldots,$$

для $\forall mg_1, mg_2 \in MG$, где E_{i,mg_1}, E_{i,mg_2} — это i-е множества рёбер первого и второго метаграфов; $\cap_i$ — это операция пересечения, определенная на i-м множестве рёбер.

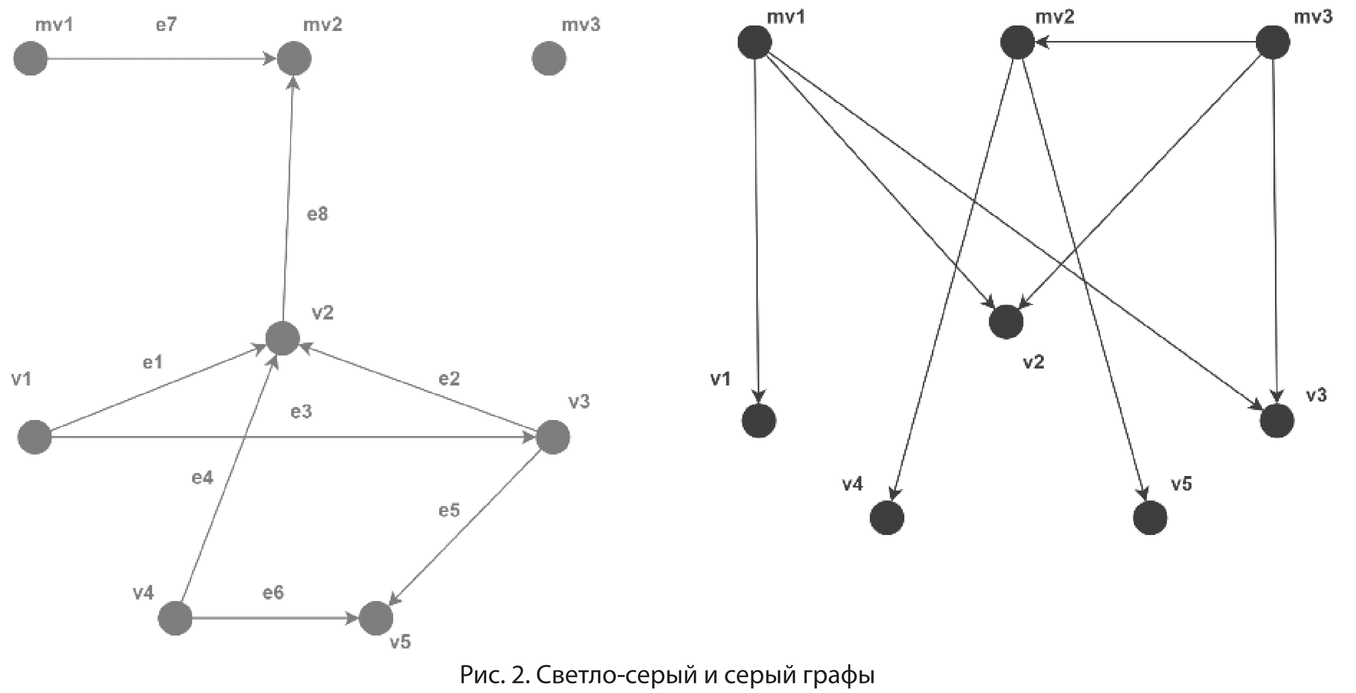

Рис. 2. Светло-серый и серый графы

Другие операции над метаграфами, включая разность, симметричную разность и т.д., определяются аналогичным образом.

Важно отметить, что при использовании операции вложения одной вершины в другую, может возникнуть конфликтная ситуация. Это происходит, если есть вершина v_2, которая вложена в вершину v_1, и мы пытаемся вложить v_1 в вершину v_2, что невозможно. Такая ситуация может возникнуть при объединении метаграфов, поэтому в алгоритме объединения метаграфов, описанном далее мы будем игнорировать применение применения операции вложения при конфликтной ситуации.

Алгоритмы объединения и пересечения

Опишем алгоритмы объединения и пересечения метаграфов. Алгоритмы представлены на рисунках 3 и 4.

В таблице 1 перечислим сложные операции для каждого из алгоритмов и укажем их асимптотическую временную вычислительную сложность от размера входных данных. На основе этих данных мы сможем дать оценку асимптотической временной сложности каждого из алгоритмов. Асимптотическая временная сложность обозначается как $O(n)$, где n — объём входных данных.

В качестве объема входных данных мы берем число вершин, получившихся в результате объединения или пересечения V_1 и V_2 то есть, величину $|V|$. При этом, важно отметить, что при подсчете асимптотической временной сложности выполняются следующие условия:

1. Не учитываются никакие константы, кроме тех, которые есть в показателе степени $|V|$.
2. Если в формуле асимптотической сложности есть некоторая сумма, мы учитываем только самое быстрорастущее слагаемое.

И того, учитывая, что самым быстрорастущим слагаемым формулы асимптотической сложности обоих алгоритмов является $|V|^3$, мы оцениваем вычислительную сложность обоих алгоритмов как $O(|V|^3)$.

Эксперименты

Приведем пример работы алгоритма и проверим его корректность. Пусть есть метаграфы, представленные на рисунках 5 и 6. Теоретические результаты пересече-

Алгоритм 1 Алгоритм объединения метаграфов

Вход метаграфы $mv_1 = \{V_1, E_{1,1}, E_{1,2}\}$ и $mv_2 = \{V_2, E_{2,1}, E_{2,2}\}$
Выход метаграф $mv = \{V, E_1, E_2\}$

$V = V_1 \cup V_2$
$E_1 = E_{1,1} \cup E_{2,1}$
$E^*_{1,2}$ – это транзитивное замыкание $E_{1,2}$ на V
$E^*_{2,2}$ – это транзитивное замыкание $E_{2,2}$ на V
Создать матрицы смежности M^1 и M^2 размера $|V|$ на $|V|$, характеризующие $E^*_{1,2}$ и $E^*_{2,2}$, i-ые строки M^1 и M^2 соответствует одной вершине $v_i \in V$, j-ые колонки соответствуют $v_j \in V$.
Создать матрицу $M = [m_{i,j} = m^1_{i,j} \vee m^2_{i,j}], 1 \le i, j \le |V|$, где $m^1_{i,j} \in M_1, m^2_{i,j} \in M_2$.
Отфильтровать матрицу M:
for $i = 1$ to $|V| - 1$ **do**
    for $j =$ i+1 to $|V|$ **do**
        if $m_{i,j} = m_{j,i} = 1$ **then**
            if $m_{i,j} = m^1_{j,i} = 1$ **then**
                $m_{j,i} = 0$
            else
                $m_{i,j} = 0$
            end if
        end if
    end for
end for
E^*_2 – это множество рёбер, представимых матрицей смежности M.
E_2 – это транзитивное сокращение E^*_2.
вернуть $mv = (V, E_1, E_2)$

Рис. 3. Алгоритм объединения метаграфов

Алгоритм 2 Алгоритм пересечения метаграфов

Input метаграфы $mv_1 = \{V_1, E_{1,1}, E_{1,2}\}$ и $mv_2 = \{V_2, E_{2,1}, E_{2,2}\}$
Output метаграф $mv = \{V, E_1, E_2\}$

1: $V = V_1 \cap V_2$
2: $E_1 = E_{1,1} \cap E_{2,1}$
3: $E^*_{1,2}$ – это транзитивное замыкание $E_{1,2}$ на V
4: $E^*_{2,2}$ – это транзитивное замыкание $E_{2,2}$ на V
5: Создать матрицы смежности M^1 и M^2 размера $|V|$ на $|V|$, характеризующие $E^*_{1,2}$ и $E^*_{2,2}$, i-ые строки M^1 и M^2 соответствуют одной вершине $v_i \in V$, j-ые колонки соответствуют $v_j \in V$.
6: Создать матрицу $M = [m_{i,j} = m^1_{i,j} \wedge m^2_{i,j}], 1 \le i, j \le |V|$, где $m^1_{i,j} \in M_1, m^2_{i,j} \in M_2$.
7: E^*_2 – множестве рёбер, представимых матрицей смежности M.
8: E_2 – это транзитивное сокращение E^*_2.
9: вернуть $mv = (V, E_1, E_2)$

Рис. 4. Алгоритм пересечения метаграфов

Таблица 1.

Оценка временной сложности частей алгоритмов

Объединение метаграфов		Пересечение метаграфов	
Операция	Временная сложность	Операция	Временная сложность
Объединение множеств V_1 и V_2	$O(\lvert V_1\rvert + \lvert V_2\rvert) = O(\lvert V\rvert)$	Пересечение множеств V_1 и V_2	$O(\lvert V_1\rvert + \lvert V_2\rvert) = O(\lvert V\rvert)$
Объединение множеств $E_{1,1}$ и $E_{2,1}$	$O(\lvert V\rvert^2 + \lvert V\rvert^2) = O(\lvert V\rvert^2)$	Пересечение множеств $E_{1,1}$ и $E_{2,1}$	$O(\lvert V\rvert^2 + \lvert V\rvert^2) = O(\lvert V\rvert^2)$
Построение транзитивных замыканий $E^*_{1,2}$ и $E^*_{2,2}$	$O(\lvert V\rvert^3 + \lvert V\rvert^3) = O(\lvert V\rvert^3)$	Построение транзитивных замыканий $E^*_{1,2}$ и $E^*_{2,2}$	$O(\lvert V\rvert^3 + \lvert V\rvert^3) = O(\lvert V\rvert^3)$
Заполнение матриц M^1 и M^2	$O(\lvert V\rvert^2 + \lvert V\rvert^2) = O(\lvert V\rvert^2)$	Заполнение матриц M^1 и M^2	$O(\lvert V\rvert^2 + \lvert V\rvert^2) = O(\lvert V\rvert^2)$
Заполнение матрицы M	$O(\lvert V\rvert^2)$	Заполнение матрицы M	$O(\lvert V\rvert^2)$
Фильтрация матрицы M	$O(\lvert V\rvert^2)$		
Построение транзитивного сокращения E^*_2	$O(\lvert V\rvert^3 + \lvert V\rvert^3) = O(\lvert V\rvert^3)$	Построение транзитивного сокращения E^*_2	$O(\lvert V\rvert^3 + \lvert V\rvert^3) = O(\lvert V\rvert^3)$

ния и объединения данных метаграфов представлены на рисунке 7 и 8. На рисунках 9 и 10 представлены результат работы программы при выполнении операции объединения и пересечения соответственно. Алгоритмы были реализованы на языке программирования golang.

Как можно видеть, множество рёбер красного графа, рассчитанное теоретически и практически, совпадают для обоих операций, аналогичный вывод можно сделать и для рёбер синего графа. Следовательно, результаты выполнения операций над метаграфами, рассчитанные теоретически и с помощью программы, идентичны, поэтому можно сделать вывод о том, что программа работает корректно.

Проведём анализ времени выполнения программы, а также количества операций, затрачиваемых на вычисление результата. Будем выполнять алгоритмы объединения и пересечения метаграфов для метаграфов различных размерностей. При этом, будем засекать время выполнения программы и считать число выполняемых операций. На основе этих данных мы построим таблицу 2, а также графики, представленные на рисунках 10 и 11.

На графиках представлены результаты выполнения алгоритма — временная сложность.

На приведенных графиках черными точками обозначаются результаты измерений, а серым цветом обозна-

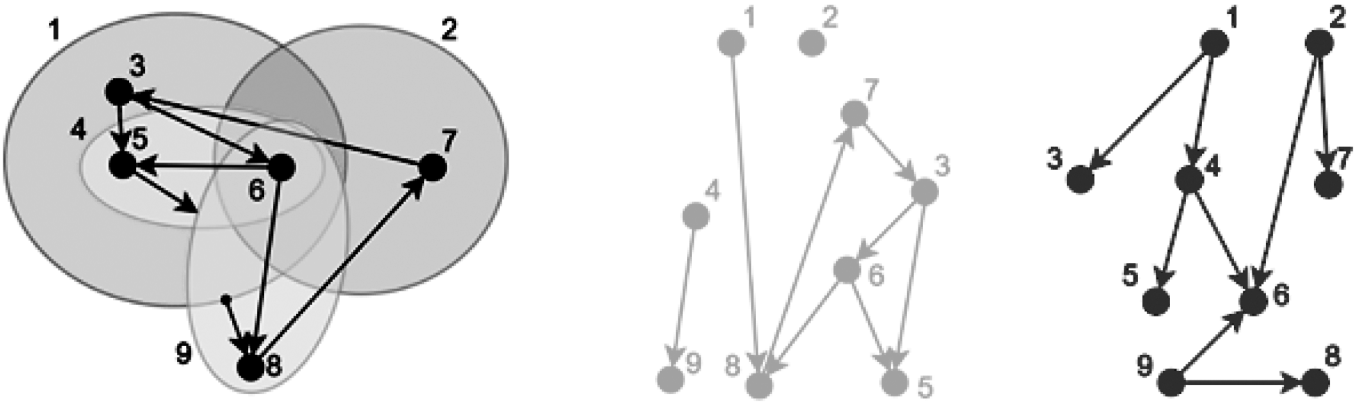

Рис. 5. Первый метаграф-операнд

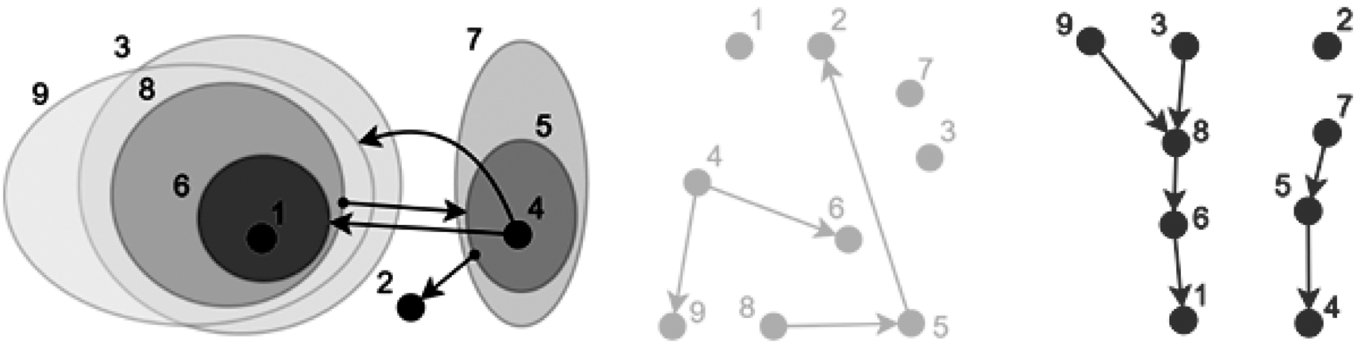

Рис. 6. Второй метаграф-операнд

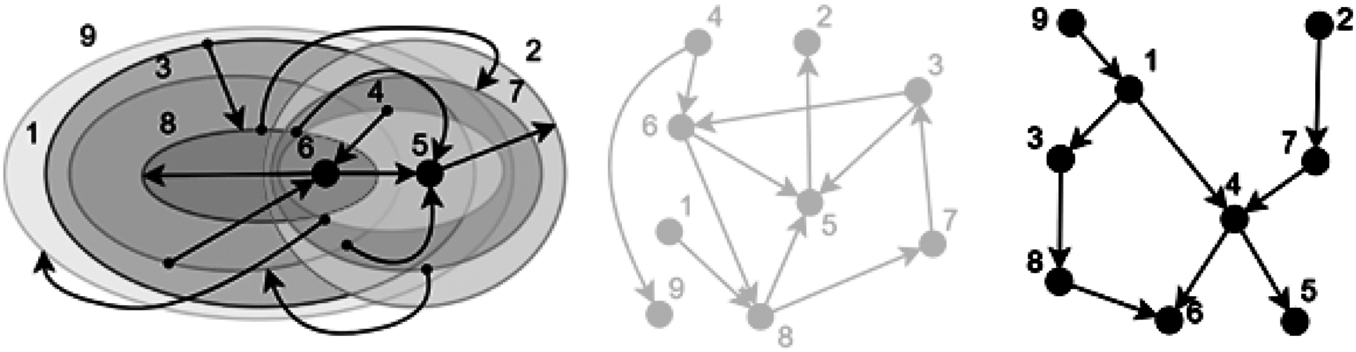

Рис. 7. Теоретический результат объединения метаграфов

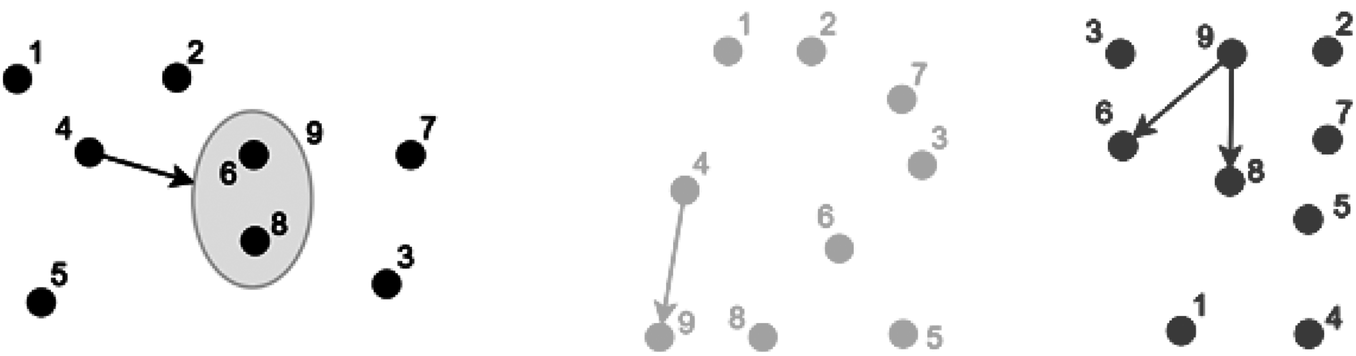

Рис. 8. Теоретический результат пересечения метаграфов

```
Результат объединения метаграфов:

Множество вершин:
(v1 v2 v3 v4 v5 v6 v7 v8 v9)

Множество рёбер красного графа:
(v7,v3),(v1,v8),(v6,v5),(v6,v8),(v4,v6),(v4,v9),(v5,v2),(v8,v5),(v8,v7),(v3,v5),(v3,v6)

Множество рёбер синего графа:
(v4,v5),(v4,v6),(v8,v6),(v7,v4),(v9,v1),(v1,v3),(v1,v4),(v2,v7),(v3,v8)
```

Рис. 9. Результат объединения метаграфов, рассчитанный с помощью программы

```
Результат пересечения метаграфов:

Множество вершин:
(v1 v2 v3 v4 v5 v6 v7 v8 v9)

Множество рёбер красного графа:
(v4,v9)

Множество рёбер синего графа:
(v9,v6),(v9,v8)
```

Рис. 10. Результат пересечения метаграфов, рассчитанный с помощью программы

чен график функции $f(x) = x^3 / 100$. Как можно заметить, набор точек, характеризующих результаты экспериментов, практически идеально ложится на график функции, что подтверждает корректность теоретической оценку временной сложности алгоритмов.

Таблица 2.

Результаты выполнения программы

Операция пересечения			Операция объединения		
Объем входных данных	Время выполнения программы (мкс)	Число выполненных операций	Объем входных данных	Время выполнения программы (мкс)	Число выполненных операций
5	0.05	2339	5	0.07	2238
10	2	15561	10	5	15214
20	530	111633	20	139	109664
30	627	360816	30	453	355816
50	2498	1609671	50	1239	1595952
70	5685	4340481	70	3457	4314014
80	4498	6442995	80	6857	6407511
100	12063	12477397	100	13730	12428014
200	81928	98113555	200	61913	97909545
300	259568	329068416	300	221013	328552578
500	1527733	1515168708	500	1104339	1513755476
650	2874252	3322012281	650	2376427	3319675520
800	5029440	6185341086	800	4581036	6181823592
900	6741087	8801138687	900	6513250	8796686613
1000	9233832	12374324675	1000	9228915	12362574328
1100	12526421	17121735216	1100	11904734	17092474322
1200	16467558	23738525614	1200	15453222	23705052731

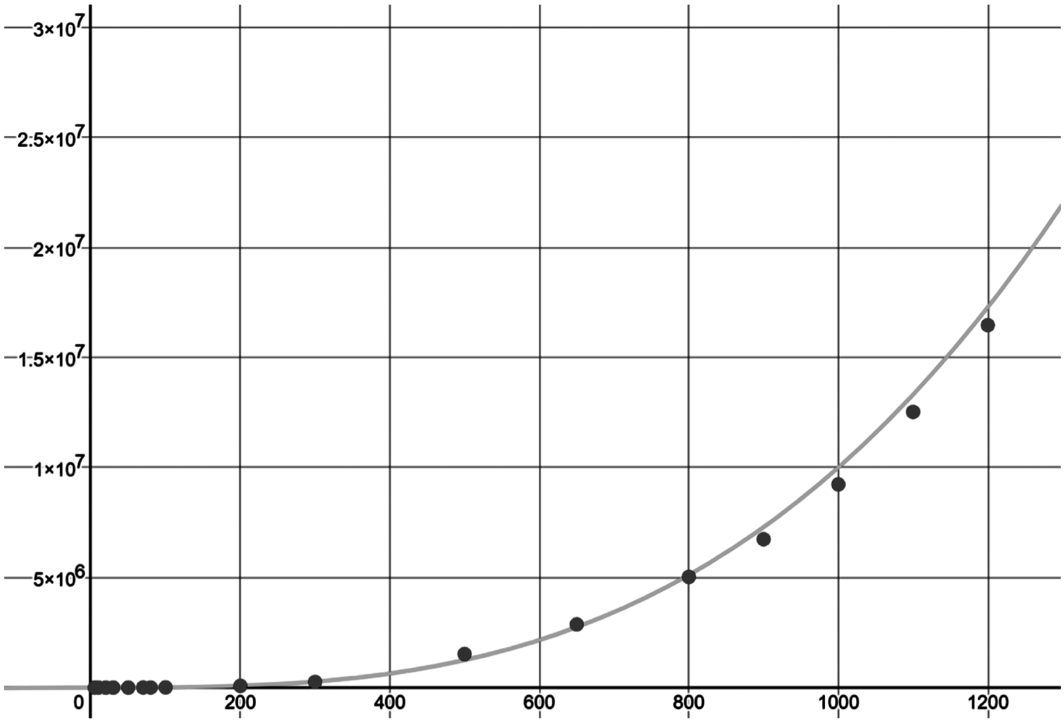

Рис. 11. График зависимости времени выполнения программы от объема входных данных для алгоритма объединения метаграфов

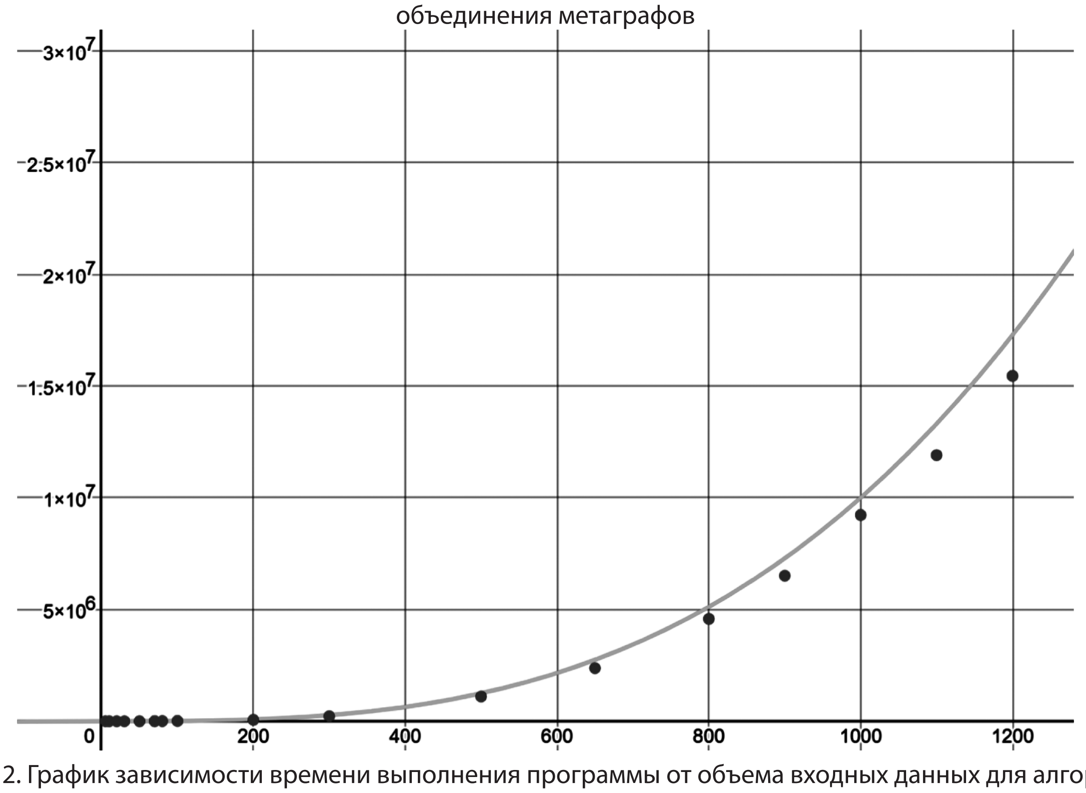

Рис. 12. График зависимости времени выполнения программы от объема входных данных для алгоритма пересечения метаграфов

Заключение

Метаграфовая модель данных — это современный инструмент для моделирования систем различной степени сложности. Помимо различных данных о характеристиках описываемой системы, метаграфы хранят информацию об иерархии этой системы, что выгодно отличает ее от других методов моделирования.

Переход от классического описания метаграфов к новому позволил удобно и лаконично описывать операции над метаграфами, а также упростил работу с данной структурой. Полученные результаты исследования применимы для дальнейшего развития метаграфовой модели данных.

Описанные в работе алгоритмы объединения и пересечения метаграфов, а также реализованная программа позволяют работать с метаграграфами не только на теоретическом, но и на практическом уровне. Эксперимент, проведенный с помощью программы, подтвердил рассчитанную теоретически асимптотическую временную сложность алгоритмов.

Дальнейшие исследования будут связаны с описанием алгебр метаграфов, изучением их свойств, описанием и оптимизацией алгоритмов на них.

ЛИТЕРАТУРА

1. Chapela, V., Criado, R., Moral, S., Romance, M. Intentional risk management through complex networks analysis. Springer International Publishing, Berlin/Heidelberg (2015)
2. Voloshin, V.I. Introduction to Graph and Hypergraph Theory. Nova Science Publishers (2009)
3. Johnson, J. Hypernetworks in the science of complex systems (Vol. 3). World Scientific Publishers (2013)
4. Basu, A., Blanning, R.W.: Metagraphs and Their Applications. Springer (2007)
5. Gapanyuk, Yu. The Development of the Metagraph Data and Knowledge Model. In: Russian Advances in Fuzzy Systems and Soft Computing: Selected Contributions to the 10th International Conference on «Integrated Models and Soft Computing in Artificial Intelligence (IMSC–2021)», pp. 1–7. Kolomna, Russia, May 17–20 (2021)
6. Ehrig, H. et al. Graph and Model Transformation. Springer (2015)
7. Baader, F., Horrocks, I., Sattler, U. Description Logics as Ontology Languages for the Semantic Web. In: Hutter, D., Stephan, W. (eds.) Mechanizing Mathematical Reasoning 2005. LNCS, vol. 2605, pp. 228–248. Springer, Berlin, Heidelberg (2005). https://doi.org/10.1007/978-3-540-32254-2_14
8. Chernenkiy, V., Gapanyuk, Y., Nardid, A., Skvortsova, M., Gushcha, A., Fedorenko, Y., Picking, R. Using the metagraph approach for addressing RDF knowledge representation limitations. In: 2017 Internet Technologies and Applications, ITA 2017 — Proceedings of the 7th International Conference, pp. 47–52. (2017)
9. Grami, A.: Discrete Mathematics: Essentials and Applications. Academic Press (2022)