|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство образования и науки Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ «Информатика, искусственный интеллект и системы управления» (ИУ)

КАФЕДРА «Системы обработки информации и управления» (ИУ5)

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

***К КУРСОВОМУ ПРОЕКТУ***

***НА ТЕМУ:***

**«Сравнение метаграфа и простого графа»**

Студент группы ИУ5-33М **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** Е. А. Белоусов

Руководитель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** Ю. Е. Гапанюк

*2022 г.*

Оглавление

[Введение 2](#_Toc121683352)

[Простой граф 2](#_Toc121683353)

[Метаграф 2](#_Toc121683354)

[Ограничения метаграфовой модели 3](#_Toc121683355)

[Сравнение метаграфа и простого графа 4](#_Toc121683356)

[1. Количество отображаемых связей 5](#_Toc121683357)

[9 сущностей 5](#_Toc121683358)

[10 сущностей 6](#_Toc121683359)

[11 сущностей 7](#_Toc121683360)

[12 сущностей 8](#_Toc121683361)

[13 сущностей 9](#_Toc121683362)

[14 сущностей 10](#_Toc121683363)

[Минусы 13](#_Toc121683364)

[2. Структурированность 15](#_Toc121683365)

[Минусы 16](#_Toc121683366)

[3. Время считывания всей модели графа 16](#_Toc121683367)

[Минусы 18](#_Toc121683368)

[4. Время поиска вершины в графе 18](#_Toc121683369)

[Сравнение 20](#_Toc121683370)

[Выводы 26](#_Toc121683371)

[Список литературы 27](#_Toc121683372)

# **Введение**

Метаграфовая модель, на сегодняшний день, является одной из наиболее интересных и перспективных моделей сложной сети. Однако, как и любые другие модели данная модель может более или менее подходить для описания некой системы в зависимости от характеристик и особенностей этой системы. В данной работе мы постараемся определить, с какой моделью (метаграфом или простым графом0 человеку будет проще взаимодействовать в тех или иных ситуациях.

# **Простой граф**

**Граф** — математическая абстракция реальной системы любой природы, объекты которой обладают парными связями. Граф как математический объект есть совокупность двух множеств — множества самих объектов, называемого множеством вершин, и множества их парных связей, называемого множеством рёбер. Элемент множества рёбер есть пара элементов множества вершин.

Графы находят широкое применение в современной науке и технике. Они используются и в естественных науках и в социальных, но наибольших масштабов применение графов получило в информатике и сетевых технологиях.

Моделируемые графами системы реальной природы обладают большим разнообразием, поэтому существуют графы различных типов. Простейшей абстракцией систем с элементами, обладающими парными связями, является простой граф.

**Простой граф** – это совокупность двух множеств – непустого множества и множества неупорядоченных пар различных элементов множества. Множество называется множеством вершин, множество называется множеством рёбер

*(1)*

то есть множество состоит из 2-элементных подмножеств множества .

# **Метаграф**

Метаграфовая модель была предложена несколько раз, и в настоящее время существует несколько ее разновидностей. Будучи изначально предложенной А. Базу и Р. Блэннингом в монографии 2007 года, модель получила ряд расширений, независимо развиваемых различными группами исследователей.

Понятие метавершины впервые появляется в работе Глобы, Тернового и Штогриной.

**Метаграф** – это тройка множеств вершин, метавершин и ребер соответственно:

(2)

– множество вершин метаграфа (порождающее множество);

M = {} – множество метавершин метаграфа;

E = {eh} – множество ребер метаграфа.

**Метавершина метаграфа**

(3)

определяется как множество вершин vr, входящих в метавершину mq, где Nmq – мощность множества.

Эмерджентность в предлагаемой модели обеспечивается за счет использования метавершин. Наличие у метавершин собственных атрибутов и связей с другими (мета)вершинами как раз и соответствует принципу эмерджентности, то есть приданию понятию метавершине нового качества, несводимости понятия-метавершины к сумме ее составных частей. Фактически, как только вводится новое понятие в виде метавершины, оно «получает право» на собственные свойства, связи и т.д., так как в соответствии с принципом эмерджентности новое понятие обладает новым качеством и не может быть сведено к подграфу базовых понятий.

Таким образом, метаграф можно описать как «сложный граф с эмерджентностью» или «сложную сеть с эмерджентностью», в котором элемент сети, состоящий из вершин и связей, может при этом сам быть вершиной.

# **Ограничения метаграфовой модели**

Метаграфовая модель подходит для описания сложных систем, в которых имеет место отношение «является частью». При этом, в моделируемых системах должно проявляться такое свойство как эмерджентность – несводимость свойств системы к сумме свойств ее компонентов. В случае, если в моделируемой системе не будут существовать отношения вида «является частью», ее можно будет описать с помощью метаграфовой модели, однако полученнуый метаграф выродится в обыкновенный простой граф и не будет иметь каких-либо конкурентных преимуществ. Соответственно, если задачей графа является показать взаимодействие одноуровневых сущностей, то скорее всего не стоит прибегать к использованию метагарфовой модели.

Также стоит отметить, что метагрфовая модель позволяет обеспечивать бесконечную вложенность (метавершина в метавершине в метавершине), однако человек сможет эффективно взаимодействовать только с несколькими уровнями вложенности одновременно (однако ничего нам не мешает переключаться на нужный уровень вложенности и смотреть именно его).

# **Сравнение метаграфа и простого графа**

Для того чтобы произвести сравнение, необходимо определить критерии, по которым мы будем оценивать модели.

Как правило, связи между объектами заключают в себе основную сложность любой модели. Чтение любого графа заключается в сопоставлении вершин друг с другом согласно связям между ними. Уменьшение количества отображаемых связей будет вести к упрощению восприятия графовой модели. Таким образом, первый критерий для сравнивания метаграфовой модели и модели простого графа в нашей работе – «Количество отображаемых связей в модели».

Все знают, что чем больше объектов приходится держать в памяти, тем менее эффективно получается с ними работать. Широко известно правило «72», согласно которому, человек, в зависимости от его особенностей, способен работать с от 5-и до 9-и сущностями. Сравним количество сущностей, с которыми необходимо одновременно работать человеку при использовании метаграфовой модели и модели простого графа в критерии «Структурированность».

Подразумевая, что графовые модели используются для описания неких систем, высоковероятна ситуация, когда человек получит наш граф в качестве документации по системе. В зависимости от сложности системы время, которое потратит человек на чтение графовой модели может различаться очень сильно. Модель будет лучше, если время ее чтения будет меньше. Сравним время, за которое можно считать всю модель в критерии «Время считывания всей модели».

Наконец, вероятна ситуация, когда пользователю не надо считывать всю модель, но найти определенную вершину (например, чтобы посмотреть, с какими вершинами она связана). «Время поиска вершины в графе» – четвертый критерий для сравнения.

Итого, сравнение будем производить по следующим критериям:

1. Количество отображаемых связей в модели;
2. Структурированность;
3. Время считывания всей модели графа;
4. Время поиска вершины графа.
5. **Количество отображаемых связей**

Построим зависимость количества связей в графе от количества сущностей. Для этого построим простой граф и метаграф, которые будут описывать некую одинаковую систему. Оценим количество связей, вершин и метавершин. После этого добавим в полученную систему некую сущность и снова оценим количество связей, вершин и метавершин. Повторим этот процесс несколько раз и постараемся сделать выводы о исследуемых зависимостях.

### **9 сущностей**

Метаграф и простой граф с 9 сущностями приведены на рисунках 1 и 2.

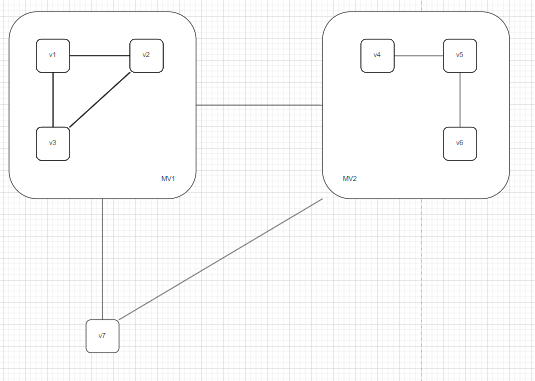


Рисунок 1. Метаграф: 7 вершин, 2 метавершины, 8 связей

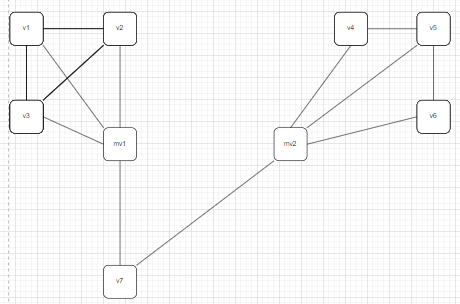


Рисунок 2. Простой граф: 9 вершин, 13 связей

**10 сущностей**

Метаграф и простой граф с 10 сущностями приведены на рисунках 3 и 4.

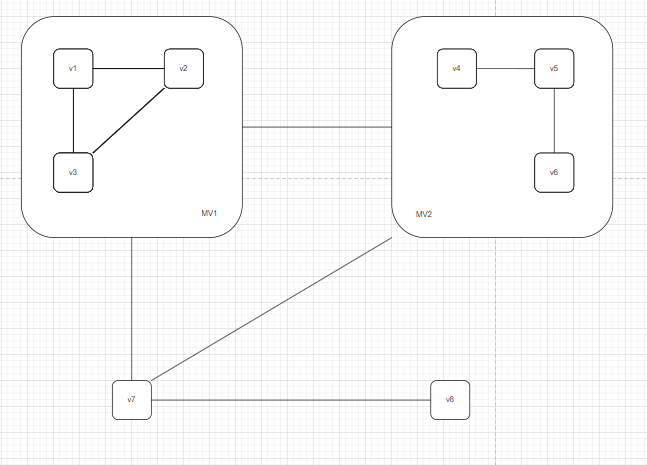


Рисунок 3. Метаграф: 8 вершин, 2 метавершины, 9 связей

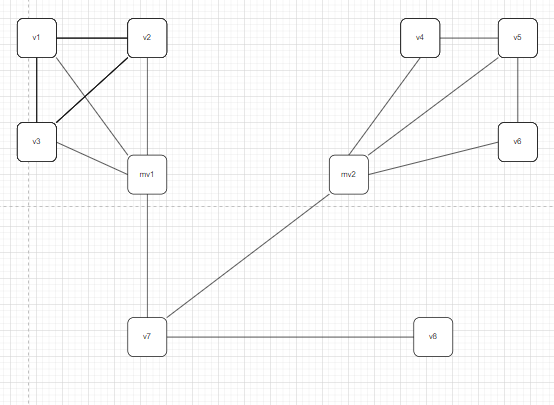


Рисунок 4. Простой граф: 10 вершин, 14 связей

**11 сущностей**

Метаграф и простой граф с 11 сущностями приведены на рисунках 5 и 6.

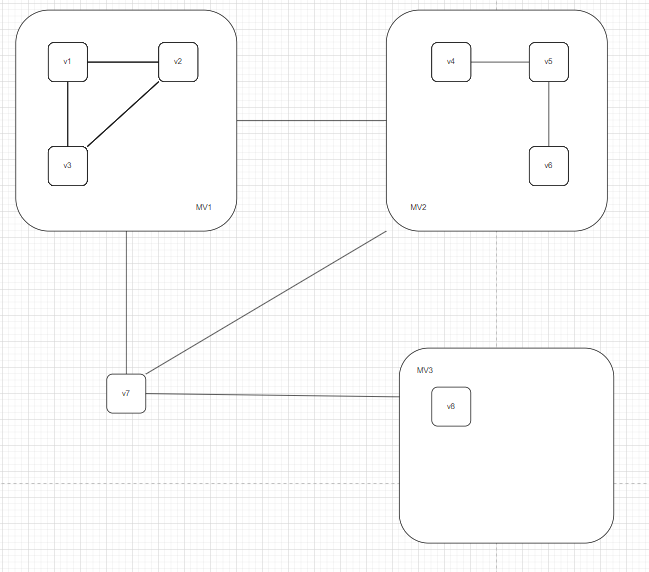


Рисунок 5. Метаграф: 8 вершин, 3 метавершины, 9 связей

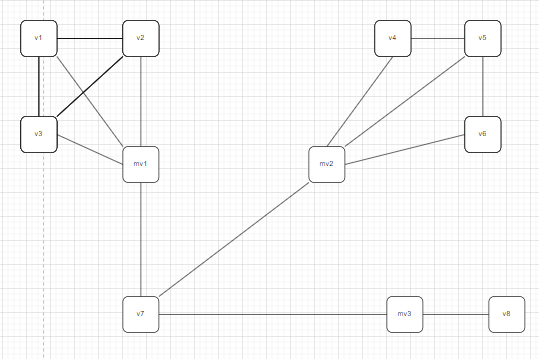


Рисунок 6. Простой граф: 11 вершин, 15 связей

**12 сущностей**

Метаграф и простой граф с 12 сущностями приведены на рисунках 7 и 8.

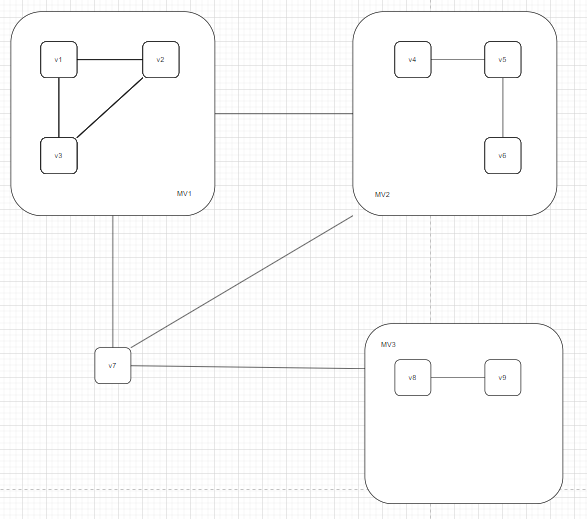


Рисунок 7. Метаграф: 9 вершин, 3 метавершины, 10 связей

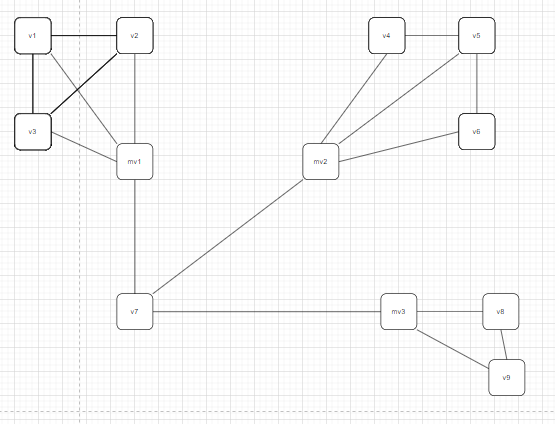


Рисунок 8. Простой граф: 12 вершин, 17 связей

**13 сущностей**

Метаграф и простой граф с 13 сущностями приведены на рисунках 9 и 10.

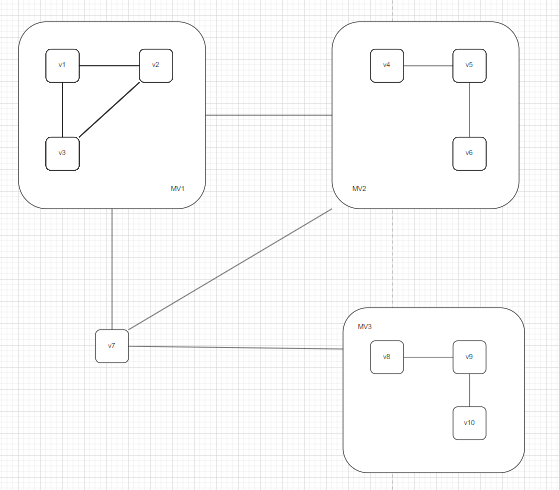


Рисунок 9. Метаграф: 10 вершин, 3 метавершины, 11 связей

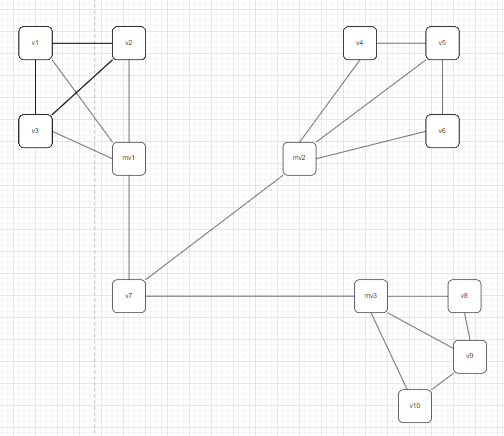


Рисунок 10. Простой граф: 13 вершин, 19 связей

**14 сущностей**

Метаграф и простой граф с 14 сущностями приведены на рисунках 11 и 12.

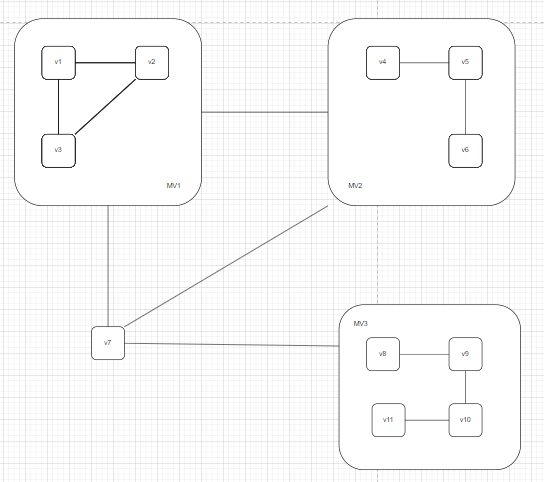


Рисунок 11. Метаграф: 11 вершин, 3 метавершины, 12 связей

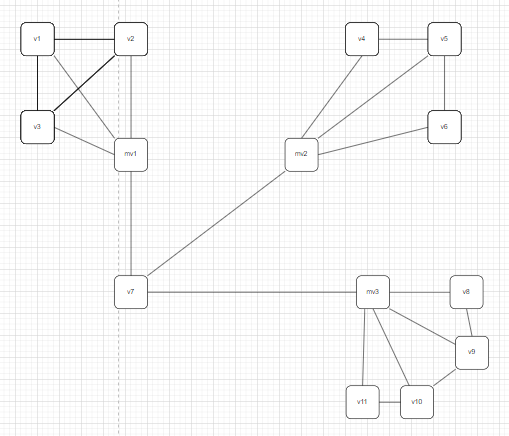


Рисунок 12. Простой граф: 14 вершин, 21 связей

В результате мы можем составить следующую таблицу:

Таблица 1. Количество связей в модели.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Количество сущностей | Количество вершин в простом графе | Количество вершин в метаграфе | Количество метавершин в метаграфе | Количество связей в простом графе | Количество связей в метаграфе |
| 9 | 9 | 7 | 2 | 13 | 8 |
| 10 | 10 | 8 | 2 | 14 | 9 |
| 11 | 11 | 8 | 3 | 15 | 9 |
| 12 | 12 | 9 | 3 | 17 | 10 |
| 13 | 13 | 10 | 3 | 19 | 11 |
| 14 | 14 | 11 | 3 | 21 | 12 |

Из таблицы 1 видна следующая зависимость:

(4)

Где - это количество вершин в моделе простого графа, количество вершин в метаграфе, – количество метавершин в метаграфе.

Количество связей в модели простого графа можно вычислить, воспользовавшись формулой для вычисления количества ребер в полном графе, где – это количество вершин простого графа:

(5)

Однако, намного чаще будет встречаться ситуация, когда вершины связаны с другими логически близкими к ним вершинами и не связаны с остальными (так как показано в примерах). Тогда количество связей можно оценить по следующей формуле:

(6)

Где – это среднее количество связей вершины или, другими словами, средний размер группы логиически близких друг к другу вершин.

Количество связей в метаграфовой модели можно вычислить по следующей формуле:

(7)

Где – количество вершин внутри i-ой метавершины,

– количество метавершин метаграфовой модели.

В случае, если в каждой метавершине находится примерно одинаковое количество вершин , получаем:

(8)

Так как в метавершине находятся логически близкие друг к другу вершины, то можно сделать следующий вывод:

(9)

Таким образом получается, что количество связей в модели простого графа можно оценить по формуле:

(10)

а, количество связей в модели метаграфа по формуле:

(11)

В рассмотренных нами примерах примерное значение k = 2,8.

Построим графики зависимостей количества связей в простом графе и метаграфе от количества сущностей в модели:

Рисунок 13. График зависимости количества связей от количества сущностей в моделях метаграфа и простого графа

**Минусы**

За счет уменьшения количества связей теряются связи между сущностями, которые находятся в разных метавершинах. Если в модели простого графа между двумя такими сущностями может существовать прямая связь, то в метаграфовой модели данная связь будет проходить через связь метавершин этих сущностей.

Например, если в примере будет существовать связь между v1 и v5, то ее без проблем можно отразить на простом графе (рисунок 14).

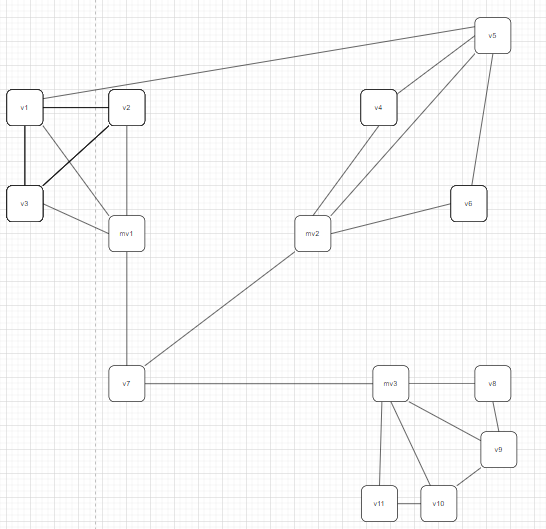


Рисунок 14. Простой граф: связь между двумя логически далекими сущностями

На метаграфе эта связь будет отображаться на уровне метавершин (рисунок 15).

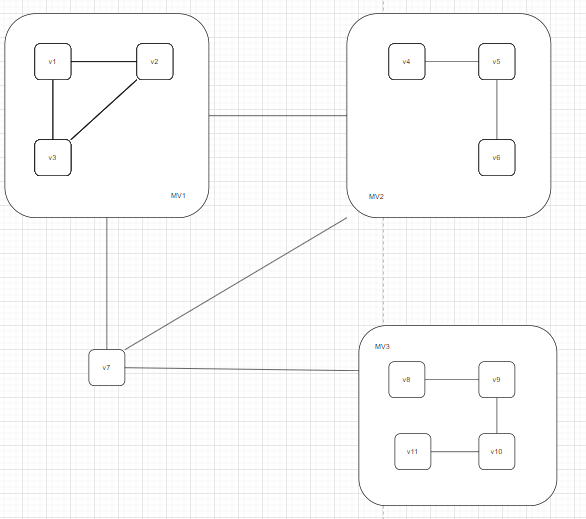


Рисунок 13. Метаграф: связь между двумя логически далекими сущностями

Чтобы «достучаться» из v1 до v5 вначале необходимо подняться на уровень метавершины mv1, потом перейти по связи до метавершины mv2 и затем спуститься до вершины v5. Таким образом, получается 3 элементарных действия вместо одного. Данное свойство метаграфовой модели передает принцип инкапсуляции.

1. **Структурированность**

Широко известен факт, что человек может контролировать 7±2 объекта. Чтобы улучшить человеческое восприятие необходимо уменьшать количество объектов, на которых человек должен сосредотачивать внимание. Рассматриваемые нами модели могут использоваться для моделирования систем, состоящих из большого количества объектов.

Метаграфовая модель подразумевает структурированность данных. Каждая метавершина отделяет вершины внутри себя от вершин снаружи. Это позволяет пользователю сосредоточиться только на внутренних вершинах конкретной метавершины при работе с моделью. Таким образом, количество объектов на которых нужно сосредотачивать внимание метаграфовой модели можно оценить следующим образом:

(12)

Где – количество метавершин;

– количество вершин*.*

Модель простого графа в общем случае не подразумевает наличия структурированности. Однако, чаще всего для восприятия графа нет необходимости сосредотачиваться на всем графе целиком. Опытный пользователь способен самостоятельно выделить вершины, на которых необходимо сосредоточить внимание. Чаще всего необходимо сосредотачивать внимание на логически близких друг к другу вершинах. Тогда из (9) получаем:

(13)

Таким образом, получается, что помимо того, что модель простого графа требует от пользователя навыков выделять важные объекты, она так же добавляет по одному элементу, на который следует обращать внимание при анализе модели.

### **Минусы**

Данное преимущество метаграфовой модели возникает за счет усложнения модели графа. Простой граф состоит из вершин и ребер между ними: двух элементарных элементов.

Метаграф состоит из вершин, метавершин и ребер, однако кроме этого неявно подразумевается отношение между метавершинами и их внутренними вершинами «является частью». Таким образом, метаграфовая модель состоит из четырех элементарных элементов. Кроме того, неявный элемент отношения «является частью» зависит от расположения вершины.

То есть, метаграфовая модель для описания систем использует большее количество элементарных элементов элементов, а сами эти элементы сложнее.

1. **Время считывания всей модели графа**

Для сравнения времени считывания простого графа и метаграфа попытаемся составить алгоритмы для их считывания и сравним времена их работы.

Алгоритм для считывания простого графа:

Для каждой вершины:

Считать вершину

Для каждой связи исходящей из вершины:

Считать связь

Считать вершину

Алгоритм для считывания метаграфа:

Для каждой метавершины:

Считать метавершину

Для каждой вершины внутри метавершины:

Считать вершину

Для каждой связи исходящей из вершины:

Считать связь

Считать вершину

Будем считать, что действие «Считать связь» выполняется за время , а действие «Считать вершину» за время . Тогда, получаем, что для первого алгоритма время считывания модели простого графа можно получить по формуле:

(14)

Где – это количество вершин в модели простого графа.

– количество связей исходящих из i-ой вершины.

В случае, если предположить, что связи между вершинами расположены равномерно, получим следующую формулу:

(15)

Где - общее количество связей в моделе простого графа.

Полученная формула точно передает смысл – основная информация любой графовой модели содержится не в вершинах, а в связях между вершинами.

Время считывания метаграфовой модели можно получить по следующей формуле:

(16)

Где - это время считывания метавершины;

– суммарное количество метавершин;

– количество вершин в i-ой метавершине.

Также как и в прошлый раз, предположим, что связи между вершинами и вершины в метавершиных в метаграфовой модели распределены равномерно, тогда получим следующую формулу:

(17)

Таким образом, получается, что формулы, по которым можно оценить время считывания моделей (15) и (17) различаются коэффициентами и временем, необходимым для считывания метавершин. Коэффициенты и будут численно равны, количество метавершин в модели, которую можно отобразить в одном графе, как правило не очень большое, однако в случае нетривиальной модели (например, больше 10 сущностей). То есть время считывания модели простого графа будет значительно большим, чем время считывания метаграфовой модели.

### **Минусы**

Если отображать на рисунке сложный метаграф с уровнем вложенности 3 и больше, то составляющая будет увеличиваться, а время считывания матаграфа – расти.

1. **Время поиска вершины в графе**

Для того, чтобы взаимодействовать с вершиной графа, ее сначала нужно найти. Поиск вершины графа не составляет труда, когда вершин не так много, однако может быть весьма внушительным при большом количестве вершин.

Для того чтобы найти вершину на простом графе необходимо поочередно просмотреть все вершины, до тех пор, пока не попадется нужная вершина. Данный поиск может быть представлен следующим алгоритмом:

Для каждой вершины:

Считать вершину

Если вершина = искомая вершина, то конец

То есть, при условии, что искомая вершина в множестве всех вершин располагается случайно, время работы данного алгоритма будет можно выразить следующей формулой:

(18)

Однако, данная формула не учитывает возможные человеческие ошибки, которые будут возникать при количестве объектов сильно больше 9. Поэтому более реальную зависимость можно с имитировать следующей формулой:

(19)

В этой формуле - время ошибки, а квадратные скобки – целая часть дроби.

Давайте теперь оценим время поиска вершины в метаграфе. Помимо обычного поиска вершины, в метаграфе, в случае его большой вложенности, будет необходимо переключаться между уровнями мететаграфа, то есть рассматривать метавершины по отдельности. Таким образом, поиск вершины в метаграфе можно реализовать следующим алгоритмом:

Для каждой метавершины:

Считать метавершину

Для каждой вершины метавершины:

Считать вершину

Если вершина = искомая вершина, то конец

В случае, когда искомая вершина располагается в метаграфе случайно, время поиска можно выразить следующей формулой:

(20)

В данном случае ошибки из-за большого количества объектов, на которые оператор должен обращать внимание, сведены к минимуму, потому что метаграф структурирован.

Стоит отметить, что время считывания метавершины занимает больше времени, чем время считывания вершины, поэтому, в случае отсутствия ошибок, поиск вершины на метаграфе занимает больше времени. Однако, когда количество вершин увеличится до количества, за которым оператору будет сложно уследить, оператор начнет совершать ошибки, что минимизирует преимущество во времени поиска вершины на простом графе по сравнению с поиском на структурированном метаграфе.

## **Сравнение**

Воспользуемся полученными формулами и построим графики зависимостей количества ребер на графе от количества сущностей, зависимость времени считывания всего графа от количества сущностей и времени поиска вершины от количества сущностей.

Зависимость количества ребер от количества вершин в простом графе приведена на рисунке 16.

Рисунок 16. График зависимости количества ребер в простом графе

Зависимость количества ребер от количества сущностей в метаграфе приведена на рисунке 17.

Рисунок 17. График зависимости количества ребер в метаграфе от количества вершин и метавершин

Для построения графиков временных зависимостей, предположим, что время считывания вершины условная единица времени, время считывания ребра условные единицы времени, время считывания метавершины условные единицы времени.

Тогда, зависимость времени считывания метаграфа от количества вершин и метавершин приведена на рисунке 18.

Рисунок 18. График зависимости времени считывания метаграфа от количества вершин и метавершин в нем

Зависимость времени считывания простого графа от количества вершин представлена на рисунке 19.

Рисунок 19. График зависимости времени считывания простого графа от количества вершин в нем

Разность времени считывания простого графа и метаграфа при одинаковом количестве сущностей представлена ниже, на рисунке 20.

Рисунок 20. График разности времени считывания простого графа и метагрфа при одинаковом количестве сущностей

На этом графике видно, что время считывания метаграфовой модели меньше, если она содержит больше вершин чем метавершин. В случае, если метаграф состоит преимущественно из метавершин, то такой метаграф будет считываться оператором дольше соответствующего ему простого графа.

Зависимость времени поиска вершины в метаграфе приведена на рисунке 21.

Рисунок 21. График зависимости времени поиска вершины в метаграфе

Из рисунка 21 видно, что при росте количества метавершин время поиска вершины возрастает.

Зависимость времени поиска вершины от количества вершин в простом графе приведена на рисунке 22.

Рисунок 22. График зависимости времени считывания простого графа

Функция времени возрастает ступеньками, потому что при росте числа вершин возрастает количество ошибок, которые допустит оператор при поиске вершины.

Разность функции времени поиска вершины в простом графе и функции времени поиска вершины на метаграфе представлена на рисунке 23.

Рисунок 23. График разности времени поиска вершины в простом графе и соответствующем ему метаграфе.

Из графика на рисунке 23 видно, что поиск вершины в простом графе производится быстрее, если в рассматриваемой модели меньше 9 сущностей. В противном случае поиск на метаграфе осуществляется быстрее.

# **Выводы**

1. Считывание метаграфа производится быстрее, если количество вершин в метаграфе больше количества метавершин.
2. Поиск вершины в метаграфе производится быстрее, если в модели больше 9 сущностей, в противном случае поиск вершины производится быстрее на простом графе.
3. Метаграф обладает структурированностью и позволяет одновременно работать с меньшим количеством объектов.

# **Список литературы**

1. Уилсон Р. Введение в теорию графов. — М.: Мир, 1977. — 208 с.
2. Харари Ф. Теория графов. — М.: Мир, 1973.
3. Оре О. Теория графов. — М.: Наука, 1968. — 336 с.
4. Basu, A. Metagraphs and their applications [Text] / A. Basu, R. W. Blanning. –– Springer, 2007. –– 172 p.
5. Глоба, Л. С. Метаграфы как основа для представления и использования баз нечетких знаний [Текст] / Л. С. Глоба, М. Ю. Терновой, Е. С. Штогрина // Открытые семантические технологии проектирования интеллектуальных систем = Open Semantic Technologies for Intelligent Systems (OSTIS-2015) : материалы V междунар. науч.-техн. конф. (Минск, 19-21 февраля 2015 года). –– Минск : БГУИР, 2015. –– С. 237––240.