DOI 10.37882/2223–2966.2020.08.03

АЛГОРИТМЫ ВОССТАНОВЛЕНИЯ К-ОДНОРОДНЫХ ГИПЕРГРАФОВ ПО ВЕКТОРУ СТЕПЕНЕЙ СВОИХ ВЕРШИН

ALGORITHMS FOR RESTORING K-HOMOGENEOUS HYPERGRAPHS BY THE VECTOR OF DEGREES OF THEIR VERTICES

I. Beretskiy
I. Irbitskiy
E. Egorova
A. Mokryakov

Summary. This paper considers homogeneous hypergraphs (complexes) and methods for their reconstruction from vertex degree vectors. Two new algorithms for implementing a vector into a complex are considered: greedy (full iteration of simplices) and reductive (construction of simplices from vertices with the largest number of incident simplices remaining). A software package for restoring hypergraphs from vertex degree vectors using both of these algorithms is implemented.

Keywords: uniform hypergraph, the restoration of the hypergraph, the greedy algorithm, the reduction algorithm.

Берецкий Игорь Сергеевич
Московский авиационный институт (национальный исследовательский университет)
ISberetskij@mai.ru
Ирбитский Илья Сергеевич
Московский авиационный институт (национальный исследовательский университет)
ILSirbitskij@mai.ru
Егорова Евгения Кирилловна
К.ф.-м.н., доцент, Московский авиационный институт (национальный исследовательский университет)
egorovaek@mati.ru
Мокряков Алексей Викторович
К.ф.-м.н., доцент, Московский авиационный институт (национальный исследовательский университет); доцент, Российский государственный университет имени А. Н. Косыгина
MokryakovAlVik@gmail.com

Аннотация. В работе рассматриваются однородные гиперграфы (комплексы) и методы их восстановления из векторов степеней вершин. Рассмотрены два новых алгоритма реализации вектора в комплекс: жадный (полный перебор симплексов) и редукционный (построение симплексов из вершин с самым большим из оставшихся количеством инцидентных симплексов). Реализован программный комплекс для восстановления гиперграфов из векторов степеней вершин, использующее оба указанных алгоритма.

Ключевые слова: однородный гиперграф, восстановление гиперграфа, жадный алгоритм, редукционный алгоритм.

Введение

Данная работа является продолжением идей, рассмотренных в работах Хакими С. П. [1], Миронова А. А. [2] и Мокрякова А. В. [3–5] о возможностях распределения вектора для n-мерного случая. В их статьях был рассмотрен одномерный случай, а также представлены алгоритмы решения задачи, результат которой обобщает результаты, полученные для одномерного случая на двумерные.

Проблема реализации гиперграфов [6–7] ряда классов из вектора степеней его вершин обозначены при рассмотрении задач о распределении ресурсов, данных в виде векторов [8]. В работе рассматриваются четыре класса гиперграфов, которые были определены в работах [9] при этом их определение было уточнено.

В своих трудах С. П. Хакими [1] поднимал проблему восстановления вектора в граф. В более поздних трудах [2–5, 8–10] идеи Хакими были расширены таким способом, что стало возможным получения всех возможных графов из исходного вектора, а не единственно из возможных. Однако при более комплексных случаях необходимо использовать понятие гиперграфа [11]. В работе [12] была предложено работать с экстремальными комплексами как с алгеброй, на которую распространены операции пересечения, дополнения и объединения. В работе [13] проведено исследование связи логических операций и вектора степеней вершин k-однородного ги-

перграфа. Работы в данной области помогли применить экстремальные гиперграфы в области криптографии [14]. Данная работа также может применяться в этом направлении.

В работе приводятся алгоритмы восстановления вектора в комплекс на вид которого были наложены ограничения. Полученные алгоритмы могут использоваться в математической модели для получения качественного результата.

1. Классы гиперграфов

Рассмотрим алгоритмы восстановления гиперграфов некоторых классов из произвольных векторов. Подход был описан при рассмотрении проблем о распределении ресурсов, определенных в виде векторов. В процессе реализации комплексов выявляется проблема их большой вариативности. Восстановить комплекс из вектора степеней вершин становится возможным при условии, что на комплекс него будут наложены ограничения.

Введем обозначение $\Gamma(k,n)$ — гиперграф на n вершинах с гиперрёбрами, которые могут содержать k смежных вершин. Также обозначим $\Gamma^1(k,n)$ — гиперграф $\Gamma(k,n)$, у которого гиперрёбра не могут содержать повторяющиеся вершины. В противовес этому в гиперграфе $\Gamma^\infty(k,n)$ каждое гиперребро может содержать до k одинаковых вершин.

Нижний индекс у $\Gamma(k,n)$ обозначает максимальный вес, который может быть у гиперребра: 1 — вес всех гиперребёр равен 1; ∞ — вес каждого гиперребра должен быть положительным целочисленным значением.

Класс $\Gamma_1^1(k,n)$ соответствует -однородному гиперграфу, именно с ними мы в дальнейшем будем работать и для удобства будем писать сокращенно: $\Gamma_1^1 = \Gamma_1^1(k,n)$.

2. Известный алгоритм восстановления k — однородного гиперграфа

Для следующего алгоритма нам потребуется ввести обозначение: $l_A(0)$ — количество координат целочисленного неотрицательного вектора $A = (a_i), i = \underline{1,n}$, равных нулю при $i \geq k$.

По работам [5, 6] известен следующий алгоритм:

Алгоритм 1

Пусть дан целочисленный вектор $Б = (a_1, \dots a_n)$, где $n \geq k$ координаты которого больше или равны нулю. Координаты вектора $Б$ упорядочены по невозрастанию. Чтобы получить гиперграф, будем по очереди отнимать от k выбранных координат по единице. При каждом вычитании будем получать новое гиперребро. Таким образом постепенно будет построен гиперграф.

Шаг 1. Пусть вектор $В = (b_1, \dots b_n)$, тогда $Б_1 = Б - В$, где

$$b_i = \{a_1 = \{a_1, \dots, a_{k-1}, n-k+1-l_Б(0)\},$$
$$i = \underline{1,k-1};\ 1,\ i = \underline{k, n-k+1+a_1}.$$

Шаг 2. Вектор $Б_2 = Б_1 - В$, где

$$b_i = \{a_2 = \{a_1, \dots, a_{k-2}, a_k, n-k-l_Б(0)\},$$
$$i = \underline{1,k-2}, i = k;\ 1, i = \underline{k+1, n-k+a_2}.$$

Шаг $n-k$. $Б_{n-k} = Б_{n-k-1} - В$, где

$$b_i = \{a_{n-k} = \{a_1, \dots, a_{k-2}, a_{n-1}, 1-l_Б(0)\},$$
$$i = \underline{1,k-2}, i = n-1;\ 1, i = n.$$

Шаг $n-k+1$. Если $a_1 = 0$, то сортируем вектор $Б_{n-k}$, по невозрастанию и переходим к шагу 1 (при построении гиперребёр учитываем, что координаты перенумерованы), иначе находим вектор $Б_{n-k-1} = Б_{n-1} - В$, где

$$b_i = \{a_{n-k+1} = \{a_1, \dots, a_{k-2}, a_k, n-k+1-l_Б(0)\},$$
$$i = \underline{1,k-2}, i = k;\ 1, i = \underline{k+1, n-k+a_{n-k+1}}.$$

В общей сложности из a_i вершины можно вычесть до C_n^k. Алгоритмы завершает свою работу, когда все варианты вычитаний перебраны или некоторый вектор $Б_p = \vec{0}$.

П р и м е р 1. Построим гиперграф с гиперрёбрами по три вершины на основе вектора $Б = (10,7,6,4,3)$.

Построим ряд гиперребёр, содержащих по три вершины:

$$\{1,2,3\}, \{1,2,4\}, \{1,2,5\}, \{1,3,4\}, \{1,3,5\}, \{1,4,5\}, \{2,3,4\}.$$

В результате получаем семь наборов по три элемента в каждом и остаток из десяти элементов (четыре первых, три вторых и два первых).

Окончательно имеем

$Б = (10,7,6,4,3)$;
$Б_1 = (9,6,5,4,3)$;
$Б_2 = (8,5,5,3,3)$;
$Б_3 = (7,4,5,3,2)$;
$Б_4 = (6,4,4,2,2)$;
$Б_5 = (5,4,3,2,1)$;

$Б_6 = (4,4,3,1,0)$;
$Б_7 = (4,3,2,0,0)$;

Легко отследить количество исходных наборов элементов, доступных для разложения.

У данного алгоритма есть недостатки:
1. В случае неудачи он не даёт однозначного ответа на вопрос о возможности восстановления гиперграфа.
2. Его реализация недостаточно проста.
3. Скорость работы алгоритма также оставляет желать лучшего.

По этим причинам было решено разработать альтернативные алгоритмы восстановления гиперграфов класса Γ_1^1.

3. Жадный алгоритм восстановления k — однородного гиперграфа

Жадный алгоритм составляет симплекс из вершин, имеющих наибольшие степени, после чего проверяется, есть ли уже в гиперграфе такой симплекс. Если нет, он добавляется в гиперграф; при наличии такого симплекса, ищется возможность составить симплекс, которого в данный момент в гиперграфе ещё нет. В общем случае алгоритм работает (если гиперграф возможно восстановить из вектора, то он гарантированно будет восстановлен), однако, имеет высокую сложность и, как следствие, большое время работы на больших векторах.

Алгоритм 2

Шаг 0. Если $vec \not\equiv 0 \ (\backslash mod\ n)$, то вектор невосстановим в комплекс.

Шаг 1. Сортировка vec по невозрастанию.

Шаг 2. Если вектор состоит из нулей, то выйти.

Шаг 3. Составить из вершин вектора (1, …, n) симплекс. Если симплекс отсутствует в комплексе, добавить его в комплекс, уменьшить степени использованных вершин на 1 и перейти к шагу 1. Если такой симплекс уже есть, удалить последнюю добавленную вершину и взять следующую, с последующим переходом к шагу 3. Если составить симплекс невозможно, удалить последний добавленный симплекс и попытаться составить новый, взяв следующую вершину вместо последней добавленной.

Плюсом алгоритма является гарантия восстановления в n-комплекс, если для входного вектора это возможно. Минусом, очевидно, медленная скорость и, как следствие, большое время работы алгоритма.

4. Фиксирующий алгоритм восстановления k — однородного гиперграфа

Фиксирующий алгоритм работает следующим образом: сначала вектор степеней вершин сортируется по невозрастанию. Далее, для -однородного гиперграфа фиксируются $(k-1)$ вершин со старшими степенями, а все оставшиеся вершины используются для составления симплексов, добавляемых в гиперграф. При обнулении старшей вершины происходит сортировка вектора, и процесс продолжается до тех пор, пока вектор не обнулится.

Алгоритм 3

Шаг 0. Если $vec \not\equiv 0 \ (mod\ n)$, то вектор невосстановим в комплекс.

Шаг 1. Сортировка vec по невозрастанию.

Шаг 2. Если вектор состоит из нулей, то выйти.

Шаг 3. Фиксация (n-1) старших вершин вектора и составление гиперрёбер со всеми оставшимися вершинами. Если во время составления гиперребра одна из зафиксированных вершин обнулится, остановиться и перейти к шагу 1.

Алгоритм работает для случаев обычного графа и большинства случаев для 3-однородного гиперграфа (каждое гиперебро объединяет 3 вершины). На данный момент найдены контрпримеры векторов для общего случая, гиперграф из которых восстановим, но алгоритм не справляется с восстановлением. Поиск улучшений для работы алгоритма в общем случае продолжается.

5. Программный комплекс

Для реализации алгоритмов был использован язык C# [23]. Этот язык является простым в освоении и относится к большому семейству языков с C подобным синтаксисом, который хорошо знаком программистам на языках C, C++, Java и JavaScript. Основными достоинствами C# можно назвать возможность его использования с другими языками (Visual Basic, F#) программирования в рамках одного приложения, типобезопасность, сборщик мусора, структурированная обработка исключений.

Для реализации клиентского приложения была выбрана технология для построения Desktop-приложений Windows Presentation Foundation (WPF).

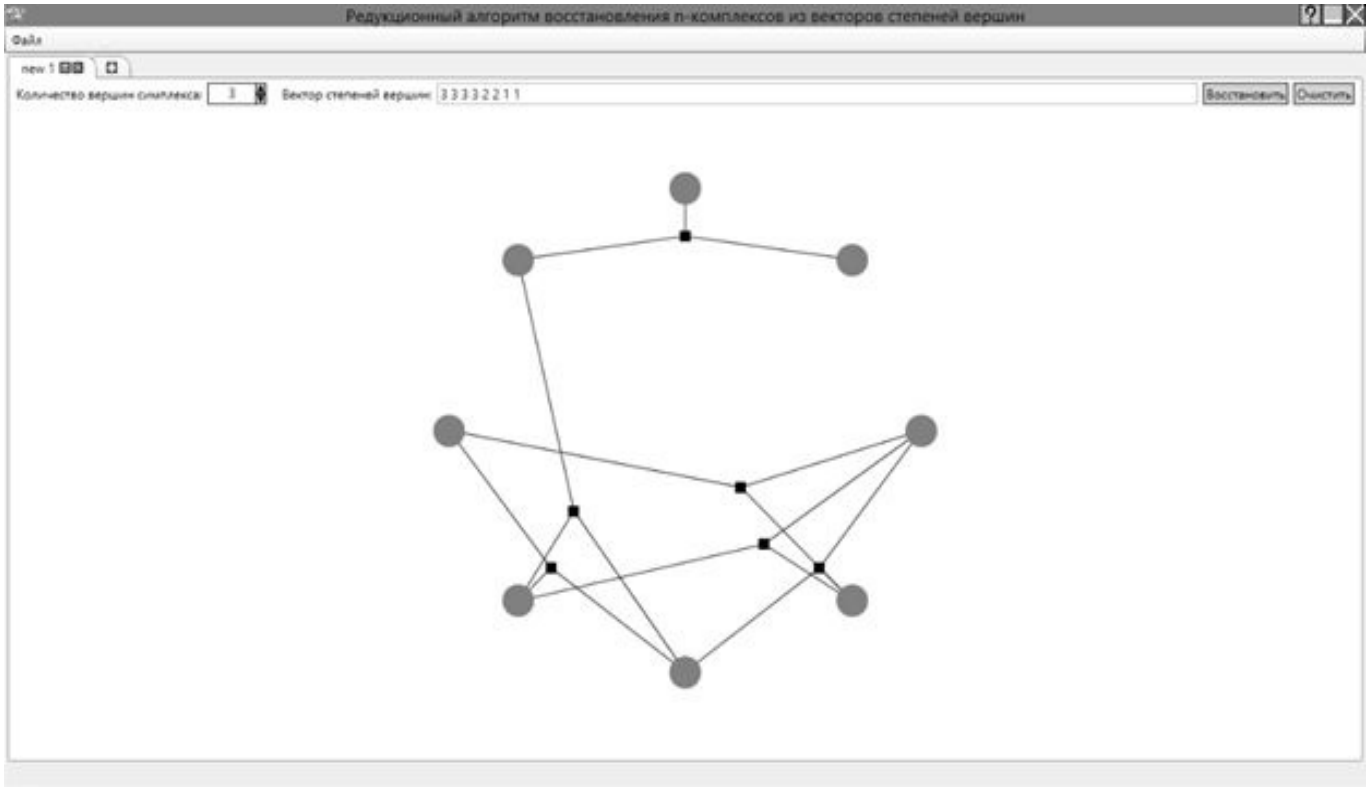

Рис. 1. Окно приложения после восстановления вектора в 3-однородный гиперграф

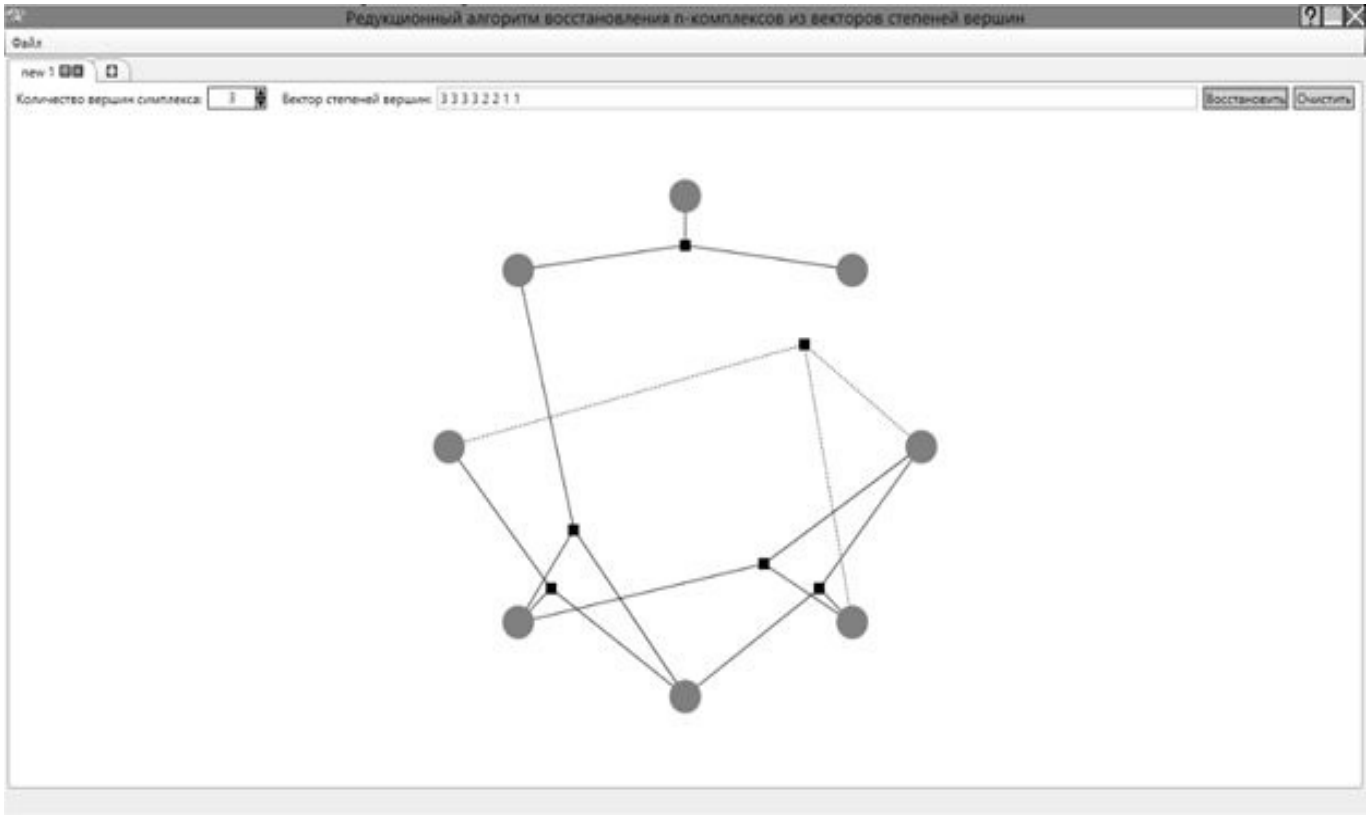

Рис. 2. Окно приложения при перетаскивании гиперрёбер 3-однородного гиперграфа

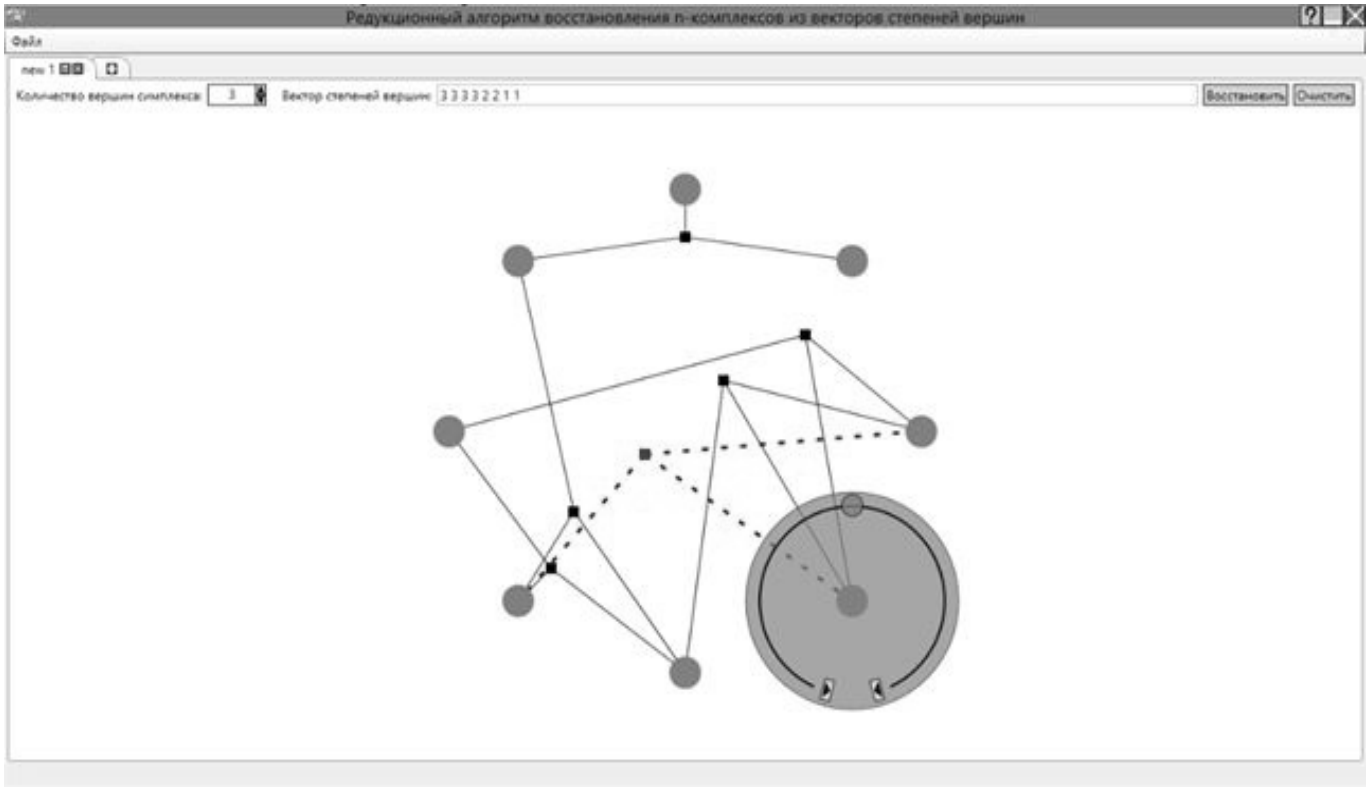

Рис. 3. Окно приложения после перетаскивании гиперрёбер 3-однородного гиперграфа

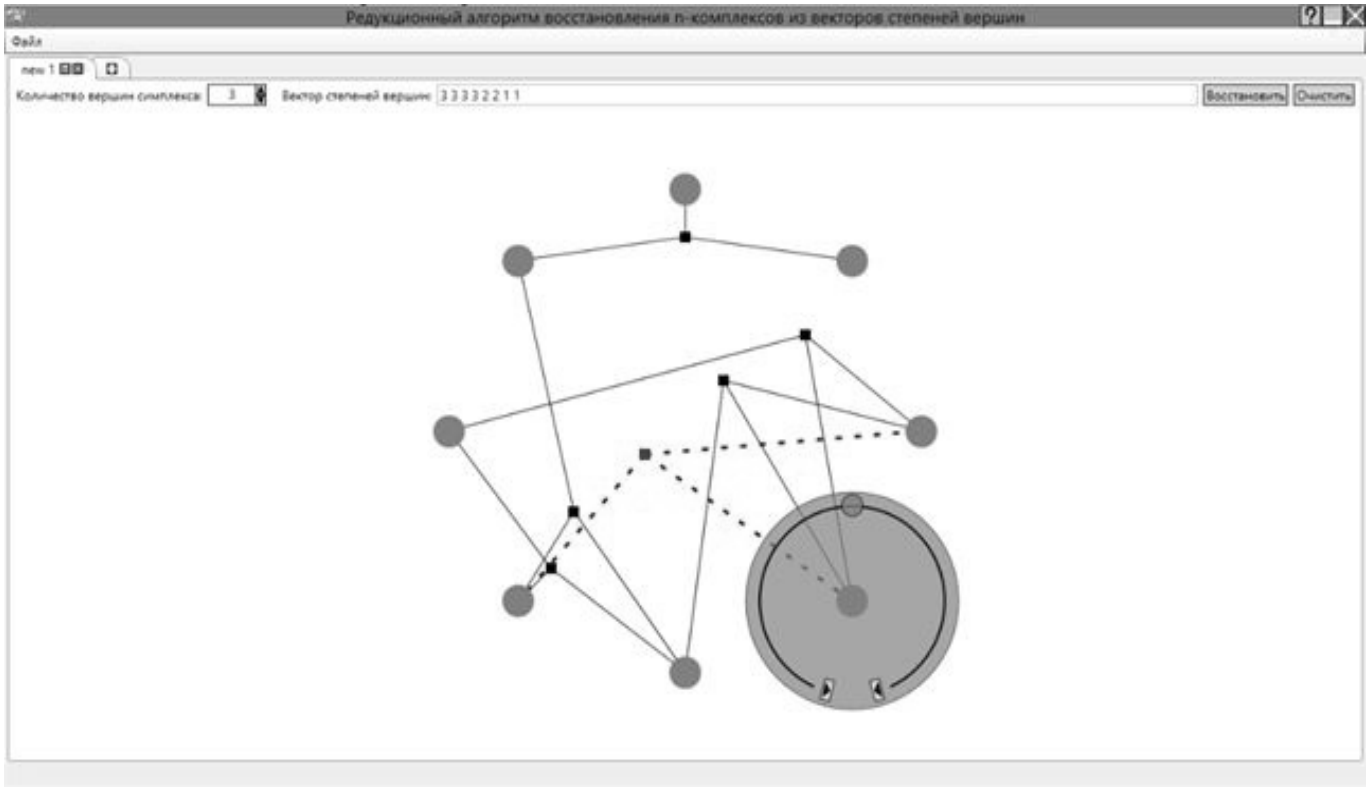

Рис. 4. Окно приложения при переключении инцидентных вершине гиперрёбер

При запуске приложения появляется окно с элементом управления TabControl с одной вкладкой. В этой вкладке (как и в остальных, создаваемых пользователем) расположены элементы управления TextBox для ввода вектора степеней вершин, NumericUpDown (реализованный самостоятельно, так как WPF не предоставляет такой элемент управления) для указания количества вершин в одном симплексе, кнопка восстановления комплекса из введённого вектора и кнопка очистки восстановленного симплекса. Под вышеуказанными элементами управления располагается отображение восстановленного комплекса.

Пользователь может очистить вектор и восстановленный комплекс при помощи нажатия кнопки "Очистить". Также пользователь может работать с гиперрёбрами (перетаскивать их) при помощи зажатия мышью квадратика (центра симплекса) и перетаскивания квадратика по рабочей области вкладки.

Особое внимание уделено именно отображению полученных гиперграфов, для чего был разработан отдельный элемент управления.

У пользователя теперь существует возможность посмотреть гиперрёбра, которые инцидентны конкретной вершине. Для этого необходимо навести на вершину мышью, и спустя небольшое время появится элемент управления со скруглённым элементом управления наподобие ScrollBar и двумя кнопками, предназначенными для перехода на предыдущий или следующий симплекс. При нажатии кнопок или передвижении ползунка элемента ScrollBar, выбранный симплекс подкрашивается в синий цвет и меняет тип линии со сплошной на штриховую.

Заключение

В дальнейшем планируется уточнить полученный редукционный алгоритм, для того чтобы он мог выступать в качестве необходимого и достаточного критерия реализуемости вектора в гиперграф.

Также интерес представляет и новый способ отображения гиперграфов, что предоставит дополнительные возможности исследователям в этой области.

ЛИТЕРАТУРА

1. Хакими С.П. О реализуемости множества целых чисел степенями вершин графа. М.: Мир, Кибернетика сб. нов. сер., вып. 2, 1966.
2. Миронов А.А. О реализуемости наборов чисел в граф и свойства графов с заданным набором степеней вершин // Тр. Гор. ГУ, 1981.
3. Миронов А.А., Мокряков А. В. Двумерные комплексы полностью описываемые степенями вершин // Труды Института системного анализа Российской академии наук. 2006. № 10. С. 178–188.
4. Mironov A.A., Mokryakov A. V., Sokolov A. A. About Realization of Integer Non-Negative Numbers Tuple Into 2-Dimensional Complexes // Applied and Computational Mathematics. 2007. T. 6. No 1. P. 58–68.
5. Mokryakov A.V., Tsurkov V. I. Reconstructing 2_Comlexes by a Nonnegative Integer_Valued Vector // Automation and Remote Control. 2011. V. 72. No 12. P. 2541–2552.
6. Зыков А.А. О некоторых свойствах линейных комплексов // Мат. сб. 1949. Вып. 24 (2). С. 163–188.
7. Зыков А. А. Гиперграфы // УМН. 1974. Т. XXIX. № 6 (180). С. 89–154.
8. Kostyanoi D.S., Mokryakov A. V., Tsurkov V. I. Hypergraph Recovery Algorithms from a Given Vector of Vertex Degrees // Journal of Computer and Systems Sciences International. 2014. T. 53. No 4. P. 511–516.
9. Гурченков А.А., Костяной Д. С., Мокряков А. В. Редукционные методы восстановления некоторого класса гиперграфов // Инженерный журнал: наука и инновации. 2014. № 6 (30). С. 1.
10. Мокряков А.В., Селин П. С., Цурков В. И. Минимакс и восстановление по вектору в графах. М.: Физматлит, 2017. 309 с.
11. Egorova E.K., Mokryakov A. V., Vang L. Development of Hypergraph Theory // Journal of Computer and Systems Sciences International. 2018. V. 57. P. 109–114.
12. Mokryakov A. V. Hypergraphs as Algebraic Structures // Journal of Computer and Systems Sciences International. 2011. T. 50. No 5. P. 734–740.
13. Егорова Е.К., Есенков А. С., Мокряков А. В. Операции над k-однородными гиперграфами и их векторы степеней вершин // Известия РАН. Теория и системы управления. 2020. № 3. С.75–80.
14. Egorova E.K., Mokryakov A. V., Suvorova A. A. The Concept of Data Encryption Using Extreme Uniform Hypergraphs // Abstracts18th International Conference "Aviation and Cosmonautics — 2019". 2019. P. 409.