

КУРС «СЛОЖНОСТИ ВЫЧИСЛЕНИЙ».

ПОИСК НАИБОЛЬШЕГО АЦИКЛИЧЕСКОГО ПОДГРАФА.

Евгений Тищенко, МФТИ, группа Б05-921

Январь 2022

Предисловие

Пусть задан ориентированный граф $G = (V, E)$. Требуется найти наибольший по числу рёбер подграф (не обязательно индуцированный), в котором нет ориентированных циклов.

В ходе данной работы мы:

1. Докажем, что задача поиска наибольшего ациклического подграфа является NP-трудной;
2. Докажем, что задача поиска ациклического подграфа, имеющего размер не меньше половины от наибольшего, решается за полиномиальное время.

1 Доказательство NP-трудности

Введём необходимые определения.

Feedback arc set - задача нахождения множества пар (G, k) , таких что в ориентированном графе G найдётся k рёбер, удалив которые, мы получим ациклический орграф.

ACYC - задача нахождения множества пар (G, k) , таких что в ориентированном графе G найдётся ациклический подграф с множеством рёбер мощности k .

Утв. 1.1 Задачи *feedback arc set* и *ACYC* эквивалентны.

Доказательство. Рассмотрим орграф G . Пусть, удалив из него k рёбер, мы получим ациклический орграф G' . Значит, в орграфе G нашёлся ациклический подграф с количеством рёбер $|E'| = |E| - k$. Обратно, если в G нашёлся ациклический подграф с k рёбрами, то удалить нужно $|E| - k$ рёбер. \square

Итак, достаточно доказать, что *feedback arc set* - NP-трудная задача.

Задача X является NP-трудной, если существует NP-полная задача Y , такая что Y сводится к X за полиномиальное время. Воспользуемся известной NP-полной задачей *VC*, которая заключается в нахождении множества пар (G, k) , таких что в графе G найдётся вершинное покрытие размера k .

Утв. 1.2 $VC \leq_p FEEDBACK\ ARC\ SET$.

Доказательство. По определению сводимости достаточно доказать следующую эквивалентность:

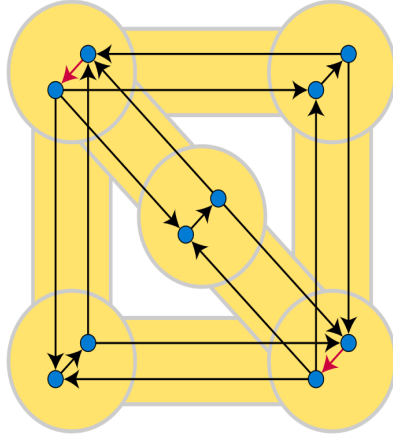
$$(G, k) \in VC \Leftrightarrow (f(G), k) \in FEEDBACK\ ARC\ SET,$$

где $f(G)$ - ориентированный граф G' , полученный из графа G за полиномиальное время.

\Rightarrow Рассмотрим граф $G = (V, E)$ с вершинным покрытием размера k . Перестроим его в ориентированный граф $G' = (V', E')$ следующим образом:

1. Каждую вершину $v \in V$ разделим на две вершины $v_0, v_1 \in V'$. Для каждой образовавшейся пары вершин проведём ребро из v_0 в v_1 . Полученные рёбра назовём *красными*;
2. Каждое ребро $(u, v) \in E$ преобразуем в рёбра $(u_1, v_0), (v_1, u_0) \in E'$. Полученные рёбра назовём *чёрными*.

Для наглядности приведём рисунок, иллюстрирующий процедуру получения графа G' с синими вершинами из жёлтого графа G .



Пусть $D \subset E'$ - множество рёбер в орграфе G' , удалив которые, получим ациклический орграф.

Утв. 1.2.1 Красные рёбра орграфа G' , соответствующие вершинам из вершинного покрытия графа G , образуют множество D .

Доказательство. Предположим противное. Тогда после удаления вышеописанных красных рёбер в орграфе G' существует цикл. В данном цикле чёрные и красные рёбра чередуются, так как чёрное ребро всегда ведёт из вершины с индексом 1 в вершину с индексом 0, а красное - из вершины с индексом 0 в вершину с индексом 1. Рассмотрим произвольное чёрное ребро (v_1, u_0) из данного цикла. Оно смежно с двумя красными рёбрами (v_0, v_1) и (u_0, u_1) . В графе G ребру (v_1, u_0) соответствует ребро (v, u) , ребру (v_0, v_1) - вершина v , ребру (u_0, u_1) - вершина u . Вершины u и v не принадлежат вершинному покрытию графа G , иначе соответствующие им рёбра (u_0, u_1) и (v_0, v_1) были бы удалены, но это означает, что ребро (v, u) не покрыто. Противоречие. \square

\Leftarrow Рассмотрим орграф G' с множеством D мощности k .

Утв. 1.2.2 Всегда можно добиться того, чтобы множество D состояло только из красных рёбер.

Доказательство. Пусть в множестве D есть чёрное ребро. Оно всегда ведёт в вершину с индексом 0. Из вершины с индексом 0 может выходить только красное ребро, причём единственное. Значит, во всех циклах, где есть упомянутое чёрное ребро, есть и смежное ему красное. Заменим чёрное ребро на красное. Повторим процедуру для всех чёрных рёбер. \square

Согласно предыдущему утверждению можно считать, что множество D состоит только из красных рёбер.

Утв. 1.2.3 Множество вершин в графе G , соответствующих рёбрам из множества D , образует вершинное покрытие графа G .

Доказательство. Предположим противное. Тогда в графе G есть непокрытое ребро (u, v) , которому в орграфе G' соответствует цикл $C: u_0 u_1 v_0 v_1 u_0$. Так как ребро (u, v) не покрыто, рёбра (u_0, u_1) и (v_0, v_1) не принадлежат множеству D . Рёбра (u_1, v_0) и (v_1, u_0) не лежат в D , так как они чёрные. Удалим из графа G' все рёбра множества D . Заметим, что цикл C сохранился. Противоречие. \square

Таким образом, мы доказали основное утверждение. \square

Осталось заметить, что для нахождения наибольшего по числу рёбер ациклического подграфа необходимо для всех пар (G, k) , $k \in \{0, 1, \dots, |E|\}$, проверить $(G, k) \in ACYC$ и выбрать максимальное k , при котором ответ положительный. Данная процедура занимает полиномиальное время от размера входных данных.

2 Алгоритмы для приближенного решения задачи

Опишем базовый алгоритм, а также докажем его корректность и полиномиальность:

Algorithm 1 Поиск приближенного наибольшего ациклического подграфа

```

Произвольным образом занумеруем вершины графа  $G = (V, E)$ 
 $A \leftarrow \emptyset, B \leftarrow \emptyset$ 
for  $(u, v) \in E$  do
    if  $u < v$  then
         $A = A \cup \{(u, v)\}$ 
    else
         $B = B \cup \{(u, v)\}$ 
    end if
end for
 $G_{ans}$  содержит ребра наибольшего по мощности множества  $A$  или  $B$ 

```

Утв. 2.1 Каждый из полученных графов ацикличесен.

Доказательство. Рассмотрим множество ребер A . Предположим, в нем есть цикл $v_0v_1 \dots v_kv_0$. По построению получаем цепочку неравенств:

$$v_0 < v_1 < \dots < v_k < v_0$$

Противоречие. Для множества B аналогично. \square

Утв. 2.2 *Найденный подграф имеет размер не менее половины от наибольшего.*

Доказательство. Пусть $\beta(G)$ - размер наибольшего ациклического подграфа в G , $\beta(G) \leq |E|$. Без ограничения общности A - найденный подграф. По построению $E = A \sqcup B$. По принципу Дирихле $|A| \geq \frac{|E|}{2}$.

$$\frac{|A|}{\beta(G)} \geq \frac{|E|}{2|E|} = \frac{1}{2}$$

\square

Утв. 2.3 *Описанный алгоритм работает за полиномиальное от числа рёбер время.*

Доказательство. Нетрудно заметить, что данный алгоритм работает за $O(|E|)$, так как мы 1 раз проходим по всем рёбрам. \square

Описанный выше алгоритм уже дает неплохое приближение решения задачи, однако его можно улучшить, а именно фиксировать не произвольную перестановку вершин, а выбранную по определенному алгоритму:

Algorithm 2 Получение перестановки для базового алгоритма

```

 $s_1 \leftarrow \emptyset, s_2 \leftarrow \emptyset$ 
while  $G \neq \emptyset$  do
    while  $G$  содержит сток  $u$  do
         $s_2 \leftarrow us_2, G \leftarrow G \setminus \{u\}$ 
    end while
    while  $G$  содержит исток  $u$  do
         $s_1 \leftarrow s_1u, G \leftarrow G \setminus \{u\}$ 
    end while
    выбрать вершину  $u$ , для которой  $deg_{out}(u) - deg_{in}(u)$  максимально
     $s_1 \leftarrow s_1u, G \leftarrow G \setminus \{u\}$ 
end while
 $s \leftarrow s_1s_2$ 

```

Перед доказательством основного утверждения введем пару обозначений и рассмотрим одно замечание:

- $\deg_{in}(v)$ - входящая степень вершины v
- $\deg_{out}(v)$ - исходящая степень вершины v
- $d(v) = \deg_{out}(v) + \deg_{in}(v)$
- $\sigma(v) = \deg_{out}(v) - \deg_{in}(v)$

Так как каждое ребро (u, v) дает вклад 1 в $\deg_{in}(v)$ и $\deg_{out}(u)$, то

$$\sum_{v \in V} \deg_{in}(v) = \sum_{v \in V} \deg_{out}(v)$$

Таким образом, если в графе G существует вершина v , такая что $\sigma(v) < 0$, в G обязательно присутствует вершина w , что $\sigma(w) > 0$. А значит в графе G всегда существует вершина v , что $\sigma(v) \geq 0$.

УТВ. 2.3 Пусть граф G связный. Выбор такой перестановки приводит к построению ациклического подграфа $G' = (V, E')$, где $|E'| \geq \frac{|E|}{2} + \frac{|V|}{6}$.

Доказательство. Рассмотрим множество вершин графа G . По способу удаления из графа эти вершины можно разделить на пять множеств:

- V_1 - вершины, которые были стоками на момент удаления, причем $\deg_{in}(v) > 0$;
- V_2 - вершины, которые были изолированными на момент удаления;
- V_3 - вершины, которые были истоками на момент удаления, причем $\deg_{out}(v) > 0$;
- V_4 - вершины, для которых на момент удаления было верно следующее равенство: $\deg_{in}(v) = \deg_{out}(v)$;
- V_5 - вершины, для которых на момент удаления были верными, следующие неравенства: $\deg_{out}(v) > \deg_{in}(v) > 0$

Также для всех $1 \leq i \leq 5$ обозначим $n_i = |V_i|$, а m_i - общее число ребер, которые были удалены из графа G в процессе удаления инцидентных вершин V_i . Тогда $n = |V| = \sum_{i=1}^5 n_i$, $m = |E| = \sum_{i=1}^5 m_i$, а также $m_2 = 0$, так как V_2 состоит из изолированных вершин на момент их удаления.

Предположим, что после удаления вершины $v \in V_4 \cup V_5$ в графе G появилась изолированная вершина u . Это означает, что вершина u была соединена только с вершиной v посредством ребра (u, v) или (v, u) .

В обоих случаях это говорит о том, что вершина u была истоком или стоком до удаления v , следовательно она должна была быть удалена ранее. Аналогично мы получаем, что удаление вершин $v \in V_3$ не может привести к появлению изолированных вершин, а значит только удаление стоков ведет к появлению изолированных вершин. Так как изначально не было изолированных вершин, это означает, что каждая изолированная вершина появилась после удаления какого-то ребра, инцидентного вершине $v \in V_2$, таким образом:

$$n_2 \leq m_1$$

Рассмотрим удаление вершины $v \in V_4$. Так как мы удаляем на каждой итерации вершину с наибольшим значением $\sigma(v)$, а по вышеописанному замечанию в графе всегда найдется вершина u , причем $\sigma(u) \geq 0$, мы получаем, что $\forall u \in V \sigma(u) = 0$. Значит после удаления вершины v появятся ровно $\deg_{in}(v)$ вершин w , таких что $\deg_{out}(w) = \deg_{in}(w) + 1$ и ровно $\deg_{out}(v)$ вершин u , таких что $\deg_{in}(u) = \deg_{out}(u) + 1$. Так как после удаления вершины из V_4 не возникают изолированные вершины, то на следующей итерации алгоритма будет удалена вершина из множества $V_1 \cup V_3 \cup V_5$, а значит

$$n_4 \leq n_1 + n_3 + n_5$$

что приводит к получению следующей цепочки неравенств:

$$n = n_1 + n_2 + n_3 + n_4 + n_5$$

$$\leq 2n_1 + n_2 + 2n_3 + n_4 + 2n_5$$

$$\leq 2n_1 + n_2 + 3n_3 + 3n_5$$

Так как $n_1 \leq m_1$ (стоки, у которых выходная степень больше нуля, а значит каждая вершина удаляет хотя бы одно ребро), а также так как $n_2 \leq m_1$, мы получаем, что

$$n \leq 3(m_1 + n_3 + n_5)$$

Теперь рассмотрим ребра, которые мы не добавим в ациклический подграф. Это будут ребра, ведущие из вершин с большими номерами в вершины с меньшими. Если $v \in V_4$, после ее удаления ровно $d(u)/2$ ребер будут смотреть влево. Если $v \in V_5$, количество ребер, смотрящих влево, хотя бы $(d(u) - 1)/2$, так как $\deg_{out}(v) > \deg_{in}(v)$. При этом по построению все ребра, удаленные после удаления $v \in V_1 \cup V_3$ будут смотреть вправо,

а значит попадут в ациклический подграф. Таким образом, множество R ребер, которые не попадут в ответ, удовлетворяет неравенству:

$$\begin{aligned} |R| &\leq m_4/2 + (m_5 - n_5)/2 \\ &= m/2 - (m_1 + m_3 + n_5)/2 \\ &\leq m/2 - (m_1 + n_3 + n_5)/2 \end{aligned}$$

так как $n_3 \leq m_3$ (аналогично идее со стоками). Если подставить полученное выше неравенство, мы получим, что

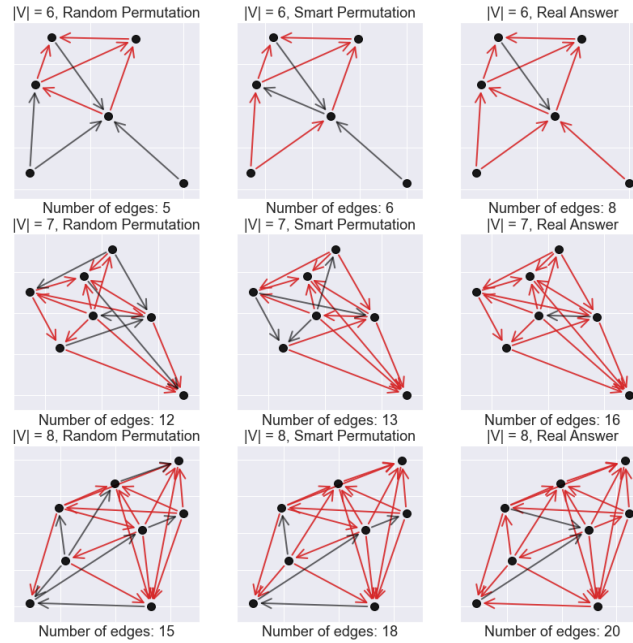
$$|R| \leq m/2 - n/6$$

а значит количество ребер, которые попадут в ациклический подграф, удовлетворяют требуемому неравенству. □

3 Реализация алгоритма и его тестирование

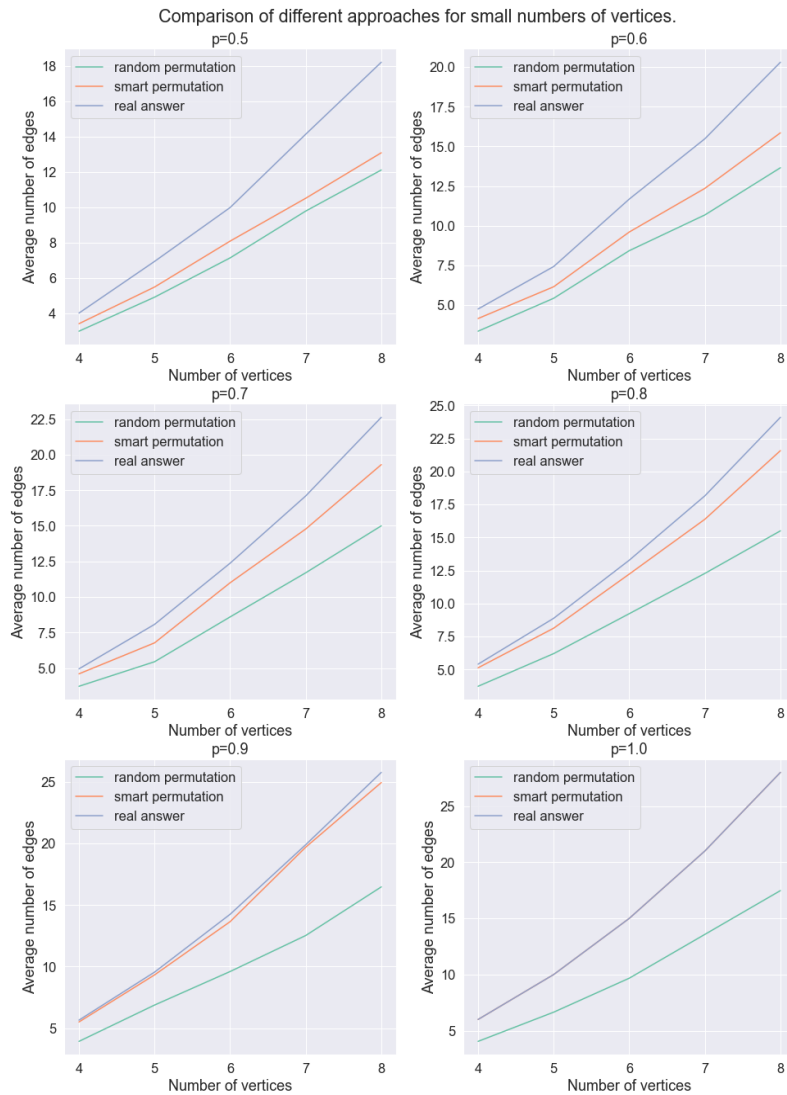
В ходе практической части были реализованы базовый алгоритм, а также более умное построение перестановки вершин. Для тестирования был имплементирован переборный алгоритм, находящий правильный ответ. Тесты проводились путем генерации случайных графов в модели Эрдеша-Реньи с различными значениями параметров n и p .

Для начала были сгенерированы случайные графы, для которых был посчитан ответ тремя разными алгоритмами:



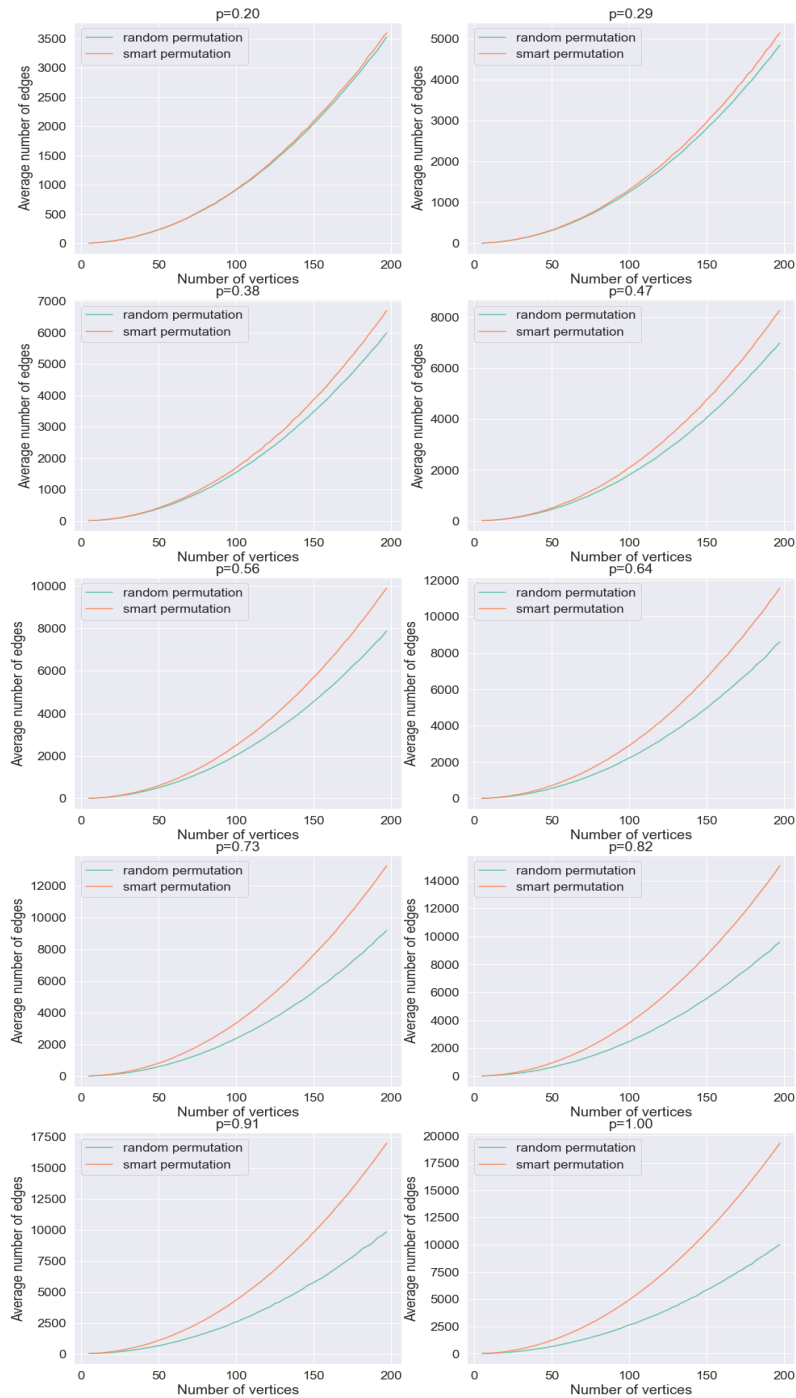
Как можно заметить, уже для малых размеров графов более умный выбор перестановки дает некое преимущество: размер ациклического подграфа получается больше.

Далее были изучены средние размеры ответа для малых размеров графов для различных значений параметра p . Данные тесты были проведены для того, чтобы можно было запускать переборный алгоритм и сравнивать результат аппроксимации с реальным ответом. Для этого для каждого значения n и p производилась генерация некоторого количества случайных графов с заданными параметрами (около 40), для которых был подсчитан ответ, используя различные подходы. Далее результаты усреднялись.

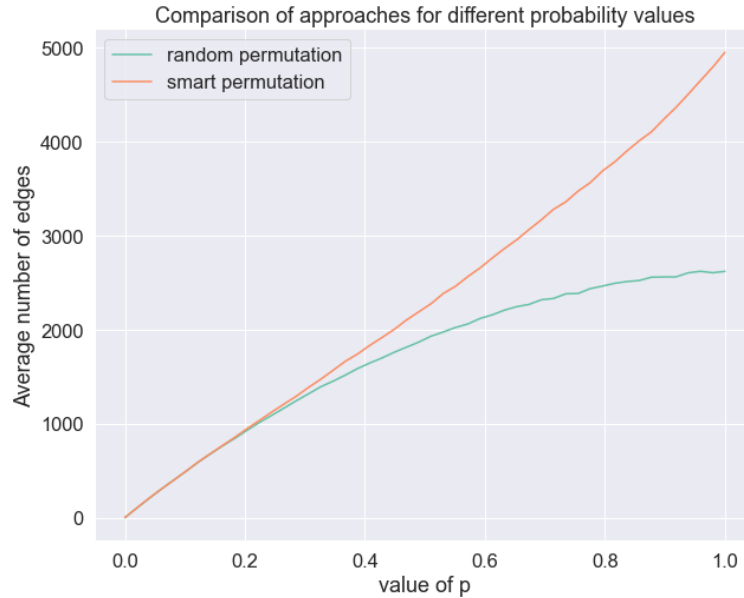


Затем аналогичная операция была проведена для графов с большим размером вершин, при этом уже переборный алгоритм не запускался, так как вычисления производились бы невероятно долго.

Comparison of different approaches for big numbers of vertices.



В конце было изучено, как влияет выбор параметра p на ответ для разных подходов. Для этого был зафиксирован определенный размер случайного графа ($n = 100$).



4 Вывод

Как можно заметить, выбор перестановки довольно существенно влияет на аппроксимацию ответа. Как видно уже на конкретно сгенерированных графах, случайная перестановка дает чуть меньший размер ациклического подграфа, в отличие от подобранной перестановки.

Далее стоит обратиться к графику, где изображена зависимость для малых размеров графов, в котором сравнивается средний ответ, полученный алгоритмом, со средним реальным ответом. Как мы видим, для всех значений вероятности p ответ получается более точный, если использовать подобранную перестановку. Также с ростом вероятности появления ребра подход с умной перестановкой дает все более точный ответ, а при $p = 1$ средний результат совпадает с реальным ответом, при этом если брать случайную перестановку, никакой динамики ответа для разных значений вероятности p не наблюдается.

Аналогичный результат продемонстрирован и для больших значений размеров графов: более умный подход дает больший ответ для всех значений n , при этом ответ увеличивается с ростом p .

Наконец, если обратиться к последнему графику, можно заметить, что существенная разница наблюдается только для не слишком разре-

женных графов: если вероятность проведения ребра не больше 0.3, то оба подхода дают одинаковые ответы. Также стоит отметить, что размер ответа, полученного умной перестановкой, растет линейно с ростом вероятности p , что нельзя сказать о случайном выборе перестановки вершин. При больших вероятностях проведения ребра случайная перестановка дает ответ, состоящий в среднем из не более, чем 3000 ребер в отличие от подхода с выбором перестановки, который строит ациклический подграф в среднем из 5000 ребер.

Таким образом, второй метод намного лучше работает для любых размеров графов, для которых вероятность проведения ребра хотя бы 0.3. При меньших вероятностях оба подхода дают примерно одинаковые ответы.

5 Список литературы

Список литературы

- [1] Approximations for the maximum acyclic subgraph problem
<http://www.shlomir.com/papers/acyclic.pdf>
- [2] A fast and effective heuristic for the feedback arc set problem.
https://researchrepository.murdoch.edu.au/id/eprint/27510/1/effective_heuristic.pdf
- [3] Feedback Arc Set Problem https://en.wikipedia.org/wiki/Feedback_arc_set