

Правительство Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Санкт-Петербургский государственный университет»

Кафедра высшей алгебры и теории чисел

Павел Евгеньевич Кунявский

Число Алкуина графа

Бакалаврская работа

Допущена к защите.
Зав. кафедрой:
проф. д.ф.м.н. Анатолий Владимирович Яковлев

Научный руководитель:
к.ф.м.н. Дмитрий Валерьевич Карпов

Рецензент:
к.ф.м.н. Федор Владимирович Фомин

Санкт-Петербург
2015

SAINT-PETERSBURG STATE UNIVERSITY

The High Algebra and Numbers Theory Department

Pavel Kunyavskiy

Alcuin number of a graph

Bachelor's Thesis

Admitted for defence.

Head of the chair:

Prof. Anatolii V. Yakovlev

Scientific supervisor:

Dmitry V. Karpov

Reviewer:

Fedor V. Fomin

Saint-Petersburg

2015

Оглавление

Введение	4
1. Базовые понятия	5
1.1. Необходимые понятия теории графов	5
1.2. Формулировка в терминах теории графов	5
2. Известные результаты	7
2.1. Структурные свойства задачи	7
2.2. Сложностные результаты	9
3. Параметризованный алгоритм	10
3.1. Решение за время $O(4^b \cdot n^{O(1)})$	10
3.2. Решение за время $O(2^{(1+\alpha)b} \cdot n^{O(1)})$	11
4. Экспоненциальные алгоритмы	13
4.1. Решение за время $O(3^{n-b} \cdot 2^b \cdot n^{O(1)})$	13
4.2. Решение за время $O((3^{n-b} + 2^b) \cdot n^{O(1)})$	13
4.3. Решение за время $O((2^{n-b} + 2^b) \cdot n^{O(1)})$	14
4.4. Решение за время $O(2^{n-b+\beta b} \cdot n^{O(1)})$	15
4.5. Комбинирование решений	15
5. Заключение	17
Список литературы	18

Введение

Задача Алкуина о перевозке через реку

Широко известна задача о перевозке волка, козы и капусты через реку. Впервые она была сформулирована в книге “Problems to sharpen the young” монаха Алкуина, жившего в 8 веке нашей эры. Задачу можно сформулировать следующим образом.

Мужчине необходимо перевезти через реку волка, козу и мешок капусты. Для этого у него есть двухместная лодка, в которую может уместиться он сам и либо волк, либо коза, либо мешок. Можно ли перевезти всех на другую сторону реки, чтобы ни волк с козой, ни коза с мешком капусты не оставались на одной стороне реки без присмотра?

Обобщение этой задачи было рассмотрено в [2]. В этой же статье было доказано, что это обобщение является NP-полной задачей, а также принадлежит классу FPT. Кроме того был представлен параметризованный размером лодки алгоритм для решения задачи со временем работы $O(4^b \cdot n^{O(1)})$, где b — размер лодки.

В этой работе представлен параметризованный размером лодки алгоритм со временем работы $O(2.4226^b \cdot n^{O(1)})$, а также экспоненциальный алгоритм со временем работы $O(1.4904^n \cdot n^{O(1)})$, что улучшает результаты из [2].

1. Базовые понятия

1.1. Необходимые понятия теории графов

Определение 1. Множество X вершин графа G называется контролирующим (вершинным покрытием), если у всех ребер графа G хотя бы один конец лежит в X .

Определение 2. Множество X вершин графа G называется независимым, если нет ребра графа G , оба конца которого лежат в X .

Обозначение 1. Множество вершин, соединенных ребром с v в графе G , будем называть множеством соседей и обозначать $N_G(v)$.

Несложно заметить, что дополнение контролирующего множества является независимым.

Обозначим размер минимального вершинного покрытия графа G , за $VC(G)$. Алгоритмы, описанные далее, будут использовать известные алгоритмы для решения задач о поиске минимального вершинного покрытия.

Обозначим за α такую константу, что задачу «существует ли в графе вершинное покрытие размера k », можно решить за время $O(2^{\alpha k} \cdot n^{O(1)})$, а за β — такую константу, что эту же задачу можно решить за время $O(2^{\beta n} \cdot n^{O(1)})$. Известно, что $\alpha \leq 0.2766$ [1], $\beta \leq 0.2632$ [3].

Заметим, что найти само вершинное покрытие можно за тоже самое время с точностью до полиномиального множителя.

1.2. Формулировка в терминах теории графов

Пусть задан граф G с множеством вершин V и множеством ребер E . Мы будем рассматривать задачу о перевозке $|V|$ вещей через реку в лодке, вмещающей перевозчика и еще b предметов, рассматривая ребра графа как запреты на то, чтобы оставить два предмета без перевозчика с одной стороны.

Планом перевозки называется указание для каждого шага, какие вещи окажутся с какой стороны и какие в лодке.

Формально, планом перевозки называется последовательность троек множеств $(L_1, B_1, R_1), (L_2, B_2, R_2) \dots (L_s, B_s, R_s)$.

Множество L_k представляет собой вещи на левом берегу реки на очередном этапе перевозки, B_k — вещи в лодке, R_k — вещи на правом берегу реки. От плана перевозки будем требовать выполнения следующих свойств:

- $L_k \sqcup B_k \sqcup R_k = V$ — каждая вещь находится либо на левом берегу, либо на правом, либо в лодке на каждом шаге;
- $|B_k| \leq b$ — на каждом шаге в лодке находится не более b вещей;

- L_k, R_k являются независимыми множества графа G — никакие две вещи, соединенные ребром, не остаются без присмотра на одном берегу
- $L_1 \sqcup B_1 = V, B_s \sqcup R_s = V$ — в начале перевозки все вещи находятся на левом берегу, а в конце на правом.
- для четных $k \geq 2$ верно $L_k = L_{k-1}$, для нечетных $k \geq 3$ выполнено $R_k = R_{k-1}$ — в лодке не оказывается вещей, которые находятся не на том же берегу, что лодка.

Определение 3. Числом Алкуина графа G ($AN(G)$) называется минимальный размер лодки, такой, что допустимый план перевозки существует.

2. Известные результаты

2.1. Структурные свойства задачи

Лемма 1. $VC(G) \leq AN(G) \leq VC(G) + 1$.

Доказательство. Если $AN(G) < VC(G)$, то невозможно выбрать корректные L_1, B_1, R_1 , потому что B_1 — контролирующее множество размера не более $AN(G)$. Если же размер лодки хотя бы $VC(G) + 1$, то план перевозки составляется тривиально. Выберем вершинное покрытие. Оно будет всегда содержаться в множестве B . Остальные вершины можно перевезти по одной. \square

Следующая теорема, доказанная в [2], дает необходимое и достаточное условие для существования плана перевозки с лодкой размера b .

Теорема 1 (структурная теорема). *В графе G существует план перевозки с лодкой размера b тогда и только тогда, когда существуют множества вершин $X_1, X_2, X_3, Y_1, Y_2 \subset V$, такие что*

1. $X = X_1 \sqcup X_2 \sqcup X_3$ — независимое множество графа G .
2. Y_1, Y_2 — непустые подмножества контролирующего множества $Y = V \setminus X$, которое имеет размер не более b . Отметим, что Y_1 и Y_2 могут пересекаться.
3. $X_1 \sqcup Y_1$ и $X_2 \sqcup Y_2$ — независимые множества графа G .
4. $|Y_1| + |Y_2| \geq |X_3|$.

Для большей наглядности, докажем достаточность этого условия. Доказательство будет состоять в явном предъявлении плана перевозки, что позволит лучше понять, в чем смысл этих 5 множеств.

Доказательство. Разобьем X_3 на два множества $X_{3,1}$ и $X_{3,2}$, так, что $|X_{3,1}| \leq |Y_1|$, $|X_{3,2}| \leq |Y_2|$. Разобьем X_1 на части $X_{1,1}, X_{1,2}, \dots, X_{1,q_1}$, каждая из которых не больше Y_1 , а X_2 на части $X_{2,1}, X_{2,2}, \dots, X_{2,q_2}$, каждая из которых не больше Y_2 . План перевозки представлен с следующей таблице:

	Левый берег	Лодка	Правый берег	Комментарий
	V			Начало перевозки
\rightarrow	X	Y		Увозим Y
\leftarrow	X	$Y \setminus Y_1$	Y_1	Оставляем Y_1 справа
\rightarrow	$X \setminus X_{1,1}$	$(Y \setminus Y_1), X_{1,1}$	Y_1	Отвозим X_1 по частям
\leftarrow	$X \setminus X_{1,1}$	$Y \setminus Y_1$	$Y_1, X_{1,1}$	и оставляем справа
...				Аналогично остальное X_1
\rightarrow	X_2, X_3	$(Y \setminus Y_1), X_{1,q_1}$	$(X_1 \setminus X_{1,q_1}), Y_1$	
\leftarrow	X_2, X_3	$Y \setminus Y_1$	Y_1, X_1	Вернулись налево
\rightarrow	$X_2, X_{3,2}$	$(Y \setminus Y_1), X_{3,1}$	Y_1, X_1	Отвезли $X_{3,1}$ направо
\leftarrow	$X_2, X_{3,2}$	Y	$X_{3,1}, X_1$	И забрали оттуда Y_1
\rightarrow	X_2, Y_2	$(Y \setminus Y_2), X_{3,2}$	$X_{3,1}, X_1$	Высадили Y_2 слева, вместо него забрали $X_{3,2}$
\leftarrow	X_2, Y_2	$(Y \setminus Y_2)$	X_3, X_1	Оставили $X_{3,2}$ справа
\rightarrow	$(X_2 \setminus X_{2,1}), Y_2$	$(Y \setminus Y_2), X_{2,1}$	X_3, X_1	Отвозим X_2 по частям
\leftarrow	$(X_2 \setminus X_{2,1}), Y_2$	$(Y \setminus Y_2)$	$X_3, X_1, X_{2,1}$	и оставляем справа
...				Аналогично остальное X_2
\rightarrow	Y_2	$(Y \setminus Y_2), X_{2,q_2}$	$X \setminus X_{2,q_2}$	
\leftarrow	Y_2	$(Y \setminus Y_2)$	X	возвращаемся за Y_2
\rightarrow		Y	X	отвозим Y направо
			V	конец перевозки

□

Также известно [2], что если существует план перевозки, то существует план, состоящий из $2|V| + 1$ перевозок.

Из структурной теоремы можно вывести ряд достаточных условий (см. [2]), более простых для проверки. Например,

Теорема 2. Если в графе есть хотя бы два минимальных вершинных покрытия, то $AN(G) = VC(G)$.

Доказательство. Пусть A_1, A_2 — минимальные вершинные покрытия. Заметим, что $|A_1 \setminus A_2| = |A_1| - |A_1 \cap A_2| = |A_2| - |A_1 \cap A_2| = |A_2 \setminus A_1|$.

Выберем $Y = A_1$, $Y_1 = Y_2 = A_1 \setminus A_2$, $X_1 = V \setminus (A_1 \cup A_2)$, $X_2 = \emptyset$, $X_3 = A_2 \setminus A_1$. Проверим выполнение условий структурной теоремы для $b = VC(G)$.

1. $X_1 \sqcup X_2 \sqcup X_3 = V \setminus A_1 = V \setminus Y$ — независимое множество.
2. $Y_1, Y_2 \neq \emptyset$, так как $A_1 \neq A_2$. $|Y| = b$.
3. $X_1 \cup Y_1 = V \setminus A_2$ и $X_2 \cup Y_2 = A_1 \setminus A_2$ — независимые, так как A_2 — контролирующее.

$$4. |Y_1| + |Y_2| = 2|A_1 \setminus A_2| \geq |A_2 \setminus A_1| = |X_3|.$$

□

2.2. Сложностные результаты

Несложно доказать NP -полноту задачи вычисления числа Алкуина. Сведём к ней задачу о вершинном покрытии. Для этого рассмотрим граф G' — объединение двух непересекающихся копий графа G . $2VC(G) = VC(G') \leq AN(G') \leq VC(G') + 1 \leq 2VC(G) + 1$. В таком случае, размер вершинного покрытия $VC(G)$, можно определить по формуле

$$VC(G) = \left\lfloor \frac{AN(G')}{2} \right\rfloor$$

В статье [2] показано, что задача лежит в классе FPT по размеру лодки. В частности, приведён алгоритм проверки существования плана размера b за время $O(4^b \cdot n^{O(1)})$.

Там же представлены полиномиальные алгоритмы для специальных классов графов, а также полиномиальная проверка на совпадение с размером вершинного покрытия на планарных графах.

3. Параметризованный алгоритм

3.1. Решение за время $O(4^b \cdot n^{O(1)})$

В качестве доказательства принадлежности классу FPT в [2] был представлен следующий алгоритм решения задачи о числе Алкуина.

Лемма 2. *Задачу «найти вершинное покрытие размера не более k , не содержащее в качестве подмножества данное вершинное покрытие Y размера b », можно решить за время $O(2^{\alpha \min(k,b)} \cdot n^{O(1)})$ или $O(2^{\beta n} \cdot n^{O(1)})$.*

Доказательство. Если $k < b$, то просто найдём любое покрытие размера k . Будем считать, что $k \geq b$.

Для доказательства сведём эту задачу к b поискам вершинного покрытия. Переберём вершину $v \in Y$, которая не будет лежать в вершинном покрытии G , которое мы ищем. Тогда все её соседи будут лежать в нем. Тогда, если искомого вершинного покрытия существует, то для хотя бы одной вершины v существует вершинное покрытие в графе $G' = G \setminus (\{v\} \cup N_G(v))$ размера не более $k - |N_G(v)|$.

Если $k - |N_G(v)| \geq b = |Y|$, то такое вершинное покрытие G' точно существует — $(Y \cap G')$. Иначе, воспользуемся алгоритмом поиска вершинного покрытия за время $O(2^{\alpha(k-|N_G(v)|)} \cdot n^{O(1)})$ (что не больше, чем $O(2^{\alpha b} \cdot n^{O(1)})$) или за время $O(2^{\beta |G'|} \cdot n^{O(1)})$ (что не больше, чем $O(2^{\beta n} \cdot n^{O(1)})$).

Значит, существование каждого из вершинных покрытий, можно проверить за необходимое время. Проверить это необходимо в b графах, что влияет только на полиномиальный множитель, откуда следует утверждение леммы. \square

Для удобства сформулируем задачу, как задачу принятия решения. Все алгоритмы ниже будут решать задачу «существует ли план перевозки в лодке размера b ?»

Из структурной теоремы сразу получается следующий алгоритм, работающий за время $O(4^b \cdot n^{O(1)})$

1. Проверим, есть ли в графе вершинное покрытие размера $b - 1$. Если есть, то план перевозки существует.
2. Проверим, есть ли в графе вершинное покрытие размера b , и найдем его, если есть. Если нет, то плана перевозки не существует.
3. Проверим, есть ли в графе G другое вершинное покрытие размера b . Если есть, то план перевозки существует.
4. Переберём все пары $Y_1, Y_2 \subset Y$.
5. В качестве X_1 возьмём вершины $V \setminus Y$, не связанные с Y_1 ,

6. В качестве X_2 — вершины $V \setminus (Y \cup X_1)$, не связанные с Y_2 .
7. В качестве X_3 все остальные вершины — $V \setminus (Y \cup X_1 \cup X_2)$.
8. Проверим для полученных множеств выполнение условия структурной теоремы. Если она выполнена, то план существует.
9. Если для всех пар $Y_1, Y_2 \subset Y$ условие структурной теоремы не выполнено, то плана не существует.

Первые три пункта алгоритма можно выполнить, за время $O(2^{\alpha b} \cdot n^{O(1)})$. Проверка каждой пары (Y_1, Y_2) занимает время $O(n^{O(1)})$, общее количество таких пар 4^b , что доказывает оценку времени работы. Известно, что $\alpha < 2$, значит, большее слагаемое — это 4^b .

3.2. Решение за время $O(2^{(1+\alpha)b} \cdot n^{O(1)})$

Лемма 3. Пусть Y — вершинное покрытие графа размера b , $Y_1 \subset Y$, Y_1 — непустое независимое множество. Пусть A — множество вершин $V \setminus Y$, которые не связаны с Y_1 . Тогда существование множеств X_1, X_2, X_3, Y_2 , выполняющих условие структурной теоремы вместе с Y_1 , эквивалентно существованию вершинного покрытия Z размера не более $b + |Y_1|$ в графе $G \setminus A$, не содержащего Y целиком.

Доказательство. В качестве доказательства достаточности явно предъявим множества.

- $X_1 = A$
- $X_2 = V \setminus (A \cup Y \cup Z)$
- $X_3 = Z \cap (V \setminus Y)$.
- $Y_2 = (Y \setminus Z)$

Проверим условие структурной теоремы.

1. X_1, X_2, X_3 попарно не пересекаются и в объединении дают $V \setminus Y$ по определению. Множества X_1, X_2, X_3 являются независимыми, как подмножество дополнения контролирующего Y .
2. Множество Y_1 не пусто по условию. Множество Y_2 не пусто, так как Z не содержит Y .
3. Множество $X_1 \cup Y_1$ независимо по условию. Множество $X_2 \cup Y_2$ независимо, как подмножество дополнения Z .

$$4. |Y_1| + |Y_2| = |Y_1| + b - |Y \cap Z| = |Y_1| + b - (|Z| - |X_3|) = |Y_1| + b - |Z| + |X_3| \geq |X_3|.$$

Таким образом, все условия структурной теоремы выполнены, что доказывает достаточность.

Для проверки необходимости покажем, что, если выполнены условия структурной теоремы, то можно выбрать $Z = X_3 \cup (Y \setminus Y_2)$, и оно будет соответствовать условиям этой теоремы (при $A = X_1$).

Так как $Y_2 \neq \emptyset$, Z не содержит Y целиком. $|Z| = |X_3| + b - |Y_2| \leq |Y_1| + |Y_2| + b - |Y_2| = |Y_1| + b$. Остается доказать, что оно контролирующее. Но оно является дополнением независимого множества $X_2 \cup Y_2$ в графе G' . \square

Теорема 3. *Задача проверки существования плана перевозки для графа размера n и лодки размера b может быть решена за время $O(2^{(1+\alpha)b} \cdot n^{O(1)})$*

Доказательство. Алгоритм состоит в проверке условий предыдущей теоремы.

Найдём вершинное покрытие Y , так же, как в алгоритме за $O(4^b \cdot n^{O(1)})$. Переберём все его подмножества Y_1 . Проверим наличие Z . По лемме 2 Z можно найти за время $O(2^{\alpha b} \cdot n^{O(1)})$.

Таким образом, время работы алгоритма составляет $O(2^{(1+\alpha)b} \cdot n^{O(1)})$. \square

4. Экспоненциальные алгоритмы

4.1. Решение за время $O(3^{n-b} \cdot 2^b \cdot n^{O(1)})$

Структурная теорема сразу даёт экспоненциальный алгоритм за время $O(4^n \cdot n^{O(1)})$.

Найдём минимальное вершинное покрытие Y в графе. Найти его можно за время $O(2^{\beta n} \cdot n^{O(1)})$. За это же время можно проверить, единственно ли оно. Если оно не единственно, то минимальный размер лодки равен минимальному вершинному покрытию, и задача решена.

Иначе, рассмотрим его дополнение X . Переберем его разбиение на $X_1, X_2, X_3 \subset X, X_1 \sqcup X_2 \sqcup X_3 = X$. Количество таких разбиений 3^{n-b} .

Для каждого разбиения надо найти как можно большие Y_1 и Y_2 , и проверить условие структурной теоремы. Заметим, что выбор Y_1 не накладывает никаких ограничений на выбор Y_2 , а значит их можно максимизировать независимо. Тогда сначала переберем все возможные Y_1 , а потом все возможные Y_2 . И то, и то можно сделать за $O(2^b \cdot n^{O(1)})$. Итого, общее время работы алгоритма $O(3^{n-b} \cdot 2^b \cdot n^{O(1)})$.

4.2. Решение за время $O((3^{n-b} + 2^b) \cdot n^{O(1)})$

Рассмотрим подробнее выбор оптимального Y_1 . Выбор X_1 запрещает вершины из Y , в которые из него есть ребра. При этом на остальных вершинах остаётся только одно условие — отсутствие ребер между ними. То есть задача выбора оптимального Y_1 (и аналогично Y_2), является поиском максимального независимого подмножества в некотором подмножестве Y . Кроме того, заметим, что от Y_1 и Y_2 теперь нам нужен только размер.

Научимся решать эту задачу за время 2^b для всех подмножеств Y сразу. Тогда все эти результаты можно будет запомнить, и обрабатывать каждое разбиение на множества за $O(n^{O(1)})$.

Для этого воспользуемся методом динамического программирования. Пусть $MI[S]$ — максимальное независимое подмножество S . Несложно заметить, что

$$MI[S] = \begin{cases} -\infty & \text{если } S = \emptyset \\ |S| & \text{если } S \text{ — независимое} \\ \max_{v \in S} MI[S \setminus \{v\}] & \text{в других случаях} \end{cases}$$

Данная формула описывает вычисление ответа для множества за время $O(n^{O(1)})$, через множества меньшего размера. Таким образом общее время работы динамического программирования есть $O(2^b \cdot n^{O(1)})$. И общее время решения задачи равно $O((3^{n-b} + 2^b) \cdot n^{O(1)})$

4.3. Решение за время $O((2^{n-b} + 2^b) \cdot n^{O(1)})$

Теперь улучшим перебор пар X . После вычисления $MI[S]$ задачу можно переформулировать следующим образом. Для каждого $A \subset X$ определён вес $w(A) = MI[Y \setminus N_G(A)]$. Необходимо найти разбиение X на три множества, так, чтобы $w(X_1) + w(X_2) \geq |X_3|$. Все остальные условия структурной теоремы получаются выполненными автоматически!

Научимся решать такую задачу быстрее, чем за время $O(3^{n-b} \cdot n^{O(1)})$.

Занумеруем вершины X от 0 до $n - b - 1$ в любом порядке. Далее вершина отождествляется с ее номером.

Для $A \subset X$ определим $bit(A) = \sum_{k \in A} 2^k$. Заметим, что $bit(A) + bit(B) = bit(A \sqcup B)$, если A и B не пересекаются.

Для $A \subset X$ и неотрицательного целого числа k определим $D(A, k) = bit(A) + 2^{|X|} \cdot k + 2^{|X| + \lceil \log 2|Y| \rceil} |A|$.

Лемма 4. Если $A \cap B = \emptyset$ и $k_1 + k_2 \leq 2|Y|$, то $D(A, k_1) + D(B, k_2) = D(A \sqcup B, k_1 + k_2)$.

Если $A \cap B \neq \emptyset$, то $D(A, k_1) + D(B, k_2) \neq D(C, k_3)$ ни для каких C, k_3 .

Доказательство. Первый пункт очевиден.

Для доказательства второго заметим, что остаток от деления $D(A, k_1) + D(B, k_2)$ на $2^{|X|}$, однозначно задает C .

С другой стороны, частное от деления на $2^{|X| + \lceil \log 2|Y| \rceil}$ однозначно задает размер C . Покажем, что, в случае пересечения множеств A и B , эти данные не могут быть согласованы.

$\left\lceil \frac{D(A, k_1) + D(B, k_2)}{2^{|X| + \lceil \log 2|Y| \rceil}} \right\rceil \geq |A| + |B|$. С другой стороны $|C| < |A| + |B|$. Для доказательства этого факта разложим все эти числа в сумму степеней двойки. Количество этих степеней — ровно размер соответствующего множества. При сложении, если были одинаковые степени, то они склеятся в одну, возможно несколько раз. При этом увеличиться их количество никак не может, что доказывает, что $D(A, k_1) + D(B, k_2)$ не представимо в виде $D(C, k_3)$, если A и B пересекаются. □

Запишем многочлен $P(z) = \sum_{\substack{A \subset X \\ w(A) \neq -\infty}} z^{D(A, w(A))}$ и рассмотрим многочлен $P^2(z)$. Заметим, что все слагаемые среди попарных произведений имеют положительный коэффициент, а значит они не могут взаимно уничтожиться. Пусть X_1 и X_2 искомые множества. Тогда в многочлене $P^2(z)$ есть слагаемое $z^{D(X_1, w(X_1))} \cdot z^{D(X_2, w(X_2))} = z^{D(X_1 \sqcup X_2, w(X_1) + w(X_2))}$, то есть слагаемое вида $z^{D(A, w)}$, такое, что $w \leq |X \setminus A|$. С другой стороны, пусть есть такое слагаемое. Тогда рассмотрим, как произведение каких слагаемых многочлена P оно было получено. По лемме 4, оно могло быть получено только из слагаемых, соответствующих не пересекающимся множествам. Тогда они подходят как X_1 и X_2 .

Многочлен $P(z)$ имеет степень $2^{|X|+\lceil \log 2|Y| \rceil} \cdot |X| = O(2^{|X|} \cdot n^{O(1)})$. С помощью дискретного преобразования Фурье этот многочлен можно возвести в квадрат за время $O(2^{|X|} \cdot n^{O(1)})$, после чего за такое же время проверить существование нужного слагаемого. Таким образом, общее время работы алгоритма $O((2^{n-b} + 2^b) \cdot n^{O(1)})$.

4.4. Решение за время $O(2^{n-b+\beta b} \cdot n^{O(1)})$

Заметим, что если b близко к n , то самым большим слагаемым становится время работы динамического программирования. Он него можно отказаться в пользу вычисления веса с помощью алгоритма поиска независимого множества в соответствующем графе каждый раз.

Это можно сделать за время $O(2^{\beta b} \cdot n^{O(1)})$, тогда время работы алгоритма составит $O(2^{n-b+\beta b} \cdot n^{O(1)})$, что лучше, чем 2^b при b , близких к n .

4.5. Комбинирование решений

Теорема 4. *Задача проверки существования плана перевозки на графе размера n с лодкой размера b может быть решена за время $O(\min(2^{(1+\alpha)b}, 2^{n-b+\beta b}, 2^{n-b} + 2^b) \cdot n^{O(1)})$*

Получим оценку времени работы, зависящую только от n . Для этого необходимо найти максимум функции $\min((1+\alpha)\frac{b}{n}, 1 - \frac{b}{n} + \beta\frac{b}{n}, \max(\frac{b}{n}, 1 - \frac{b}{n}))$.

Он может достигаться только при равенстве одной из первых двух функций третьей, то есть в точках

- $\frac{b}{n} = \frac{1}{2+\alpha}$
- $\frac{b}{n} = \frac{1}{2-\beta}$

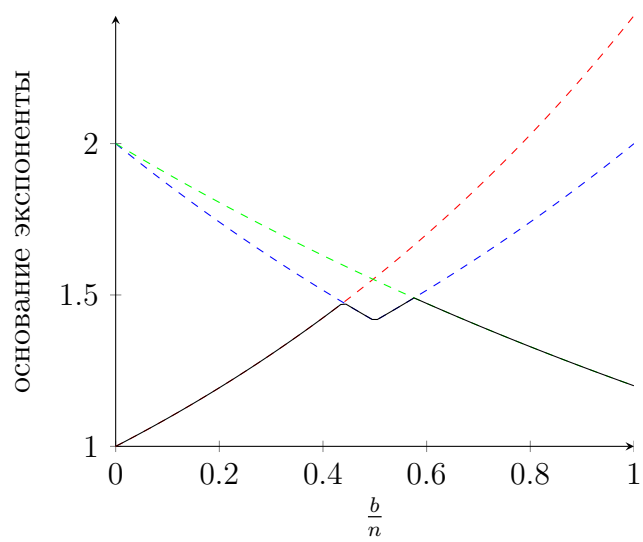
Таким образом, верна следующая теорема.

Теорема 5. *Задача проверки существования плана перевозки на графе размера n может быть решена за время $O(2^{\max(\frac{1+\alpha}{2+\alpha}, \frac{1}{2-\beta})n} \cdot n^{O(1)})$*

В частности, используя известные оценки на α и β , получаем следующую теорему

Теорема 6. *Задача проверки существования плана перевозки на графе размера n может быть решена за время $O(2^{\max(0.5608, 0.5758)n} \cdot n^{O(1)}) = O(2^{0.5758n} \cdot n^{O(1)}) = O(1.4904^n \cdot n^{O(1)})$*

На графике ниже изображена зависимость основания экспоненты по n от отношения $\frac{b}{n}$



5. Заключение

Таким образом, было получено существенное улучшение верхних оценок на время поиска числа Алкуина, по сравнению с [2].

Остаются открытыми вопросы о нижних оценках на время работы, а также о трудности проверки равенства числа Алкуина и вершинного покрытия графа.

Список литературы

- [1] Chen Jianer, Kanj Iyad A., Xia Ge. Improved upper bounds for vertex cover // Theor. Comput. Sci. — 2010. — Vol. 411, no. 40-42. — P. 3736–3756.
- [2] Csorba Peter, Hurkens Cor A. J., Woeginger Gerhard J. The Alcuin Number of a Graph and Its Connections to the Vertex Cover Number // SIAM Journal on Discrete Mathematics. — 2010. — Vol. 24, no. 3. — P. 757–769.
- [3] Xiao M, Nagamochi H. Exact algorithms for maximum independent set // Algorithms and Computation. — 2013. — Vol. Springer.