# Справедливые раскраски гиперграфов в $r$ цветов

М. Ахмеджанова[*], Д.А. Шабанов[†]

**Аннотация**

В нашей работе получена оценка на число ребер $n$-однородного гиперграфа, которая обеспечивает существование не только правильной раскраски в $r$ цветов, но и справедливой. Раскраска множества вершин гиперграфа называется справедливой (в мировой литературе используется термин equitable), если она является правильной (нет одноцветных ребер) и при этом мощности всех цветовых классов отличаются не более чем на единицу. Данный результат усиливает ранее известную теорему Косточки.

## 1 Введение

В работе рассматривается известная задача экстремальной комбинаторики, связанная с раскрасками гиперграфов.

Дадим основные определения из теории гиперграфов. Гиперграфом называется пара $H = (V, E)$, где $V = V(H)$ — некоторое множество, называемое множеством вершин гиперграфа, а $E = E(H)$ — произвольная совокупность подмножеств множества $V$, называемых ребрами гиперграфа. Гиперграф является $n$-однородным, если каждое его ребро содержит ровно $n$ вершин.

Раскраска множества вершин $V$ гиперграфа $H = (V, E)$ называется правильной, если в этой раскраске все ребра из $E$ не являются одноцветными. Если для гиперграфа $H$ существует правильная раскраска в $r$ цветов, то говорят, что $H$ является $r$-раскрашиваемым.

А.Косточка [1] доказал следующую теорему: если $r < \sqrt{\frac{1}{8}\ln\frac{ln}{2}}$ и число ребер $n$-однородного гиперграфа $H$ не превосходит

$$|E| \leq e^{-4r^2}\left(\frac{n}{\ln(n)}\right)^{\lfloor log_2(r)/\log_2(r)+1\rfloor} r^n,$$

то $H$ является $r$-раскрашиваемым.

[*]Московский государственный университет имени М.В.Ломоносова, механико-математический факультет; Московский физико-технический институт, лаборатория продвинутой комбинаторики и сетевых приложений

[†]Московский физико-технический институт, лаборатория продвинутой комбинаторики и сетевых приложений; Математический институт им. В.А. Стеклова Российской академии наук

Правильные раскраски допускают различные обобщения, одним из которых являются справедливые раскраски. Раскраска множества вершин гиперграфа называется *справедливой* (в мировой литературе используется термин *equitable*), если она является правильной и при этом мощности всех цветовых классов отличаются не более, чем на единицу.

Легко показать, что тривиальная оценка $|E| < r^{n-1}$ обеспечивает возможность справедливой раскраски $H$ в $r$ цветов. Однако, в отличии от правильных раскрасок другие оценки на число ребер авторам неизвестны.

Наш результат сформулирован в следующей теореме:

**Теорема 1.** *Пусть $H = (V, E)$ произвольный $n$-однородный гиперграф с условием*

$$|E| \leq 0.05 \left(\frac{n}{\ln n}\right)^{\lfloor \log_2(r)/\log_2(r)+1 \rfloor} r^{n-1},$$

*где число цветов $r \leq \sqrt{\frac{\ln n}{-64(\ln c)(\ln \ln n)^3}}$. Тогда для $H$ существует не только правильная, но справедливая раскраска в $r$ цветов.*

Таким образом, любой $n$-однородный гиперграф с ограниченным числом ребер допускает не только правильную раскраску в $r$ цветов, но и справедливую. Эта теорема существенно усиливает ранее известный результат.

Возможность справедливо раскрасить можно исследовать не только при ограничении на число ребер, но и при условии, когда ограничены степени вершин гиперграфа. Так, в 1970 году А. Хайнал и Е. Семереди [3] доказали знаменитую гипотезу П. Эрдеша [2]: *любой граф $G$ с максимальной степенью вершины $\Delta(G)$ допускает не только правильную, но и справедливую раскраску в $\Delta(G) + 1$ цветов.* Позже, доказательство этого фундаментального факта было значительно упрощено Х. Киерстедом и А.В. Косточкой [4], они же совместно с М. Мидларжем и Е. Семереди отыскали [5] быстрый алгоритм для получения искомой справедливой раскраски.

В случае $n$-однородных гиперграфов $\mathcal{H}$ Шабанов [6] показал, что для всякого простого гиперграфа( любые два его ребра имеют не более одной общей вершины) $H \in \mathcal{H}$, а также для всякого гиперграфа $H \in \mathcal{H}$ с большим количеством вершин условие на максимальную степень вершины $\Delta(H)$ :

$$\Delta(H) \leq c \cdot 2^{n-1}/\sqrt{n \ln n}$$

обеспечивает возможность справедливой раскраски $H$ в два цвета. Позже, этот результат был улучшен И. Акользиным [7], который доказал, что для больших $n$ верна оценка:

$$\Delta(H) \leq c \cdot 2^{n-1}$$

# 2 Построение справедливой раскраски: алгоритмы перекраски

Пусть $H = (V, E)$ — гиперграф из условия теоремы 1. Для удобства будем считать, что число вершин $m$ кратно $r$. Под сбалансированной раскраской $C = C(K_1, K_2, ..., K_r)$ будем понимать некоторое разбиение множества вершин гиперграфа на $r$ равных долей: $V = K_1 \sqcup K_2 ... \sqcup K_r, |K_1| = |K_2| = ... = |K_r|$.

**Лемма 1.** *Пусть $H = (V, E)$ произвольный $n$-однородный гиперграф, в котором выполнены следующие соотношения:*

$$|E| < c \left(\frac{n}{\ln(n)}\right)^{\lfloor \log_2(r)/\log_2(r)+1 \rfloor} r^{n-1}, \qquad |V| < \frac{n(n-1)(r-1)\lfloor \frac{log_2(r)+1}{\log_2(r)} \rfloor}{2\ln(n)}.$$

*Тогда при $c < 1$ для $H$ существует справедливая раскраска в $r$ цветов.*

*Доказательство.* Рассмотрим случайную сбалансированную раскраску $C$ гиперграфа $H$.

Тогда,

$$\mathsf{P}(\text{есть одноцветное ребро в } C) \leq \sum_{e \in E} \frac{r\binom{m-n}{m/r-n} \cdot \binom{m-m/r}{m/r} \cdot \ldots \cdot \binom{m/r}{m/r}}{\binom{m}{m/r} \cdot \binom{m-m/r}{m/r} \cdot \ldots \cdot \binom{m/r}{m/r}} = |E| \frac{r\binom{m-n}{m/r-n}}{\binom{m}{m/r}} =$$

$$= |E| \frac{r^{-n+1}(1-r/m)...(1-r(n-1)/m)}{(1-1/m)...(1-(n-1)/m)} \leq$$

$$\leq c \left(\frac{n}{\ln(n)}\right)^{\lfloor \frac{\log_2(r)}{\log_2(r)+1} \rfloor} \frac{\exp(\ln(1-r/m) + ... + \ln(1-r(n-1)/m))}{\exp(\ln(1-1/m) + ... + \ln(1-(n-1)/m))} =$$

$$= c \left(\frac{n}{\ln(n)}\right)^{\lfloor \frac{log_2(r)}{\log_2(r)+1} \rfloor} \prod_{x=1}^{n-1} \exp\left(\ln(1-rx/m) - \ln(1-x/m)\right)$$

Из разложения натурального логарифма в ряд Тейлора следует, что $\ln(1-rx/m) - \ln(1-x/m) < -x(r-1)/m$ при $x \in (0; 1)$. Поэтому, складывая показатели степеней у произведения экспонент, окончательно получаем:

$$\mathsf{P}(\text{есть одноцветное ребро в } C) \leq c \left(\frac{n}{\ln(n)}\right)^{\lfloor \frac{log_2(r)}{\log_2(r)+1} \rfloor} e^{-n(n-1)(r-1)/2m} \leq \frac{c}{n^{\log_2(r)/\log_2(r)+1}} < 1$$

Следовательно, с положительной вероятностью случайная сбалансированная раскраска $C$ является справедливой раскраской. Лемма 1 доказана. □

Таким образом, нам остается рассмотреть гиперграфы, в которых число вершин $m$ больше, чем $n(n-1)(r-1)\lfloor \frac{log_2(r)+1}{\log_2(r)} \rfloor / 2\ln n$.

## 2.1 Алгоритм 1: построение правильной раскраски

Следующим шагом в доказательстве Теоремы 1 является применение модифицированной версии алгоритма перекраски А. Косточки. Алгоритм А. Косточки описан в [1], а наша модификация состоит в том, что, значение случайной величины, которая отвечает за то, что вершина из одноцветного ребра будет перекрашена, фиксируется вначале и не меняется в процессе работы алгоритма.

Наша цель — построить из случайной раскраски $C^0$ некоторую новую случайную раскраску и показать, что она является правильной с положительной вероятностью. Для построения подобной раскраски зададим случайный порядок на множестве вершин гиперграфа $V$ с помощью отображения $\sigma : V \to [0,1]$, где $\sigma(v), v \in V$ — независимые случайные величины с равномерным распределением на $[0,1]$. Если $\sigma(v) < p$, где $p$ — некоторый параметр нашей конструкции, то вершину $v$ будем называть свободной. Пусть $x = \lfloor \log_2 r \rfloor$.

Предположим, что в раскраске $C^0$ есть одноцветные ребра. Для исправления ситуации мы используем следующий алгоритм перекраски:

Этап $l, 0 < l \leq x$. Для каждой вершины $v$ проверяем выполнимость двух условий:

1. она никогда не меняла свой цвет и является свободной,
2. в текущий раскраске имеется инцидентное $v$ одноцветное ребро $A, v \in A$ некоторого цвета $\alpha \in \{1, \dots, r\}$ , и никакая из его вершин не была перекрашена на этапе $l$.

Если выполнены оба условия, перекрашиваем вершину $v$ в цвет $(\alpha + 2^{l-1}) (\bmod\ r)$. В этом случае будем говорить, что перекрашенная вершина $v$ обвиняет ребро $A$.

Если какое-то из условий 1.-2. не выполняется, то переходим к следующей вершине. Если мы перебрали все вершины, то переходим на этап $(l+1)$.

Отметим, что в рамках процедуры перекраски каждая вершина меняет цвет не более одного раза, поэтому алгоритм не может работать бесконечно и обязательно остановится.

## 2.2 Алгоритм 2: восстановления равенства мощностей цветовых классов

Мощности цветовых классов могут отличаться от $m/r$. Определим для каждого цветового класса $K_i$ неотрицательные целые числа $ex_i$ и $sh_i$, где $ex_i$ — величина избытка, т.е. на сколько вершин цвета $r_i$ больше чем $m/r$, аналогично для недостатка. Ясно что, если $ex_i > 0$, то $sh_i = 0$, и наоборот.

Выберем случайно в каждом цветовом классе $K_i$ подмножество вершин $V_i$, размера $q$. Значение параметра $q$ определим позже. А пока предположим, что $q > 2ex_i$ для любого $i$.

**Data:** набор чисел $ex_i$ и $sh_i, i = 1, 2, ..., r$.
**for** $(i = 1; i < r; i++)$ **do**
  **if** $(sh_i! = 0)$ **then**
    {берем минимальное $t : ex_1 + ex_2 + ... + ex_t \geq sh_i$
    В каждом множестве $V_s, s < t$ перекрашиваем в цвет $i$ первые $ex_s$ вершин.
    А в последнем множестве $V_t$ перекрашиваем в цвет $i$ первые
    $sh_t - (ex_1 + ex_2 + ... + ex_{t-1})$ вершин.
  $ex_1 = 0, ex_2 = 0, ..., ex_t = ex_t - (sh_t - (ex_1 + ex_2 + ... + ex_{t-1}))$.}

**Комментарии:**

- Шаг 1. Пусть цвет $i$ — это первый цвет, который в недостатке. Берем минимальное $t : ex_1 + ex_2 + ... + ex_t \geq sh_i$. В каждом множестве $V_s, s < t$ перекрашиваем в цвет $i$ первые $ex_s$ вершин. А в последнем множестве $V_t$ перекрашиваем в цвет $i$ первые $sh_t - (ex_1 + ex_2 + ... + ex_{t-1})$ вершин. После перекраски мощности цветовых классов $K_1, ..., K_{t-1}, K_i$ становятся равными $m/r$. Теперь цвета $1, ..., t-1$ больше не нужны и мы кладем $ex_1 = 0, ex_2 = 0, ..., ex_t = ex_t - (sh_t - (ex_1 + ex_2 + ... + ex_{t-1}))$. На этом шаг 1 заканчивается.

- Шаг 2. Берем следующий цвет, который в недостатке. Проделываем тоже самое, что и в шаге 1. Множества из которых мы теперь набираем вершины отличаются от $V_1, ..., V_{t-1}$, за исключением множества $V_t$.

Ясно, что Алгоритм 2 сделает все цветовые классы равные $m/r$. Если при этом, мы сможем выбирать вершины для перекраски таким образом, чтобы не появлялись новые одноцветные ребра, то мы получим справедливую раскраску.

# 3 $T$-деревья

Обозначим через $C^1$ раскраску, которая получилась после работы Алгоритма 1, а через $C^2$ — финальную раскраску, т.е. раскраску после работы Алгоритма 2. Нас интересует количество одноцветных ребер, которые получились в раскрасках $C^1$ и $C^2$.

Начнем с одноцветных ребер в раскраске $C^1$. Обозначим через $A$ одно из подобных одноцветных ребер некоторого цвета $\alpha$. Ясно, что в силу алгоритма перекраски каждая вершина $v$ ребра $A$ в раскраске $C^0$ могла быть только следующих цветов: $\alpha, \alpha - 2^0, \alpha - 2^2, ..., \alpha - 2^{k-1} (\mathrm{mod}\ r)$. Предположим, что $\{i_1^A, i_2^A, ..., i_s^A\}$ — это цвета, которые были у вершин ребра $A$ в $C^0$. Определим через $a_{i_1^A}, a_{i_2^A}, ..., a_{i_s^A}$ набор вершин ребра $A$, где каждая вершина $a_{i_k^A}$ обладает самым большим весом среди всех вершин цвета $i_k^A$ ребра $A$. Для сокращения записи, будем называть такой набор вершин доминантным набором — $Dominant(A)$ ребра $A$, а вершины набора доминантными.

Начнем построение конфигурации ребер $H$, которую мы назовем *$T$-деревом* с корнем $A$.

- В качестве корня мы возьмем ребро $A$, а его потомками будут все ребра $B$, которые обвиняют доминантные вершины $a_{i_1^A}, a_{i_2^A}, ..., a_{i_s^A}$ (по одному ребру для каждой вершины).

- Далее, в каждом ребре $B$ есть свой набор цветов $\{i_1^B, i_2^A, ..., i_t^B\}$ и соответствующий ему доминантный набор вершин $b_{i_1^B}, b_{i_2^B}, ..., b_{i_t^B}$, которые были перекрашены до того, как ребро $B$ стало одноцветным. Подобные вершины обязаны обвинять некоторые новые ребра $C$, иначе ребро $B$ не могло бы стать одноцветным и, в свою очередь, никто не смог бы его обвинить. Добавим подобные ребра $C$ в качестве потомков ребра $B$

- Продолжим процесс построения, пока будет возможно.

В результате мы получим $T$-дерево, вершинами которого выступают ребра гиперграфа $H$, а связь определяется отношениями обвинения. Листьями получившегося $T$-дерева будут как раз те ребра гиперграфа $H$, которые являлись одноцветными в исходной сбалансированной случайной раскраске $C^0$. Корнем $T$-дерева является одноцветное ребро в раскраске $C^1$.

Введем понятие $T$-поддерева с корнем $A$. Для этого рассмотрим вершину $A \in T$. Пусть $N(A)$ множество всех предков $A$ в $T$-дереве. Тогда $A$ вместе с $N(A)$ будем называть *$T$-поддеревом*. Корнем $T$-поддерева является ребро, которое было одноцветным на некотором этапе работы Алгоритма 1. Соответственно, под *высотой* $h(A)$ $T$-поддерева с корнем $A$ будем понимать наибольшую длину пути из корня $A$ в лист.

**Следствие 1.** *Пусть заданы $\sigma$ и случайная раскраска $C^0$. Предположим, что Алгоритм 1 не сработал, т.е. не построил правильную раскраску, тогда появилось хотя бы одно $T$-дерево.*

## 3.1 $T$-сложные деревья

Вернемся к рассмотрению одноцветных ребер в раскраске $C^2$. Зафиксируем некоторый цвет $i$ и рассмотрим все одноцветные ребра цвета $i$ в раскраске $C^2$. Каждое такое ребро $A$ может содержать вершины только двух типов: каждая $v \in A$

- либо была цвета $i$ уже после завершения работы Алгоритма 1.
- либо получила цвет $i$ во время работы Алгоритма 2. Обозначим через $U(A)$ множество вершин ребра $A$, имеющих тип 2.

Рассмотрим вершины $w$, которые имеют цвет $i$ уже после Алгоритма 1. Как и в предыдущем случае среди этих вершин $w$ мы можем выбрать доминантный набор вершин $Diominant(A)$, а для этого доминантного набора набор ребер $B_1, ..., B_s$, которые обвиняют доминантные вершины.

Начнем построение конфигурации ребер $H$, которую мы назовем *$T$-сложным деревом.* Зафиксируем ребро $A$ и неупорядоченный набор ребер $B_1, ..., B_s$. Теперь в качестве корня мы возьмем ребро $A$, а его потомками будут все ребра $B_1, ..., B_s$. После этого мы строим уже обычные $T$–поддеревья: $T(B_1), ..., T(B_s)$. На этом процесс построения заканчивается. Обозначим получившееся $T$–сложное дерево через $T(A, B_1, ..., B_s)$.

Рассмотрим теперь в ребре $A$ те вершины $u$, которые получили цвет $i$ во время работы Алгоритма 2. Если $u \in A \cap T(B_t)$, то изначальный цвет такой вершины уже известен. Поэтому рассмотрим те вершины $u$, которые не попадают в такие пересечения. Согласно Алгоритму 2 каждая такая вершина принадлежит некоторому подмножеству $V_i$ и имеет в $C^0$ цвет $i$.

## 3.2 Вспомогательные утверждения о $T$-деревьях

**Лемма 2.** *$T$-дерево — это граф без циклов, в котором для любых двух вершин $x_1$ и $x_2$*

- *либо $e(x_1) \cap e(x_2) = \emptyset$,*
- *либо $|e(x_1) \cap e(x_2)| = 1$ и $x_1$, $x_2$ смежные вершины $T$-дерева.*

*Доказательство.* Докажем сначала, что в графе $T$ нет циклов. Предположим обратное: пусть $f_1, f_2, ..., f_k, f_1$ — цикл. Пусть $f_1$ начало цикла и главный цвет ребра $e(f_1)$ равен $\alpha$. Для того чтобы цикл замкнулся на вершине $f_1$ ребро $e(f_1)$ должно снова стать одноцветным. Посчитаем цвет, который должен получиться у ребра $e(f_1)$: обозначим через $h(v)$ высоту поддерева с корнем $v$, тогда главные цвета ребер $f_1, f_2, ..., f_k, f_{k+1} = f_1$ будут равны соответственно $\alpha, (\alpha + 2^{h(f_2)-1}) (\text{mod } r), ..., (\alpha + 2^{h(f_2)-1} + ... + 2^{h(f_{k+1})-1}) (\text{mod } r)$.

С другой стороны, в ребре $e(f_1)$ есть перекрашенная вершина, которая согласно Алгоритму 1 не может уже поменять свой цвет — $(\alpha + 2^{h(f_2)-1}) (\text{mod } r)$. Числа $(\alpha + 2^{h(f_2)-1}) (\text{mod } r)$ и $(\alpha + 2^{h(f_2)-1} + ... + 2^{h(f_{k+1})-1}) (\text{mod } r) = (\alpha + 2^{h(f_2)-1} \cdot (2^{k+1} - 1)) (\text{mod } r)$ не могут совпасть, поскольку тогда должно быть выполнено следующее сравнение по модулю $r$:

$$
\begin{gathered}
2^{h(f_2)-1} \equiv 2^{h(f_2)-1} \cdot (2^{k+1} - 1) \pmod r \\
2^{h(f_2)} \equiv 2^{h(f_2)+k} \pmod r
\end{gathered}
$$

Последнее не может быть выполнено, поскольку $h(f_2) + k < \lfloor \log_2 r \rfloor$. В итоге мы получили противоречие, что $f_1, f_2, ..., f_k, f_1$ цикл.

Докажем второе утверждение. Пусть сначала $x_1$ и $x_2$ смежные вершины. Тогда если в пересечение $e(x_1) \cap e(x_2)$ больше одной вершины, то получается, что в $C^0$ есть две доминантные вершины одного цвета, чего не может быть. Пусть теперь $|e(x_1) \cap e(x_2)| \geq 1$ для двух несмежных вершин. Тогда, есть два ребра $f_i$ и $f_j$ (мы допускаем возможность, что $f_i = f_j$), у которых совпадают главные цвета. Пусть их главный цвет равен $\beta$. Рассмотрим два пути:

$$
\begin{gathered}
P_1 = (C_0 = root, C_1, ..., C_t = f_i, ..., C_k) \\
P_2 = (C_0 = root, \tilde{C}_1, ..., \tilde{C}_s = f_i, ..., \tilde{C}_l),
\end{gathered}
$$

где $C_k$ и $\tilde{C}_l$ — листья. Согласно Алгоритму 1, главный цвет ребра $e(C_0)$ равен $(\beta + 2^{k-t} + 2^{k-t+1} + ... 2^{k-1}) (\text{mod } r) = (\beta + 2^{k-t} \cdot (2^t - 1)) (\text{mod } r)$ и $(\beta + 2^{l-s} + 2^{l-s+1} + ... + 2^{l-1}) (\text{mod } r) = (\beta + 2^{l-s} \cdot (2^s - 1)) (\text{mod } r)$ одновременно. Ввиду того, что числа $2^t - 1$ и $2^s - 1$ нечетные при $s, t > 1$, сравнение выполнено только при $s = t$ и $l = k$. Но это противоречит Алгоритму 1, так как у корня $T$-дерева будут две доминантные вершины одного цвета. □

**Следствие 2.** *Если в $T$-дереве известен итоговый цвет $\alpha$ корня $A$, то изначальные цвета всех доминантных вершин гиперграфа $H$, входящих в $T$-дерево, однозначно определяются по структуре графа $T$.*

**Лемма 3.** *Из ребер гиперграфа $H$ можно составить не более, чем $m \cdot \binom{|E|}{m}$ $T$–деревьев размера $m$.*

*Доказательство.* Возьмем произвольный неупорядоченный набор из $m$ ребер гиперграфа $H$. Выберем среди одно ребро, которое будет соответствовать корню $A$. Обозначим его через $A$. Среди оставшихся ребер рассмотрим все такие ребра — $B_1, ..., B_s$, которые пересекаются с ребром $A$. Добавим их в качестве предков $A$. Далее, среди оставшихся ребер ищем те, которые смежны с ребрами $B_1, ..., B_s$, они будут потомками вершин $B_1, ..., B_s$. Продолжая действовать таким образом мы в итоге получим граф $G$. Если он удовлетворяет условию леммы 1 и в нем $m$ вершин, то мы построили $T$-дерево, иначе делаем вывод, что из данного набора ребер нельзя составить $T$-дерево. Таким образом, число $T$-деревьев не превосходит $m \cdot \binom{|E|}{m}$. □

**Лемма 4.** *Из ребер гиперграфа $H$ можно составить не более, чем $\binom{|E|}{m} m 2^{m-1}$ $T$-сложных деревьев размера $m$.*

*Доказательство.* Зафиксируем число $s$ — число предков первого порядка у вершины $A$. Возьмем произвольный неупорядоченный набор из $m$ ребер гиперграфа $H$. Выберем среди них ребро, которое будет соответствовать корню $A$, среди оставшихся ребер выберем ребра, которые будут соответствовать предкам $B_1, ..., B_s$. Это можно сделать не более, чем $m\binom{m-1}{s}$ способами. Далее, для каждого $T$-поддерева $T(B_k), k = 1, 2, ..., s$ будем набирать ребра так как мы делали в лемме 3. Таким образом, общее число $T$-сложных деревьев не превосходит $\binom{|E|}{m} m \sum_{s=0} \binom{m-1}{s}$. □

# 4 Оценка числа одноцветных ребер, образовавшихся после работы Алгоритма 1

Во всех дальнейших вычислениях мы будем предполагать, что есть некоторый $n$-однородный гиперграф $H$, в котором $|E| \leq c \left(\frac{n}{\ln n}\right)^{\lfloor \log_2(r)/\log_2(r)+1 \rfloor} r^{n-1}$, а число цветов $r \leq \sqrt{\frac{\ln n}{\tilde{c}(\ln \ln n)^3}}$.

**Лемма 5.** *Пусть $\sigma$ и $C^0$ заданы случайно, тогда с положительной вероятностью Алгоритм 1 построит правильную раскраску $C^1$ в $r$ цветов.*

*Доказательство.* Рассмотрим событие заключающееся в том, что Алгоритма 1 не сработал, когда $\sigma$ и $C^0$ были заданы случайно. Согласно следствию 1, это событие могло произойти только тогда, когда появилось хотя бы одно $T$-дерево. Мы переберем все возможные $T$-деревья, оценивая для каждого $T$-дерева вероятность события, заключающегося в том, что в результате процесса случайной перекраски нашелся набор ребре гиперграфа $H$, который образовал это $T$-дерево.

Итак, пусть мы зафиксировали некоторое $T$-дерево с корнем $A$. Пусть $\alpha$ — цвет ребра $A$ в раскраске $C^1$. Предположим, что доминантный набор вершин ребра $A$ состоит из $s$ вершин, которые обвиняют ребра $B_1, ..., B_s$. Согласно следствию 2 изначальные цвета этих $s$ вершин однозначно определяются по структуре графа $T$. Обозначим через $x_1, ..., x_s$ соответствующие веса вершин доминантного набора ребра $A$. Теперь, согласно Алгоритму 1 цвет любой $v \in A : v \notin \mathcal{D}(\mathcal{A})$

- либо равен $\alpha$

- либо равен цвету одной из доминатных вершин $\{a_{i_j}^A : \sigma(v) \leq \sigma(a_{i_j}^A)\}$

Рассмотрим теперь ребро $B_1$. Пусть у $B_1$ ровно $t$ доминантных вершин, а их веса равны $y_1, ..., y_t$. Тогда цвет любой $v \in B_1 : v \notin \mathcal{D}(B_1)$ равен либо главному цвету ребра $B_1$, либо равен цвету одной из доминантных вершин $\{b_{i_j}^{B_1} : \sigma(v) \leq \sigma(b_{i_j}^{B_1})\}$.Повторяя рассуждения для всех вершин $T$-дерева, мы ограничим множество цветов, которые могут быть у вершин гиперграфа $H$.

Тем не менее вероятность появления $T$-дерева сильно зависит от номера этапа $l$ на котором завершилось образование $T$-дерева. Поэтому мы рассмотрим два случая и проведем вычисления для каждого случая отдельно.

**1.** Образовалось дерево за $l < x = \lfloor \log_2 r \rfloor$ этапов.

Обозначим данное событие через $\mathcal{A}$. В этом случае все вершины $v \in A$ цвета $\alpha$ имееют вес больше $p$. Пользуясь леммой 3 о числе $T$-деревьев мы можем оценить вероятность событие $\mathcal{A}$ следующим образом:

$$\mathsf{P}(\mathcal{A}) \leq r \sum_{m=1} m \cdot |E|^m \underbrace{\iint ... \int}_{[0;p]^{m-1}} \left(\frac{x_1}{r} + ... \frac{x_s}{r} + \frac{1-p}{r}\right)^{n-s} \cdot$$
$$\cdot \left(\frac{y_1}{r} + ... \frac{y_t}{r} + \frac{1-x_1}{r}\right)^{n-t-1} \cdot ... \cdot \left(\frac{1}{r}\right)^{m-1}$$

Оценим числитель подынтегрального выражения:

$$(x_1 + ...x_s + 1 - p)^{n-s} \cdot (y_1 + ... + y_t + 1 - x_1)^{n-t-1} \cdot ... \cdot \leq$$
$$\leq [(x_1 + ...x_s + 1 - p)(y_1 + ... + y_t + 1 - x_1) \cdot ... \cdot]^n \cdot (1-p)^{-s} \cdot (1-p)^{-t-1} \cdot ... \cdot \leq$$
$$\leq \left(1 - \frac{p}{m}\right)^{mn} \cdot (1-p)^{-2(m-1)}$$

С учетом полученной оценки на числитель,

$$\mathsf{P}(\mathcal{A}) \leq r \sum_{m=1} m \cdot |E|^m p^{m-1} \left(\frac{1}{r}\right)^{(n-1)m+1} \left(1 - \frac{p}{m}\right)^{mn} \cdot (1-p)^{-2(m-1)} \leq$$
$$\leq \sum_{m=1} \left[\left(\frac{2cn}{\ln n}\right)^{\lfloor \frac{\log_2(r)}{\log_2(r)+1} \rfloor}\right]^m p^{m-1} \left(1 - \frac{p}{m}\right)^{mn} \cdot (1-p)^{-2(m-1)} \leq$$
$$|\ p = \gamma \frac{\ln n}{n}|$$
$$\leq 2c \sum_{m=1} \left(\frac{n}{\ln n}\right)^{\lfloor \frac{\log_2(r)}{log_2(r)+1} \rfloor} \left(\frac{\ln n}{n}\right)^{\frac{m-1}{\log_2(r)+1}} \frac{1}{n^\gamma} \leq \frac{2cn^{1-\gamma}}{\ln n} \sum_{m=1} \exp\left(-\frac{\ln \frac{n}{\ln n}}{\log_2(r)+1}(m-1)\right)$$

Ряд $\sum_{m=1} \exp(-\alpha m)$ сходится при $\alpha \geq 1$. Таким образом, когда число цветов невелико и $\gamma \geq 1$, вероятность события $\mathcal{A}$ будет стремится к нулю при $n \to \infty$, при этом $\mathsf{P}(\mathcal{A})$ заведомо меньше, чем $4c$.

**2.** Образовалось дерево на последнем этапе Алгоритма 1. $l = x = \lfloor \log_2 r \rfloor$.

Обозначим данное событие через $\mathcal{B}$. В этом случае мы не можем ничего сказать о вершинах $v \in A$ цвета $\alpha$. Однако, за счет того, что $T$-дерево построилось в конце, мы можем утверждать, что в нем много вершин, а именно $m \geq \lfloor \log_2 r \rfloor$.

$$\mathsf{P}(\mathcal{B}) \leq r \sum_{m > \lfloor \log_2 r \rfloor} m \cdot |E|^m \underbrace{\iint \dots \int}_{[0;p]^{m-1}} \left( \frac{x_1}{r} + \dots \frac{x_s}{r} + \frac{1}{r} \right)^{n-s} \cdot$$
$$\cdot \left( \frac{y_1}{r} + \dots \frac{y_t}{r} + \frac{1 - x_1}{r} \right)^{n-t-1} \cdot \dots \cdot \left( \frac{1}{r} \right)^{m-1}$$

С учетом того, что числитель подынтегрального выражения не превосходит 1,

$$\mathsf{P}(\mathcal{B}) \leq r \sum_{m > \lfloor \log_2 r \rfloor} m \cdot |E|^m p^{m-1} \left( \frac{1}{r} \right)^{(n-1)m+1} \leq$$
$$\leq \sum_{m > \lfloor \log_2 r \rfloor} \left[ \left( \frac{2cn}{\ln n} \right)^{\lfloor \frac{\log_2(r)}{log_2(r)+1} \rfloor} \right]^m p^{m-1} \leq$$
$$\leq 2c \sum_{m > \lfloor \log_2 r \rfloor} \left( \frac{n}{\ln n} \right)^{\lfloor \frac{\log_2(r)}{log_2(r)+1} \rfloor} \left( \frac{\ln n}{n} \right)^{\frac{m-1}{\log_2(r)+1}} \leq 2c \sum_{m=1} \exp\left( - \frac{\ln \frac{\ln n}{n}}{\log_2(r) + 1} (m-1) \right) < 4c$$

при ограничении на число цветов.

$\square$

## 4.1 Оценка на максимальный избыток и максимальный недостаток

В этом параграфе мы получим явное соотношение на параметр $q$ из Алгоритма 2. Напомним, что мы хотим, чтобы $q \geq 2 \max_i ex_i$. Поэтому следующим действием будет оценка сверху величины $ex_i$.

Пусть $\sigma$ и $C^0$ заданы случайно. Обозначим через $Rec(\alpha)$ случайную величину равную числу вершин $v$, перекрашенных в цвет $\alpha$ в результате Алгоритма 1. Также, для каждого цветового класса $K_\alpha$ рассмотрим случайную величину $X_{K_\alpha}$ , равную число вершин цвета $\alpha$ в $C^0$. Ясно, что $X_{K_\alpha}$ имеет биномиальное распределение $Bin(m, 1/r)$.

Таким образом, суммируя вышесказанное, заметим, что для каждого цвета $\alpha$, который в избытке, выполнено, что

$$ex_\alpha \le Rec(\alpha) + \left(X_{K_\alpha} - \frac{m}{r}\right).$$

Используя неравенство Чернова, которое утверждает, что для случайной величины $X \sim Bin(n,p)$

$$\mathsf{P}(X > EX + t) \le \exp\left(-\frac{t^2}{2(EX + t/3)}\right)$$

с $t = \sqrt{\frac{2m \ln r/c}{r}}$, получаем оценки

$$\mathsf{P}\left(\exists \alpha : X_{K_\alpha} > \frac{m}{r} + t\right) \le r \exp\left(-\frac{t^2}{2(\frac{m}{r} + t/3)}\right) < \frac{cr}{r} = c$$

Получим теперь оценку на величину $Rec(\alpha)$. Для этого заметим, что каждой $v$, перекрашенной в цвет $\alpha$, соответствует некоторое $T$-поддерево — $T(A)$. Соответствие заключается в том, что вершина $v$ обвиняет ребро $A$. При этом, доминантный цвет ребра $A$ в $T$-поддереве $T(A)$ однозначно определяется по структуре графа $T$. Действительно, чтобы вершина $v$ перекрасилась в цвет $\alpha$ необходимо, чтобы доминантный цвет ребра $A$ был равен в точности $(\alpha - 2^{h(A)-1}) (\mathrm{mod}\ r)$, где $h(A)$ — высота дерева $T$. Таким образом, нам остается оценить число $T$-поддеревьев, у которых доминантный цвет корня заранее определен.

По аналогии с уже проделанными вычислениями для $T$-деревьев, математическое ожидание числа $T$-поддеревьев, а следовательно, и $\mathsf{E}\mathsf{Rec}(\alpha)$ можно оценить следующим образом:

$$\mathsf{E}Rec(\alpha) \le \sum_{m=1} p^{m-1} m \cdot \binom{|E|}{m} \left(\frac{1}{r}\right)^{(n-1)m+1}$$

Для сокращения записи обозначим через $k$ следующее выражение : $\left(\frac{n}{\ln(n)}\right)^{\lfloor \log_2(r)/\log_2(r)+1 \rfloor}$. Тогда, при $p = \gamma \frac{\ln n}{n}$ и $r < \left(\frac{n}{\ln n}\right)^{\ln 2}$.

$$\sum_{m=1} p^{m-1} m \cdot \binom{|E|}{m} \left(\frac{1}{r}\right)^{(n-1)m+1} \le \frac{2c}{r} \sum_{m=1} \left(\frac{n}{\ln n}\right)^{\lfloor \frac{\log_2(r)}{log_2(r)+1} \rfloor} \left(\frac{\ln n}{n}\right)^{\frac{m-1}{\log_2(r)+1}} \le \frac{4ck}{r}$$

Окончательно получаем, что с положительной вероятностью для каждого цвета $\alpha$

$$Rec(\alpha) \le \frac{k}{2}.$$

Положим параметр $q$ равным

$$q = \left(\frac{n}{\ln(n)}\right)^{\lfloor \log_2(r)/\log_2(r)+1 \rfloor} + 2\sqrt{\frac{2m\ln r/c}{r}}. \tag{1}$$

Таким образом, с положительной вероятностью

$$max_i ex_i \leq q/2. \tag{2}$$

А повторяя предыдущие рассуждения для $sh_i$, можно получить, что с положительной вероятностью

$$max_i sh_i \leq q/2. \tag{3}$$

# 5 Оценка числа одноцветных ребер, образовавшихся после работы Алгоритма 2

В предыдущей главе мы доказали, что с большой вероятностью после завершения работы Алгоритма 1 все ребра гиперграфа $H$ будут неодноцветными. Поэтому предположим, что все одноцветные ребра, которые мы имеем в финальной раскраске $C^2$, появились во время работы Алгоритма 2. В силу данного предположения, каждое такое одноцветное ребро имеет хотя бы одну вершину, которая поменяла свой цвет во время работы Алгоритма 2.

Нам удобно ввести следующее обозначение: Пусть $\sigma$ и $C^0$ заданы случайно. Обозначим через $X$ случайную величину равную числу одноцветных ребер в раскраске $C^2$.

**Лемма 6.** *Пусть $\tilde{p} = \frac{q}{m/r - q/2}$, где $q$ — параметр из Алгоритма 2. Тогда,*

$$\mathsf{E}X < ck(1 + \tilde{p}(r-1))^n.$$

*Доказательство.* Как и прежде, зафиксируем некоторое $T(A, T(B_1), ..., T(B_s))$ —

$T$-сложное дерево с корнем $A$ и потомками первого порядка: $B_1, ..., B_s, s \in \{1, ..., \log_2 r\}$. Обозначим через $t_i$ число вершин в пересечении $A \cap T(B_i)$, а через $m_i$ — размер $T$-поддерева $T(B_i)$. Пусть $\alpha$ цвет ребра $A$ в раскраске $C^2$.

Как и в случае $T$-деревьев, по структуре графа $T$ однозначно определяются цвета всех доминантных вершин в раскраске $C^0$. Заметим, что $T(B_1), ..., T(B_s)$ являются непересекающимися $T$-поддеревья, и поэтому для вершин гиперграфа $H$, попадающих в эти $T$-поддеревья выполняются все соотношения из Леммы 5.

Рассмотрим теперь вершины ребра $A$:

1. Если $u \in A \cap (\cup_t T(B_t))$, то изначальный цвет такой вершины уже известен.

2. Если $w \in A \setminus (\cup_t T(B_t))$, то она либо цвета $\alpha$ , либо она попала в одно из множеств $V_i, i = 1, ..., r$ и имеет в $C^0$ цвет $i$.

Для удобства введем обозначения для некоторых событий. Обозначим через $\mathcal{F}_A(n-t)$, событие, заключающееся в том, что вершины $w$ корня $A$, покрашены так, что $T(A, T(B_1), ..., T(B_s))$ — $T$-сложное дерево. Соответственно, через $\mathcal{F}_{B_i}(t_i)$ — событие, заключающееся в том, что вершины $u$ раскрашены соответствующим образом. Тогда,

$$\mathsf{P}(T(A, B_1, ..., B_s)) \leq \left(\prod_{i=1}^{s} \mathsf{P}(T(B_i))\right)\left(\prod_{i=1}^{s} \mathsf{P}(\mathcal{F}_{B_i}(t_i))\right)\mathsf{P}(\mathcal{F}_A(n-t)) \tag{4}$$

Выпишем оценки для каждого множителя:

$$\mathsf{P}(T(B_i)) \leq \left(\frac{1}{r}\right)^{(n-1)m_i+1} p^{m_i-1}, \tag{5}$$

$$\mathsf{P}(\mathcal{F}_{B_i}(t_i)) \leq t_i \cdot p \cdot \tilde{p}^{(t_i-1)} = r^{-(t_i-1)} \cdot (r\tilde{p})^{(t_i-1)} \cdot t_i \cdot p, \tag{6}$$

$$\mathsf{P}(\mathcal{F}_A(n-t)) \leq r\left(\frac{1+\tilde{p}(r-1)}{r}\right)^{n-t} \leq r^{t+1-n}\left(1+\tilde{p}(r-1)\right)^n \tag{7}$$

Оценки (3) и (4) получены из того факта, что если подмножество вершин $V_i$ размера $q$ выбирается случайно, то для любого набора $\{v_{i_1}, ..., v_{i_l}\}$ совместная вероятность $\mathsf{P}(\{v_{i_1}, ..., v_{i_l}\} \in V_i) \leq \prod_t \mathsf{P}(v_{i_t} \in V_i \leq \prod_t \left(\frac{q}{m/r-q/2}\right)$, где $m/r - q/2$ — минимально возможный размер цветового класса $K_i$, из которого выбирается подмножество $V_i$(см.3 ). Отсюда, в частности, следует, что

$$\tilde{p}r = \frac{r \cdot q}{m/r - q/2} \leq \frac{r \cdot \left(\frac{n}{\ln(n)}\right)^{\lfloor \log_2(r)/\log_2(r)+1 \rfloor} + 2r\sqrt{\frac{2m\ln r/c}{r}}}{m/r - q/2} \to 0$$

при $n \to \infty$ и $r < \left(\frac{n}{\ln n}\right)^{\ln 2}$.

Подставим в (1) оценки (2), (3) и (4).

$$\mathsf{P}(T(A, B_1, ..., B_s)) \leq r\left(\frac{1}{r}\right)^{(n-1)m+1} \cdot p^{m-1} \cdot (1+\tilde{p}(r-1))^n \cdot \prod_{i=1}^{s} (r\tilde{p})^{(t_i-1)} \cdot t_i$$

Рассмотрим функцию $f(x) = (r\tilde{p})^{(x-1)} \cdot x$ на $[1; \infty)$. Ее можно переписать в виде $f(x) = \exp(-\beta(x-1)) \cdot x$, где $\beta = \ln\left(\frac{1}{r\tilde{p}}\right) > 1$. Функция $f(x)$ убывает на $[1; \infty)$, так как ее производная $f'(x) = (1-\beta x)\exp(-\beta(x-1))$ равна нулю при $x = 1/\beta < 1$ и отрицательна при $x > 1/\beta$. Поэтому, наибольшее значение $f(x)$ на $[1; \infty)$ равно 1 при $x = 1$.

С учетом полученной оценки на $f(x)$:

$$\mathsf{P}(T(A, B_1, ..., B_s)) \leq p^{m-1} \cdot (1+\tilde{p}(r-1))^n \left(\frac{1}{r}\right)^{(n-1)m}$$

Суммируя по всем $m$ произведения $\mathsf{P}(T(A, B_1, ..., B_s))m \cdot \binom{|E|}{m} 2^{m-1}$,

$$\mathsf{E}X \leq \sum_{m=1} p^{m-1} m \cdot \binom{|E|}{m} \cdot 2^{m-1} \cdot (1 + \tilde{p}(r-1))^n \left(\frac{1}{r}\right)^{(n-1)m} \leq ck(1 + \tilde{p}(r-1))^n.$$

□

**Следствие 3.** *Пусть раскраска $C^0$ и отображение $\sigma$ заданы случайно, тогда с положительной вероятностью одноцветных ребер в раскраске $C^2$ будет меньше, чем $q/2$.*

*Доказательство.* Согласно лемме 6,

$$\mathsf{E}X \leq ck(1 + \tilde{p}(r-1))^n,$$

поэтому с положительной вероятностью

$$X < \frac{k(1 + \tilde{p}(r-1))^n}{2}$$

Оценим $2X$:

$$2X \leq k(1 + \tilde{p}(r-1))^n \leq k \exp\{n\tilde{p}(r-1)\} = k \exp\left\{ n(r-1) \frac{q}{m/r - q/2} \right\}$$

(используя явные соотношения (1) на $q$ и $k$, а также оценку на $m$ из леммы 1)

$$\leq k \exp\left\{ n(r-1) \frac{\left(k + 2\sqrt{2(\ln r/c)m/r}\right)}{m/r - q/2} \right\} \leq k \exp\left\{ n(r-1) \frac{\left(4\sqrt{2(\ln r/c)m/r}\right)}{m/r - q/2} \right\} \leqslant$$

$$\leq k \exp\left\{ 4(r-1)\sqrt{2(\ln n) 2 \ln r/c} \right\} \leqslant k \exp\left\{ (\ln n) \cdot 8r \sqrt{\frac{\ln r/c}{\ln n}} \right\}$$

(предположим, что $\frac{8r\sqrt{\ln r/c}}{\sqrt{\ln n}} \leq \frac{1}{\log_2 r + 1}$ )

$$\leqslant k \cdot \exp\left\{ \frac{\ln n}{\log_2 r + 1} \right\} \leq k \cdot n^{\frac{1}{\log_2 r + 1}} = \frac{n}{(\ln n)^{\frac{\log_2 r}{\log_2 r + 1}}} \leq q = \left( k + 2\sqrt{\frac{2m(\ln r/c)}{r}} \right).$$

Осталось показать, что действительно верно соотношение

$$\frac{8r\sqrt{\ln r/c}}{\sqrt{\ln n}} \leq \frac{1}{\log_2 r + 1},$$

когда $r \leq \sqrt{\frac{\ln n}{\tilde{c}(\ln \ln n)^3}}$

$$\frac{8r(\log_2 r + 1)\sqrt{\ln r/c}}{\sqrt{\ln n}} \leq \frac{8 \cdot \sqrt{\frac{\ln n}{\tilde{c}(\ln\ln n)^3}} (\ln \ln n + 1) \sqrt{\ln \ln n + \ln 1/c}}{\sqrt{\ln n}} \leq$$

(подставляя оценку на $r$)

$$\leq 8 \cdot \sqrt{\frac{1}{\tilde{c}(\ln \ln n)^3}} (\ln \ln n + 1) \sqrt{\ln \ln n \cdot \ln 1/c} < 1$$

(уже при $\tilde{c} \geq 64 \ln \frac{1}{c}$)

□

# 6 Доказательство Теоремы 1

Пусть $H = (V, E)$ $n$-однородный гиперграф $H$, в котором
$|E| \leq c\left(\frac{n}{\ln n}\right)^{\lfloor \log_2(r)/\log_2(r)+1\rfloor} r^{n-1}$, а число цветов $r \leq \sqrt{\frac{\ln n}{\tilde{c}(\ln\ln n)^3}}$.

Докажем, что $H$ можно справедливо раскрасить в $r$ цветов. Согласно лемме 1, мы можем считать что число вершин $m > \frac{n^2(r-1)}{2\ln n}$. Рассмотрим случайную раскраску $C^0$ и отображения $\sigma : V \to [0,1]$, где $\sigma(v), v \in V$ — независимые случайные величины с равномерным распределением на $[0,1]$. Применим к гиперграфу $H$ Алгоритм 1 для построение правильной раскраски. Алгоритм 1, согласно Лемме 5 построит из $C^0$ некоторую правильную раскраску $C^1$ с вероятностью не меньше, чем $1 - 8c$.

После работы Алгоритма 1 мощности цветовых классов $K_i$ могут отличаться от $m/r$, но с вероятностью не меньше, чем $1-10c$ выполнено, что $q/2 \leq |K_i| - m/r \leq q/2$. Выберем случайно в каждом цветовом классе $K_i$ подмножество вершин $V_i$ размера $q$ и применим Алгоритм 2. Алгоритм 2 сделает все цветовые классы равными $m/r$. Если при этом, мы сможем выбирать из множеств $V_i$ вершины для перекраски таким образом, чтобы не появлялись новые одноцветные ребра, то мы получим справедливую раскраску.

Следствие 3 утверждает, что с вероятностью не меньше, чем $1 - 2c$ число всех одноцветных ребер будет меньше, чем $q/2$. Тогда если мы не будем брать по одной вершине из каждого такого ребра, то в каждом $V_i$ останется еще хотя бы $q/2$ вершин. Этого числа достаточно, чтобы Алгоритм 2 уровнял мощности цветовых классов.

Выбор параметров: вероятность того, что у нас не получилось справедливой раскраски не превосходит $8c + 10c + 2c = 20c < 1$ уже при $c < 0.05$. Теорема 1 доказана.

# Список литературы


[1] A. Kostochka, “Coloring uniform hypegraphs with few colors”, *Random Structures and Algorithms*, **24** (2010), 1–10.

[2] P. Erdős , “Theory of Graphs and Its Applications” (M. Fieldler, Ed.) Czech. Acad. Sci. Publ., Prague, **9** (1964), 159.

[3] A. Hajnal, E. Szemeredi, “Proof of a conjecture of P. Erdős”, *Combinatorial theory and its applications*, North-Holland, London, **II** (1969), 601-–623

[4] H. A. Kierstead, A. V. Kostochka, “A short proof of the Hajnal-Szemeredi Theorem on equitable coloring”, *Combinatorics, Probability and Computing*, **17** (2008), 265–270.

[5] A. V. Kostochka, M. Mydlarz, E. Szemeredi, H.A. Kierstead, “A fast algorithm for equitable coloring”, *Combinatorica*, **30(2)** (2010), 217–224.

[6] D. A. Shabanov "Equitable two-colorings of uniform hypergraphs"/ European Journal of Combinatorics, т.43 (2015), с. 185-203

[7] И. А. Аколзин "О справедливых раскрасках простых гиперграфов"/ ТРУДЫ МФТИ, т.9, №4 (2017), с. 161–173