2-k ядро для Vertex Cover. Целочисленное линейное программирование. Imbalance problem параметризованная вершинным покрытием.

Николай Чухин

5 октября 2023 г.

Содержание

Целочисленное линейное программирование

ILP

Дан набор целочисленных переменных, требуется найти такие значения переменных, что их линейные комбинации удовлетворяют набору линейных ограничений(тоже целых) и максимизируют некоторую линейную функцию. Т.е. дано $A \in \mathbb{Z}^{m \times p}, b \in \mathbb{Z}^m, c \in \mathbb{Z}^p$, требуется макс. $c^T x \colon Ax \leqslant b$.

Vertex Cover как задача ILP

Переменные x_v для каждой вершины $v \in V(G)$.

$$\sum_{v \in V(G)} x_v \to \min$$

$$\begin{aligned} x_u + x_v \geqslant 1, &\quad \forall (u,v) \in E(G) \\ 0 \leqslant x_v \leqslant 1, &\quad \forall v \in V(G) \\ x_v \in \mathbb{Z}, &\quad \forall v \in V(G) \end{aligned}$$

Линейное программирование

LP

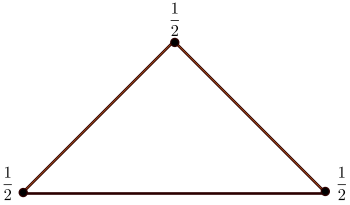
Аналогична задаче ILP, но переменные(и ограничения) могут принимать любые вещественные значения. Решается за полиномиальное время.

Vertex Cover как задача LP (LPVC)

$\sum_{v \in V(G)} x_v \to \min$

$x_u + x_v \geqslant 1, \quad \forall (u, v) \in E(G)$

$0 \leqslant x_v \leqslant 1, \quad \forall v \in V(G)$



Контрпример

2-k ядро для Vertex Cover

У задачи Vertex Cover существует ядро размера $2k$.

Зафиксируем решение $(x_v)_{v \in V(G)}$ задачи LPVC(G). Разобьем вершины графа на 3 множества:

$$V_0 = \left\{ v \in V(G) \mid x_v < \frac{1}{2} \right\}$$
$$V_{\frac{1}{2}} = \left\{ v \in V(G) \mid x_v = \frac{1}{2} \right\}$$
$$V_1 = \left\{ v \in V(G) \mid x_v > \frac{1}{2} \right\}$$

Nemhauser-Trotter theorem

Существует минимальное вершинное покрытие S графа G, такое что $V_1 \subseteq S \subseteq V_1 \cup V_{\frac{1}{2}}$

Пусть $S^* \subseteq V(G)$ - минимальное вершинное покрытие, положим $S = (S^* \setminus V_0) \cup V_1$. Тогда S тоже вершинное покрытие, ведь у каждой взятой вершины из V_0 мы обязаны взять и соседа, чтобы сумма на ребре была больше 1. Более того, $V_1 \subseteq S \subseteq V_1 \cup V_{\frac{1}{2}}$.

Покажем, что S является минимальным покрытием. Пусть $|S| > |S^*|$, но т.к. $|S| = |S^*| - |V_0 \cap S^*| + |V_1 \setminus S^*|$ мы получаем:

$$|V_0 \cap S^*| < |V_1 \setminus S^*|$$

Положим $\varepsilon = \min_{v \in V_0 \cup V_1}(|x_v - \frac{1}{2}|)$. Тогда уменьшим все значения переменных из $V_1 \setminus S^*$ и увеличим в $V_0 \cap S^*$ на ε мы не нарушим ограничений LP, а при этом уменьшим целевую функцию на положительное число. $\square$

Алгоритм нахождения ядра:

1. Если решение LPVC$(G) > k$, то возвращаем NO-instance
2. Добавляем все вершины из V_1 в S(удаляя из графа элементы $V_0 \cup V_1$) и уменьшаем k на $|V_1|$.
3. Продолжаем процесс пока размер $V_0 \cup V_1$ положителен.

Итоговый граф и будет ядром.

Lemma

Размер ядра не превосходит $2k$.

Доказательство.

(G', k') - граф и новое число k после редукций.

$$|V(G')| = |V_{\frac{1}{2}}| = \sum_{v \in V_{\frac{1}{2}}} 2x_v = 2 \sum_{u \in V(G')} x_v \leqslant 2k$$

$\square$

Содержание

Выполнимость задачи линейного целочисленного программирования

Integer Linear Programming Feasibility(ILPF)

Аналогична задаче ILP, но достаточно просто проверить, существует ли решение удовлетворяющие ограничениям, т.е.: Т.е. дано $A \in \mathbb{Z}^{m \times p}, b \in \mathbb{Z}^m$, существует ли $x \in \mathbb{Z}^p \colon Ax \leqslant b$?

Экземпляр ILPF размера L с p переменными можно решить за время $O(p^{2.5p+o(p)} \cdot L)$.

Следствие

Экземпляр ILP размера L с p переменными, где переменные ограничены числом M_x и ограничения числом M_c можно решить за время:

$$O\left(p^{2.5p+o(p)} \cdot (L + \log M_x) \log(M_x M_c)\right)$$

Imbalance problem

Imbalance problem

Дан неориентированный граф $G = (V, E)$, $n = |V(G)|$. **Порядок** на вершинах это любое биективное отображение $\pi\colon V(G) \to [n]$. Для $v \in V(G)$ положим

$$L_\pi(v) = \{u \in N(V)\colon \pi(u) < \pi(v)\}$$

$$R_\pi(v) = \{u \in N(V)\colon \pi(u) > \pi(v)\} = N(v) \setminus L_\pi(v)$$

$$\iota_\pi(v) = ||L_\pi(v)| - |R_\pi(v)||\text{- imbalance вершины } v$$

$$\iota_\pi = \sum_{v \in V(G)} \iota_\pi(v) \to \min$$

Параметризуем Imbalance problem вершинным покрытием X размера k.

FPT алгоритм для Imbalance (1)

Граф G, независимое мн-во X размера k.
Переберем все возможные $\pi_X : X \to [k]$ за время $k!$. Будем искать оптимальную $\pi : V(G) \to [n]$ согласованную с π_X, т.е. $\forall u, v \in X$, чтобы $\pi_X(u) < \pi_X(v) \iff \pi(u) < \pi(v)$.
Пусть $X = \{u_1, u_2, \dots, u_k\}$, где $\pi_X(u_1) < \pi_X(u_2) < \dots < \pi_X(u_k)$.
Обозначим $X_i = \{u_1, u_2, \dots, u_i\}$ - префикс X длины i.
Т.к. X - вершинное покрытие, то множество $I = V(G) \setminus X$ - независимое множество.
Назовем **типом** вершины v множество $N(v) \subseteq X$. Для фиксированного типа $S \subseteq X$, обозначим $I(S)$ множество вершин из I с типом S. Всего различных типов не более 2^k. Понятно, что две вершины одного типа для нас теперь не различимы.
Скажем, что **позиция** вершины v это такое наибольшее $i : \pi(u_i) < \pi(v)$.
Множество вершин на позиции i обозначим как L_i.

FPT алгоритм для Imbalance (2)

Будем искать оптимальный порядок в два этапа - сначала разобьем множество I на блоки $L_0, L_1, \ldots, L_k$. А потом внутри каждого блока L_i ищем оптимальный порядок уже на элементах этого блока. Заметим, что второй этап алгоритма не меняет ι какой либо вершины. Ведь для $v \in I$ все ее соседи лежат в X, а порядок на X зафиксирован, как и относительный порядок для вершины $v \in I$ и любой $u \in X$.
Сформулируем поиск оптимального порядка в терминах ILP.
Для каждого типа S и позиции i переменная x_S^i кодирует количество вершин типа S на позиции i.
Также, для каждой вершины $u_i \in X$ введем y_i кодирующую нижнюю границу $\iota(u_i)$.
И для каждого типа S и позиции i посчитаем $C(S, i)$ - imbalance вершины типа S если ее позиция i, т.е.

$$C(S, i) = ||S \cap X_i| - |S \cap (X \setminus X_i)||$$

FPT алгоритм для Imbalance (3)

Обозначим $e_i = |N(u_i) \cap X_{i-1}| - |N(u_i) \cap (X \setminus X_i)|$.
Итоговый вид в ILP:

$$
\begin{aligned}
& \sum_{i=1}^{k} y_i + \sum_{i=0}^{k} \sum_{S \subseteq X} C(S,i) x_S^i \to \min \\
\text{s.t.} & \sum_{i=0}^{k} x_S^i = |I(S)| && \forall S \subseteq X \\
& x_S^i \geqslant 0 && \forall 0 \leqslant i \leqslant k,\ \forall S \subseteq X \\
& y_i \geqslant \left| e_i + \sum_{S \subseteq X, u_i \in S} \left(\sum_{j=0}^{i-1} x_S^j - \sum_{j=i}^{k} x_S^j \right) \right| && \forall 1 \leqslant i \leqslant k
\end{aligned}
$$

FPT алгоритм для Imbalance (4)

Т.к. значения всех переменных не превосходят n, то запустив алгоритм для ILP мы получим итоговое время работы:

$$2^{2^{O(k)}} n^{O(1)}$$

А значит, задача Imbalance находится в классе FPT.