CSDM: Теоретическая Информатика и Дискретная Математика

17 мая 2026 г.

Данное направление, как понятно из названия, посвящено различным задачам, возникающим на стыке CS и DM. Речь идет в первую очередь о комбинаторных задачах, к которым сводятся различные алгоритмические вопросы. например, часто возникают задачи на стыке теории графов и вероятностного метода, а также задачи округления линейных программ. Ещё одно направление – это алгоритмические построения на случайных графах. Ниже мы приводим некоторые из тем, над которыми будем работать на ЛИПС'26.

1 Спаннеры, или как эффективно приближать расстояния

Пусть дан (невзвешенный) граф $G = (V, E)$. Между любыми двумя вершинами $u, v \in V$ мы можем посчитать графовое расстояние $d_G(u, v)$. (α, β)*-спаннер* $H \subset G$ это подграф G, для которого выполнено, что $d_H(u, v) \leq \alpha d_G(u, v) + \beta$ для любых вершин $u, v \in V$. Мы пока сконцентрируемся на случае $(1, \beta)$-спаннеров, или $+\beta$-спаннеров.

Общий вопрос в этой теме звучит так: найти оптимальный трейд-офф между количеством рёбер в H и параметрами спаннера, в частности, β для $+\beta$-спаннерами. Известно, что для $\beta = 2$ нужно $\Theta(n^{3/2})$ ребер, для $\beta \geq 6$ — $\Theta(n^{4/3})$ ребер. Однако, для $\beta = 4$ правильное количество рёбер неизвестно. Поэтому центральный вопрос здесь следующий:

Problem 1. Доказать (или опровергнуть), что для +4-спаннеров в произвольном графе G на n вершинах достаточно $\tilde{O}(n^{4/3})$ рёбер.

Одна из причин, по которым нам интересны спаннеры – это то, что они позволяют эффективно вычислять приближенные расстояния. Следующий шаг в этом направлении - это компактные "distance certificates" $L(v)$, т.е., строки со свойством, что по $L(v)$, $L(u)$ можно приближенно вычислить расстояние между u, v. (Например, опять же, с ошибкой не более $+\beta$.) Мы бы, например, хотели, чтобы из +6-спаннера с $\tilde{O}(n^{4/3})$ рёбрами можно было получить сертификаты размера $\tilde{O}(n^{1/3})$. Один из способов это сделать - построить семейства *tree covers (древесных покрытий)*: набора из k деревьев $T_1, \dots, T_k \subset G$, со свойством, что для любой пары вершин u, v $\min_{i\in[k]} d_T(u, v) \leq d_G(u, v) + \beta$. Однако, для покрытий деревьями неизвестны оценки, аналогичные вышеуказанным.

Problem 2. Обобщить оценки для аддитивных спаннеров на случай покрытий деревьями ($n^{1+\gamma}$ ребер $\to n^\gamma$ деревьев).

Строить сертификаты расстояний можно и другими способами, поэтому более простой вопрос — это получить сертификаты размера $\tilde{O}(n^\gamma)$ в постановке выше.

Ещё одно интересное промежуточное направление — изучить задачи об аддитивных спаннерах (или даже подструктурах, сохраняющих расстояния в точности) для подмножеств

$P \subset V \times V$. Здесь в большинстве случаев неизвестны правильные пороги на количество ребёр в зависимости от $|P|$ и от β, и интересных открытых вопросов множество.

Возможно, самая фундаментальная комбинаторная задача в этом направлении — это Erdős Girth Conjecture, которая утверждает, что максимальное количество ребер в графе без циклов длины $\leq 2k$ равно $\Theta(n^{1+\frac{1}{k}})$. Эта гипотеза открыта для всех k кроме $k = 2, 3, 5$. В теории спаннеров есть немало "условных" результатов, которые предполагают верность Erdős Girth Conjecture.

2 Discrepancy, Fair Division For Non-Additive Valuations and Rounding for Approximation Algorithms

Discrepancy, в широком смысле, позволяют производить округления дробных решений: дана матрица ограничений A и дробное решение x, (обычно полученное как оптимальное решение линейной релаксации), и нужно округлить x до целочисленного решения y, почти не нарушив линейные ограничения:

$$\|A(x - y)\|_\infty \approx 0.$$

Формально,

$$\mathrm{lindisc}(A) := \max_{x \in [0,1]^n} \min_{y \in \{0,1\}^n} \|A(x - y)\|_\infty.$$

Иными словами, каждое ограничение, заданное строкой A, после округления портится не более чем на $\mathrm{lindisc}(A)$.

Мы хотим исследовать связь между discrepancy theory, fair division и rounding techniques for approximation algorithms. Отправной точкой являются работы [3] и [4], где combinatorial discrepancy обобщается за пределы аддитивных функций и применяется к fair division для non-additive valuations.

В нашей недавней работе мы доказали результат для аддитивных оценок в group fair division и закрыли соответствующую открытую задачу. Теперь мы хотим понять, можно ли перенести эти методы на non-additive valuations, где ценность множества предметов уже не раскладывается в сумму независимых вкладов.

Также нас интересует приложение этих идей к scheduling, а именно к задаче Unrelated Parallel Machine Scheduling with Machine Groups. Её можно рассматривать как задачу дележа с минимизацией самого богатого участника: работы назначаются с учётом групповой структуры машин, а цель состоит в контроле максимальной нагрузки.

Параллельно мы хотим изучить несколько вариантов discrepancy, возникающих в approximation algorithms. Работы [1] и [2] показывают, что prefix discrepancy особенно полезна для rounding в scheduling problems и задачах с временной префиксной структурой. В таких приложениях нужно контролировать не только итоговое отклонение $A(x - y)$, но и отклонение на всех префиксах или, более общо, на выделенных подмножествах переменных.

Для этого естественно рассматривать следующий параметр:

$$\mathrm{combdisc}(A, \mathcal{F}) := \max_{x \in [0,1]^n} \min_{y \in \{0,1\}^n} \max_{F \in \mathcal{F}} \|A_F(x - y)\|_\infty,$$

где A_F обозначает подматрицу A, состоящую из столбцов с индексами из F, а $\mathcal{F}$ — произвольное семейство подмножеств столбцов A. Prefix discrepancy является важным частным случаем:

$$\mathrm{prefdisc}(A) := \mathrm{combdisc}\big(A, \{[1], [2], \ldots, [n]\}\big).$$

Наша цель — получить общие bounds на `prefdisc` и `combdisc` и понять, как использовать их как универсальные rounding primitives для LP и convex relaxations. Основные цели проекта:

1. Обобщить методы group fair division с аддитивных оценок на non-additive valuations.
2. Связать эти методы с non-additive discrepancy framework из [3] и [4].
3. Применить идеи group fair division к Unrelated Parallel Machine Scheduling with Machine Groups как к отдельному scheduling-приложению.
4. Получить общие оценки на `prefdisc` и `combdisc`, мотивированные приложениями в approximation algorithms, в духе [1] и [2].

Список литературы

[1] Nikhil Bansal, Haotian Jiang, Raghu Meka, Sahil Singla, and Makrand Sinha. Prefix discrepancy, smoothed analysis, and combinatorial vector balancing. *arXiv preprint arXiv:2111.07049*, 2021.

[2] Nikhil Bansal, Lars Rohwedder, and Ola Svensson. Flow time scheduling and prefix beck-fiala. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 331–342, 2022.

[3] Alexandros Hollender, Pasin Manurangsi, Raghu Meka, and Warut Suksompong. Discrepancy beyond additive functions with applications to fair division. *arXiv preprint arXiv:2509.09252*, 2025.

[4] Max Dupre la Tour and Kaito Fujii. Discrepancy and fair division for non-additive valuations. *arXiv preprint arXiv:2509.16802*, 2025.

3 Semi-Random Graph Process

Рассмотрим следующий случайный процесс: мы стартуем с n вершин, а далее на каждом шаге нам выдают равномерно случайную вершину, а мы вольны выбрать ребро из неё в любую другую вершину. Это называется Semi-Random Graph Process, и он весьма активно изучался в последние 5 лет. Одна из основных целей - насколько можно в таком варианте ускорить появление различных подструктур в таком графе по сравнению со случайным графом $G(n, m)$ (в котором m рёбер выбраны полностью случайно). В частности, исследователи изучали задачи о том, за сколько шагов можно построить клики, сделать граф связным, построить совершенное паросочетание и т.д.

Недавно мы предложили следующую обобщённую модель такого типа, параметризованную числом $k \in \{1, \ldots, n\}$: на каждом шаге нам выдают случайную вершину, а также случайное множество S размера k. И далее мы вольны провести ребро в любую вершину из множества S.. Т.е., при $k = 1$ получаем $G(n, m)$, а при $k = n$ получаем semi-random process. Цель рассмотрения данной модели - точно понять, сколько случайности может содержать модель, чтобы при этом ускорять построение различных подструктур. В рамках школы мы планируем получить пороги появления различных подструктур в данной модели.