Об одной задаче, сводимой к нахождению максимального потока в сети

Филимонов Иван Сергеевич

Тверской государственный университет, математический ф-т

Задача 1. Формулировка 1

- Пусть имеется прямоугольное **клеточное поле** (матрица), состоящее из n строк и m столбцов. Требуется соединить некоторые **диагонали** (назовём эти соединения штрихами) в клетках так, чтобы штрихи **не пересекались** между собой и никакие две диагонали, проведённые в разных клетках, **не имели общих концов**. При этом количество штрихов должно быть **максимально**.
- Эту же задачу можно сформулировать иным образом

Задача 1. Формулировка 2

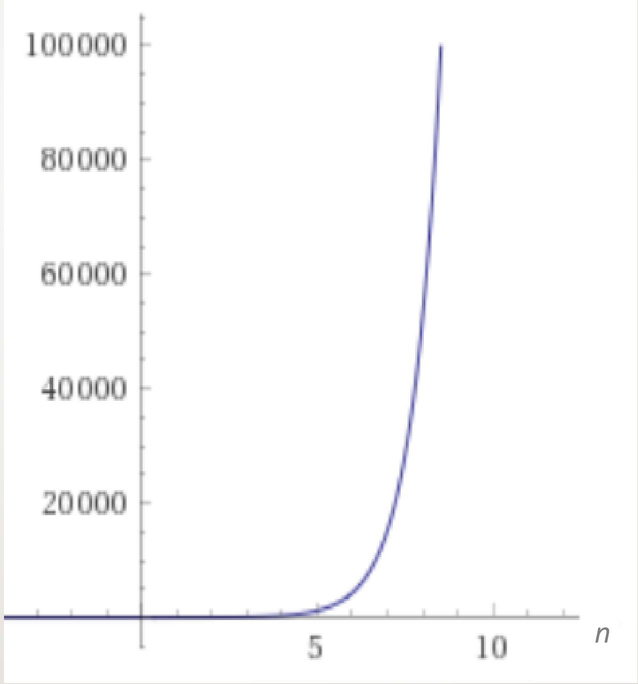
- Провести диагонали в некоторых клетках поля так, чтобы **ни через какую точку** (из четырёх, ограничивающих каждую клетку) не проходило две диагонали и ни в одной клетке не было проведено сразу **две** диагонали. Действительно, то, что штрихи не пересекаются между собой, как раз означает, что ни в одной клетке не проведено сразу две диагонали, а требование о том, что никакие две диагонали, проведённые в разных клетках, не имеют общих концов, эквивалентно требованию о единственности диагонали, проходящей через каждую конкретную точку.

Задача 1. Медленное решение

- Давайте для каждого **столбца** будем кодировать возможное **состояние** каждой его ячейки – пустая, диагональ из верхнего правого угла с левый нижний, из верхнего левого в правый нижний – **числами** 0, 1 и 2 соответственно. Получим для каждого возможного состояния одного столбца набор чисел – **профиль**. Для примера это (1, 0, 2, 2)
- Задача решается с помощью **динамического программирования по профилю**

Задача 1. Оценка сложности

- Если применять стандартные методы, придётся для каждого столбца перебрать все возможные профили в зависимости от n, а их 3^n, и пройтись по всем столбцам, которых m
- **Итоговая асимптотика решения:**
$$\boldsymbol{m * 3^n}$$
- Решение методом ДП по профилю можно ускорить, но экспонента останется. Можно ли быстрее?



Количество операций алгоритма в зависимости от n при примерно равных n и m

Задача 2. Формулировка

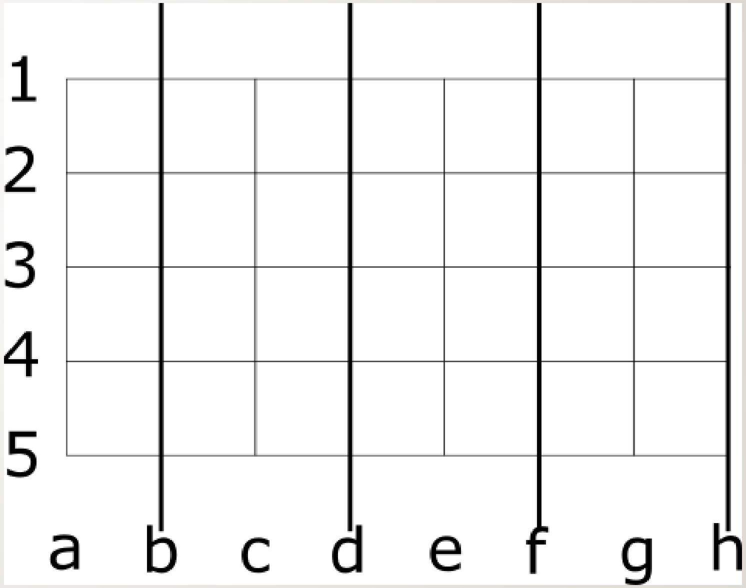
- Пусть имеется прямоугольное клеточное поле (матрица), состоящее из n строк и m столбцов. Провести максимальное количество диагоналей в некоторых клетках поля так, чтобы **ни через какую точку** (из четырёх, ограничивающих каждую клетку) не проходило две диагонали.
- **Отличие от задачи 1 – отсутствие второго условия.**
- Оказывается, такое упрощение позволяет решить задачу с помощью потоков в сетях за **полиномиальное** от n*m время.

Задача 2. Сводим к потокам в сетях

Обозначаем:

- границы строк – числами $0 \dots n+1$
- границы столбцов – латинскими буквами
- точка на пересечении строки i и столбца α - αi

Делим столбцы на чётные и нечётные

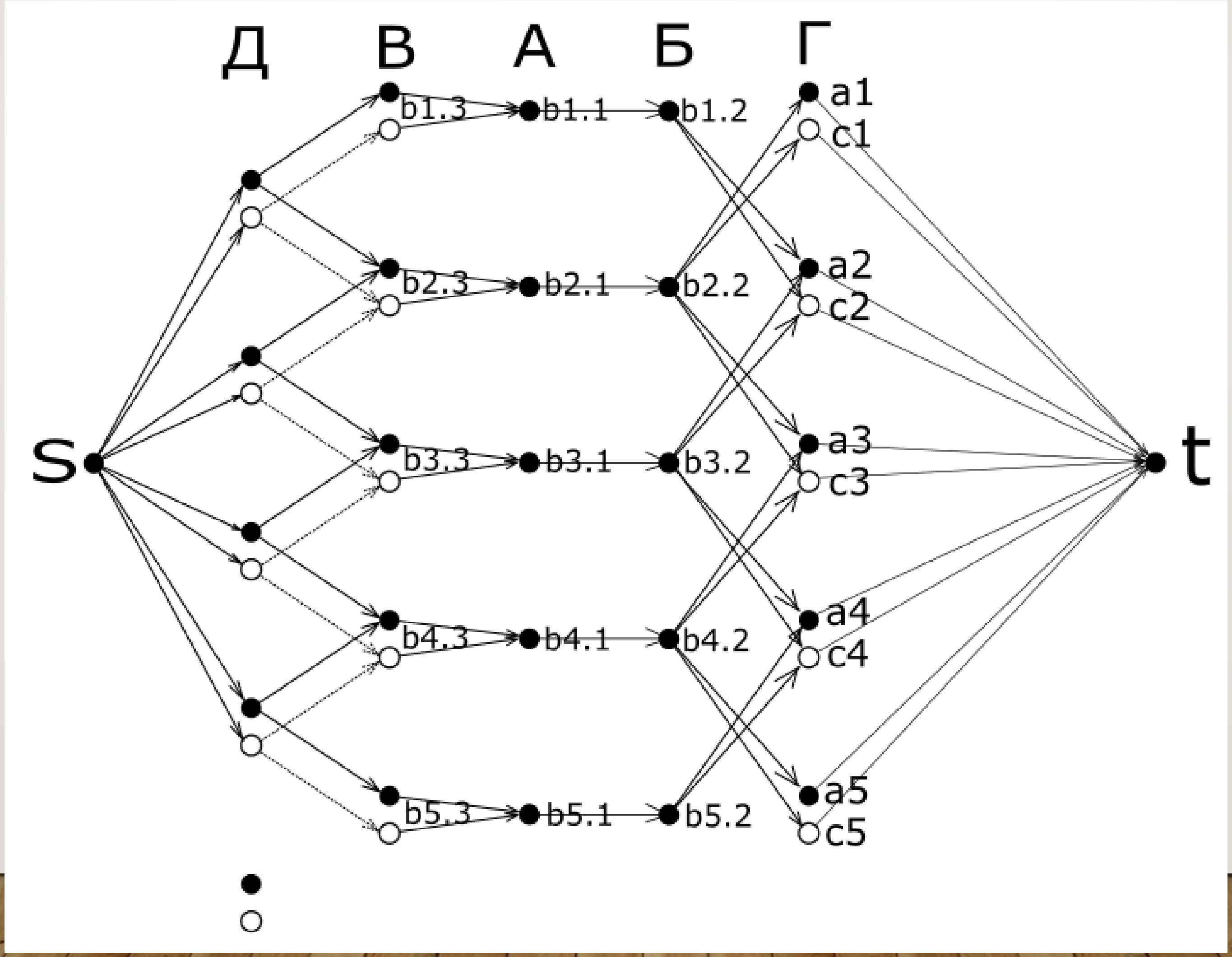


Задача 2. Сводим к потокам в сетях

Рассмотрим столбец b

- Из него могут идти диагонали в столбцы a и c. Сопоставим столбцу b группы А, Б, В, Д вершин сети, столбцам a и c – группу Г (см. рисунок на следующем слайде)
- Исток – вершина s, сток – t
- Точки лежат в одном столбце/одной строке - буквенные/первые числовые индексы одинаковые
- Пропускная способность каждого ребра - единица

Задача 2. Сеть для решения

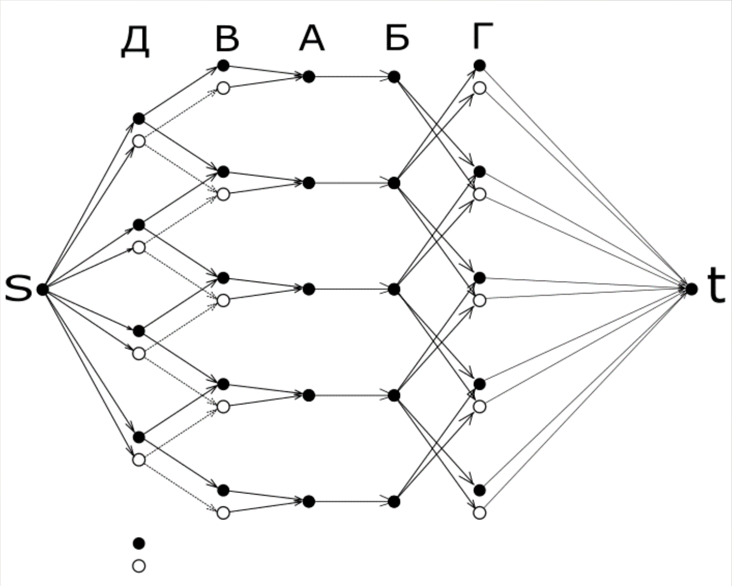


Задача 2. Достраиваем граф

- Для каждого столбца в вершины группы Д добавим две дополнительные изолированные вершины. Это не несёт смысловой нагрузки, но упрощает реализацию и подсчёты.
- Теперь количество вершин группы Д становится равно количеству вершин группы В.
- Аналогично достраивается граф для оставшейся части чётных столбцов: группы А, Б, В, Д для каждого чётного столбца создаются снова, а в некоторые «старые» вершины группы Г рёбра идут из точек новых чётных столбцов.

Задача 2. Свойства сети

- Единственность диагонали через каждую точку - в каждую вершину группы Б, входит ровно одно ребро. Аналогично для вершин группы Г.
- Максимальность количества задействованных диагоналей гарантируется алгоритмом поиска максимального потока в сети.
- Вершины групп А, В и Д могут быть полезны для модификации сети под решение задачи 1.



Задача 2. Итог

- Нахождение максимального потока в сети действительно даст решение задачи 2.
- Количество вершин в сети линейно $O(nm)$ относительно количества клеток nm
- Количество рёбер также есть $O(nm)$
- Максимальный поток можно найти с помощью алгоритма Диница. В единичной сети, которой является нами построенная – за время порядка $O(e\sqrt{v})$, где e – число рёбер, v – вершин.
- Итоговая асимптотика решения - $O\left((nm)^{\frac{3}{2}}\right)$

Как решить задачу 1?

- Как видим, задача 2 с помощью алгоритма поиска максимального потока успешно решается за полиномиальное время. Для исходной же задачи (1) формализовать условие отсутствия двух диагоналей в одной клетке при сохранении второго условия автору статьи не удалось. Вполне возможно, такая формализация есть, и эту задачу также можно решить за полиномиальное время.

Источники информации

- Домнин, Л.Н. Элементы теории графов: учеб. пособие / Л.Н. Домнин. – Пенза: Изд-во Пенз.гос. ун-та, 2007. – 144 с.
- Свами, М. Графы, сети и алгоритмы / М. Свами, К. Тхуласираман. – М.: Книга по Требованию, 2013. – 450 с
- Лааксонен, И. Олимпиадное программирование / пер. с анг. А. А. Слинкин – М.: ДМК Пресс, 2018. – 300 с.