ЦЕНТРАЛЬНЫЙ УНИВЕРСИТЕТ

Домашнее задание #2

Hard deadline по теоретическому заданию: ***23:59 25 февраля***
Hard deadline по заданию на программирование: ***23:59 3 марта***

1 Теоретическое задание [50 баллов]

На лекции мы разбирали задачу распределения статей по рецензентам и сформулировали ее в терминах целочисленного линейного программирования (ILP). Пусть у нас есть d статей $\{1, 2, 3, \ldots, d\} = [d]$ и n рецензентов $\{1, 2, 3, \ldots, n\} = [n]$. Для каждой пары (статья i, рецензент j) определим экспертизу $s_{ij} \in [0, 1]$ рецензента i в проверке статьи j и объединим значения экспертиз в матрицу $S \in [0, 1]^{d \times n}$.

Цель задачи — найти распределение рецензентов по статьям, которое максимизирует суммарную экспертизу по всем парам (статья, рецензент) и при этом не нарушает ограничений:

- **Paper load constraints.** Каждую статью проверяют *не менее* $\lambda \in \mathbb{N}$ рецензентов[1]
- **Reviewer load constraints.** Каждый рецензент проверяет *не более* $\mu \in \mathbb{N}$ статей

Чтобы формализовать эту задачу, введем бинарную матрицу $X \in \{0, 1\}^{d \times n}$. Матрица X определяет распределение статей по рецензентам: элемент x_{ij} равен 1 тогда и только тогда, когда рецензент j назначен на проверку статьи i. В этих обозначениях мы можем формально ввести задачу распределения статей по рецензентам:

$$
\begin{aligned}
\underset{X \in \{0,1\}^{d \times n}}{\text{maximize}} \; & \sum_{i \in [d]} \sum_{j \in [n]} s_{ij} x_{ij} \\
\text{subject to} \; & \sum_{i \in [d]} x_{ij} \leq \mu \quad \forall j \\
& \sum_{j \in [n]} x_{ij} \geq \lambda \quad \forall i
\end{aligned}
\tag{1}
$$

Современные конференции (например, NeurIPS) каждый год получают тысячи статей, которые нужно распределить по рецензентам. Решение произвольной задачи целочисленного линейного программирования (ILP) таких размеров может быть непрактично долгим.

К счастью, в этой домашней работе мы покажем, что наша конкретная задача (1) обладает особой структурой, которая позволяет снять ограничения на целочисленность и эффективо решить задачу с помощью симплекс-метода (решение LP-релаксации сразу даст целочисленное решение). *Цель теоретической части домашнего задания — найти эту структуру.*

[1] В отличие от лекции в этом задании мы разрешаем назначить на статью больше чем λ рецензентов. Это техническое изменение немного упростит доказательства.

Пункт 1 **10 баллов**

Перепишите задачу (1) в каноническом виде целочисленного линейного программирования:

$$\begin{aligned} \underset{x\in\{0,1\}^{\mathbf{dim}}}{\text{maximize}} \quad & \langle c, x\rangle \\ \text{subject to} \quad & Ax \le b, \end{aligned}$$

где x это вектор, полученный за счет конкатенации строк матрицы X. Вам нужно задать A, b, c и **dim**.

Пункт 2 **30 баллов**

Докажите, что матрица A, полученная вами в предыдущем пункте, выполняет свойство TU (totally unimodular).

Определение Матрица выполняет свойство TU (totally unimodular matrix) если любая ее квадратная подматрица имеет определитель -1, 0 или 1.

Набросок доказательства[2]

Докажем утверждение с помощью индукции по размеру квадратной подматрицы в определении свойства TU.

База индукции

Проверьте, что для каждой подматрицы A' размера 1×1 свойство TU выполняется.

Шаг индукции

Пусть свойство TU выполняется для всех квадратных подматриц размера $k \times k$. Покажем, что оно также выполняется для подматриц размера $(k+1)\times(k+1)$. Для этого рассмотрим произвольную подматрицу M размера $(k+1)\times(k+1)$. Заметим, что матрица M принадлежит одному из трех типов:

- *Тип 1.* В матрице M есть столбец, состоящий целиком из нулей.
- *Тип 2.* В матрице M есть столбец, состоящий из k нулей и одного ненулевого элемента.
- *Тип 3.* Каждый столбец матрицы M имеет ровно два ненулевых элемента.

Аккуратно докажите, что возможны только эти три типа. Дальше для каждого из типов проанализируйте определитель матрицы M и докажите, что матрица M удовлетворяет свойству TU.

Подсказка 1. При исследования типа 2 воспользуйтесь свойством, что перестановка строк матрицы меняет только знак определителя.

Подсказка 2. При исследования типа 3 сложите строки матрицы. Вспомните, что если строки матрицы линейно зависимы, то определитель такой матрицы равен нулю.

Пункт 3 **10 баллов**

Примените теорему из лекции (Lecture 5: Optimization II, слайд 35) и запишите LP-релаксацию задачи, решение которой дает решение исходной задачи ILP (1).

[2] Ваша задача — аккуратно восстановить доказательство.

2 Задание на программирование [50 баллов]

В этой задаче вам предстоит реализовать алгоритм Branch and Bound для решения судоку. В этой задаче судоку определяется двумерным массивом размера 9×9. Например,

```
puzzle = [[4, 8, 0, 3, 0, 0, 0, 0, 0],
          [0, 0, 0, 0, 0, 0, 0, 7, 1],
          [0, 2, 0, 0, 0, 0, 0, 0, 0],
          [7, 0, 5, 0, 0, 0, 0, 6, 0],
          [0, 0, 0, 2, 0, 0, 8, 0, 0],
          [0, 0, 0, 0, 0, 0, 0, 0, 0],
          [0, 0, 1, 0, 7, 6, 0, 0, 0],
          [3, 0, 0, 0, 0, 0, 4, 0, 0],
          [0, 0, 0, 0, 5, 0, 0, 0, 0]],
```

Нулевые элементы обозначают клетки, которые должны быть заполнены вашим алгоритмом, а остальные числа (от 1 до 9) обозначают клетки, которые заполнены по условию. Ваша функция `solve_sudoku` должна возвращать tuple `(solved_puzzle, constraints)`, где `solved_puzzle` это решенная задача судоку, а `constraints` это дополнительные ограничения, которые позволяют алгоритму Branch and Bound найти целочисленное решение. Например, для задачи выше решение будет выглядеть так:

```
solved_puzzle = [[4, 8, 7, 3, 1, 2, 6, 9, 5],
                 [5, 9, 3, 6, 8, 4, 2, 7, 1],
                 [1, 2, 6, 5, 9, 7, 3, 8, 4],
                 [7, 3, 5, 8, 4, 9, 1, 6, 2],
                 [9, 1, 4, 2, 6, 5, 8, 3, 7],
                 [2, 6, 8, 7, 3, 1, 5, 4, 9],
                 [8, 5, 1, 4, 7, 6, 9, 2, 3],
                 [3, 7, 9, 1, 2, 8, 4, 5, 6],
                 [6, 4, 2, 9, 5, 3, 7, 1, 8]]

constraints = [(8, 5, 2, 1), (5, 3, 4, 1)].
```

Значение переменной `constraints` означает, что если мы добавим два ограничения $(x_{8,5})_2 = 1$ и $(x_{5,3})_4 = 1$, то LP-релаксация задачи целочисленного линейного программирования (см. детали ниже) даст целочисленное решение. Обратите внимание, что разные дополнительные ограничения могут приводить к одному и тому же решению.

Решение задачи состоит из двух ключевых шагов, которые мы сейчас обсудим.

Шаг 1. Решение LP-релаксации

Ключевой компонент алгоритма Branch and Bound — симплекс метод, который решает задачу LP-релаксации. Используя библиотеку `cvxpy`, напишите функцию, которая решает LP-релаксацию задачи судоку. Подробнее про `cvxpy` можно прочитать в документации `http://cvxpy.org`.

На лекции 5 мы обсудили, что в задаче судоку мы работаем с переменными $\{(x_{i,j})_k : i \in [9], j \in [9], k \in [9]\}$. Переменная $(x_{i,j})_k$ равна 1 тогда и только тогда, когда в клетке (i, j) записано число k.[3]. В этих обозначениях, LP-релаксация задачи судоку выражается следующим образом:

[3]Будьте аккуратны с тем, что нумерация элементов массива в питоне начинается с нуля

$$
\begin{aligned}
\text{minimize} \quad & \sum_{i,j}^{9} \max_k (x_{i,j})_k \\
\text{subject to} \quad & (x_{i,j})_k \in [0,1] \quad i,j,k = 1...9 && \text{(релаксация бинарных ограничений)} \\
& \sum_{k=1}^{9} (x_{i,j})_k = 1, \quad i,j = 1...9 && \text{(в каждой клетке записано одно число)} \\
& \sum_{j=1}^{9} (x_{i,j})_k = 1, \quad i,k = 1...9 && \text{(числа в строках не повторяются)} \\
& \sum_{i=1}^{9} (x_{i,j})_k = 1, \quad j,k = 1...9 && \text{(числа в столбцах не повторяются)} \\
& \sum_{m,l=1}^{3} (x_{i+m,j+l})_k = 1, \quad i,j \in \{0,3,6\}, k = 1...9 && \text{(числа в 9 квадратах не повторяются)} \\
& (x_{i,j})_k = 1 \ \text{ для } (i,j) \text{ заполненных в условии} && \text{(фиксируем значения клеток из условия).}
\end{aligned}
$$

Важно: Даже когда LP-релаксация теоретически дает целочисленное решение, результаты, которые вы получите на практике, могут не быть целочисленными. Например, вместо 1 вы можете получить 0.999995. В этом задании вы можете округлять числа в ε-окрестности 0 и 1, где $\varepsilon = 0.005$.

Бонусные очки Вы можете получить до 10 бонусных очков, если самостоятельно проанализируете целевую функцию задачи LP-релаксации и объясните ее выбор (либо предложите другой, более оптимальный выбор целевой функции).

Шаг 2. Branch and Bound

Реализуйте алгоритм Branch and Bound для решения судоку.

Подсказка: В случае если LP-релаксация не дает целочисленного решения, эффективно фиксировать значение той переменной $\{(x_{(i,j)})_k : i,j,k = 1,...,9\}$, чье текущее значение лежит ближе всего к 0.5.

Шаблон

Шаблоны функций даны в Google Colab:

`https://colab.research.google.com/drive/1kjZ6r-c8ryA_KQ4RAr-0I9EjPCFXXCyf?usp=sharing`

Для решения скопируйте ноутбук себе и оформите решение в нем. Прикрепите ссылку на ваш ноутбук к решению на EDU и убедитесь, что вы выдали доступ на просмотр вашего решения. Пожалуйста, не изменяйте имена заданных функций и убедитесь, что весь ноутбук можно запустить после перезапуска ядра.