

Планирование расписаний и управление доходами

Александр Широков ПМ-1701

Преподаватель:

Васильев Юрий Михайлович

Санкт-Петербург
2020 г., 7 семестр

Список литературы

[1]

Содержание

1	Планирование расписаний и управление доходами	2
1.1	Задача из авиакомпании Россия	2
1.1.1	BiWay (ToWay) Number Partitional Problem	2
1.1.2	MultiWay (ToWay) Number Partitional Problem	4
1.2	Multi Dimensional Multi Way NPP	6
1.3	Рассмотрим взвешенную сумму средних квадратов отклонений	7
1.4	Критерий равномерности	7
1.5	Minimize Difference	8
1.6	Weighted Minimize	10
1.7	Weighted Choose Minimize	12
1.8	Натурные данные 16.09.2020	14
1.9	Метод Сугиямы	17
1.10	Set Covering Formulation	18
1.11	Неравенство Треугольника	19
1.12	Layering	20
1.13	Распределение вершин по слоям с ограничением на ширину слоя	21
1.14	Гиперграф	23
2	Построение расписания для цеха	27
3	Задача маршрутизации транспорта с ограничением на грузо- подъемность	28

1 Планирование расписаний и управление дорогами

1.1 Задача из авиакомпании Россия

В задачах планирования авиаперелетов:

- расписание судов
- маршрутизация
- построение графика полета летного состава

Мы поговорим о построении графика полета летного состава. Зарплата бортпроводника зависит от навыков и от некоторых других факторов, но значительная часть денег тратилась на штрафы, которые выплачивались в пользу бортпроводников, потому что есть *приказ*, о котором бортпроводник не может проводить в воздухе больше определенного времени в воздухе.

Расписание в авиакомпании Россия делалось вручную и компания тратила много денег на выплаты бортпроводникам. OpenSky - программное обеспечение для обслуживания бортпроводников, но оно использовалось.

Множество бортпроводников разбито на 4 подмножества с примерно одинаковыми характеристиками. Каждое подмножество называется **книга**.

Рейс - перелет из Петербурга в Москву, а **связка** - перелет из Петербурга в Москву и обратно.

Множество связок разбивалось на 4 подмножества.

После этого соединяется первая книга и первый рейс и получается **рабочий стол**. Каждый рабочий стол можно описать характеристиками какими-то. С каждым рабочим столом работает один эксперт и все оказываются без перегрузов.

Задача: необходимо так разбить связки на подмножества, чтобы характеристики рабочих столов были примерно одинаковы.

1.1.1 BiWay (ToWay) Number Partitional Problem

Дано n натуральных чисел и мультимножество (элементы могут повторяться) S , которое описывает этот набор n . Нам необходимо разбить подмножество $S = \{s_1, \dots, s_n\}$ на два подмножества, каждое подмножество характеризует сумму чисел, чтобы минимизировать максимальную сумму чисел в подмножестве.

Greedy alghorytm

1. Отсортировать S в порядке убывания
2. На каждом шаге мы последовательно распределяем в две группы, кладем в группу с текущей наименьшей суммой. Если сумма одинакова, то кладем случайно.

Complete Greedy Alghorytm

1. Сортируем мультимножество в порядке убывания (распределяем большие числа и добиваем маленькими)
2. Данный алгоритм исследует бинарное дерево, где каждому уровню - число из отсортированного мультимножества, в каждой вершине - ветвление. В левой ветке - кладем в группу с наименьшей суммой, а в правой ветке - с наибольшей.

Правила, позволяющие сократить размер нашего дерева:

- Если сумма чисел в подмножествах равна, то мы кладем число только в одно подмножество
- Если оставшиеся распределенные числа не превосходят разницу между подмножествами, то мы кладем эти числа в группу с наименьшей суммой.

Домашнее задание: реализация алгоритма, причем настрока алгоритма в трех вариантах:

- Исследует полное дерево решений и находит ответ;
- Алгоритм работает заданное число секунд и возвращает наилучший найденный результат за t время (рекурсивная функция(оставшиеся числа, подмножества 1, подмножества 2))
- Первое найденное решение (первый лист, который мы нашли).

Алгоритм Кармаркара-Карпа (эвристический)

1. Сортируем мультимножество в порядке убывания (распределяем большие числа и добиваем маленькими)
2. Два наибольших числа заменяются на их разницу и кладем эту разницу в список с сортировкой и опять пересортировываем - кладем числа в два разных подмножества (интерпретация).
3. Так делаем, пока не получим одно число: разницу между максимальным и минимальным подмножеством
4. Восстанавливаем

Пример:

$\{16, 15, 12, 10, 5, 1\} \mapsto \{12, 10, 5, 1, 1\} \mapsto \{5, 2, 1, 1\} \mapsto \{3, 1, 1\} \mapsto \{2, 1\} \mapsto \{1\}$

Compete алгоритм Кармакара-Карпа

1. Сортируем мультимножество в порядке убывания (распределяем большие числа и добиваем маленькими)
2. Исследуем бинарное дерево в глубину, исследуя левую ветку

Домашнее задание: реализовать алгоритм для решения.

1.1.2 MultiWay (ToWay) Number Partitional Problem

Дано n натуральных чисел и мультимножество (элементы могут повторяться) S , которое описывает этот набор n . Нам необходимо разбить подмножество $S = \{s_1, \dots, s_n\}$ на K подмножества, каждое подмножество характеризует сумму чисел, чтобы минимизировать:

1. минимизация максимальной суммы
2. максимизация минимальной суммой
3. минимизация разности между наибольшей и наименьшей суммой в подмножествах
4. идеальная сумма - $\frac{S}{K}$ - минимизировать отклонения идеальной суммы

$$X_{i,j} = \begin{cases} 1, & \text{if } S_i \text{ in } j \\ 0 & S_j \end{cases}$$

$$Z - W \rightarrow \min$$

$$\sum_{j=1}^k X_{s,j} = 1 \quad \forall s \in S$$

Z - наибольшая сумма через x , а W - наименьшую сумму через подмножества

$$Z \geq \sum_{i=1}^n s_i X_{i,j} \quad \forall j \in \{1, \dots, k\}$$

$$W \leq \sum_{i=1}^n s_i X_{i,j} \quad \forall j \in \{1, \dots, k\}$$

$$X_{i,j} \in \{0, 1\}, \quad \forall i \in \{1, \dots, n\}, \forall j \in \{1, \dots, k\}$$

Жадный алгоритм

$$L_j(S_1, S_2, \dots, S_k, S_i)$$

данная функция возвращает значение целевой функции, если мы положим число S_i в j -е подмножество.

На каждом шаге алгоритма мы ищем такое $j \in \{1, \dots, k\}$ такое, что значение целевой функции $gr = \operatorname{argmin} L_j$ и так до тех пор пока мы не распределим все наши числа из отсортированного подмножества.

$$S_{gr} = S_{gr} \cup \{S_i\}$$

Программирование:

c - список неизвестных, m - коэффициенты при ограничениях, $\{\{const\}, \{type\}\}$.

Если 0, то равенство, если 1, то \geq , если -1, то \leq . 4-ый аргумент - интервалы, в которых могут применять значения неизвестные - $lboud, ubound$.

Последний - какому множеству чисел принадлежит тип.

Домашнее задание: минимизация сумма отклонения по модулю от идеального разбиения и реализация.

$$\bar{y} = \frac{\sum S}{K}$$

$$\sum_{i=1}^k |y_i - \bar{y}|$$

Линеаризация

- Линеаризация модуля в ограничениях

$$|X| \leq b (X = \sum_{i=1}^n a_i x_i, b \geq 0)$$

$$\begin{cases} x \leq b \\ x \geq -b \end{cases}$$

- Допустимые значения

$$x = 0 \text{ или } 0 \leq X \leq b, a > 0$$

$$y = \begin{cases} 0, & x = 0 \\ 1 \end{cases}$$

$$\begin{cases} x \geq ay \\ x \leq by \\ y \in \{0, 1\} \end{cases}$$

- Условие ИЛИ

$$\sum_{i=1}^n c_i x_i \rightarrow \min$$

$$\sum_{i=1}^n a_{1,i} x_i \leq b_1 + M_1 y$$

$$\sum_{i=1}^n a_{2,i} x_i \leq b_2 + M_2 (1 - y) \quad y \in \{0, 1\}$$

- Модуль со знаком \geq

- IF

$$\text{if } \sum_{i=1}^n a_{1,i} x_i \leq b_1 \rightarrow \sum_{i=1}^n a_{1,i} x_i \geq b_1 + \varepsilon$$

$$\text{then } \sum_{i=1}^n a_{2,i} x_i \leq b_2$$

и мы превратили в третий пункт

$$y \in \{0, 1\}$$

- Умножение бинарных переменных

$$\dots + x_1 \cdot x_2 + \dots \leq b$$

$$x_1, x_2 \in \{0, 1\}$$

$$y \in \{0, 1\}$$

$$y \leq x_1$$

$$y \leq x_2$$

$$y \geq x_1 + x_2 - 1$$

x_1	1	0	0	0
x_2	1	0	1	0
y	1	0	0	0
Линеаризация				

1.2 Multi Dimensional Multi Way NPP

Будем заниматься векторами. Минимизация максимальной разности по координатам между подмножествами.

$$1 : (a_1, a_2, a_3)$$

$$2 : (b_1, b_2, b_3)$$

$$3 : (c_1, c_2, c_3)$$

$$\max(|a_1 - b_1|, |a_2 - b_2|, |a_3 - b_3|) \rightarrow \min$$

Пусть NC - размерность вектора, NV - количество векторов, NK - число групп.

Множество:

$$S = \{s_i | s_i = (s_{i,2}, s_{i,2}, \dots, s_{i,NC})\}, i \in \{1, \dots, NV\}$$

Неизвестные:

$$x_{s,k} = \begin{cases} 1, & -s \in NK \\ 0 \end{cases}$$

Введем дополнительную переменную $y_{c,k}$ - сумма векторов из подмножества k по координате c :

$$\max |y_{c,k_1} - y_{c,k_2}| \rightarrow \min$$

$$k_1, k_2 \in \{1, \dots, NK\}$$

$$c \in \{1, \dots, NC\}$$

Нам нужно найти группу k_1, k_2 и c дают разницу по модулю между соответствующими c .

Так мы делаем для:

1.

$$\sigma \geq y_{c,k_1} - y_{c,k_2} \quad \forall k_1, k_2 \in \{1, \dots, NK\}, k_1 < k_2$$

$$\sigma \geq -y_{c,k_1} + y_{c,k_2} \quad \forall k_1, k_2 \in \{1, \dots, NK\}, k_1 < k_2$$

$$\sigma \rightarrow \min$$

2.

$$\sum_{k=1}^{NK} x_{s,k} = 1, \quad \forall s \in S$$

3.

$$y_{c,k} = \sum_{s \in S} S_c \cdot x_{s,k} \quad \forall c \in \{1, \dots, NC\}, k \in \{1, \dots, NK\}$$

$$x_{s,k} \in \{0, 1\} \quad \forall s \in S, \forall k \in \{1, \dots, NK\}$$

Всего неизвестных: $NV * NK + 1$

1.3 Рассмотрим взвешенную сумму средних квадратов отклонений

$$\sum_{k=1}^{NK} \sum_{c=1}^{NC} w_c \left(1 - \frac{y_{c,k}}{\hat{y}_c} \right)^2 \rightarrow \min$$

\hat{y}_c - суммируем по координатам s и делим на NK - идеальное значение по характеристике c в подмножестве.

Чем больше w_c , тем больше значит тебя критерий равномерности - тем больше равны координаты векторов.

Такая запись нелинейна по y , то на дом будет модуль.

1-е ограничение нужно заменить на связь сигм с дельтами.

Усложним еще задачу.

1.4 Критерий равномерности

1. Общее число связей

2. Среднее рабочее время на бортопроводника - берем подмножество связей, попавших на рабочий стол - суммируем время.

Можем обобщить: что каждый вектор $S_{i,c,k}$ имеет разные координаты для разных подгрупп.

Приращение по характеристике c при добавлении i в k подгруппе.

1.5 Minimize Difference

ВХОДНЫЕ ДАННЫЕ:

1. S - множество векторов
2. NV - мощность множества S
3. NC - размерность вектора $s \in S$, то есть каждый вектор описывается NC числовыми координатами

$$C = \{1, \dots, NC\}$$

Множество S задаётся следующим образом:

$$S = \{s_{i,1}, s_{i,2}, \dots, s_{i,NC}\} \quad \forall i \in \{1, \dots, NV\}$$

4. NK - число групп:

$$K = \{1, \dots, NK\}$$

ДОПОЛНЕНИЕ К ВХОДНЫМ ДАННЫМ:

- Введём дополнительную переменную $y_{c,k}$ - суммарное значение координаты $c \in C$ для группы $k \in K$.

ЗАДАЧА: необходимо распределить векторы из S по NK группам, причём каждый вектор должен быть представлен в единственной группе.

ЦЕЛЕВАЯ ФУНКЦИЯ: минимизация максимальной разницы по модулю между двумя группами по координате среди всех координат и всех групп:

$$\max_{\substack{k_1, k_2 \in K \\ c \in C}} |y_{c,k_1} - y_{c,k_2}| \rightarrow \min$$

Пояснение: необходимо найти две группы k_1 и k_2 и такую координату c , которые бы минимизировали максимальную разность.

ОГРАНИЧЕНИЯ:

1. Каждый вектор должен находиться строго в одной из групп:

$$\sum_{k=1}^{NK} x_{s,k} = 1 : \forall s \in S$$

$$x_{s,k} \in \{0, 1\}$$

2. Сумма по каждой координате в каждой группе:

$$y_{c,k} = \sum_{s \in S} s_c \cdot x_{s,k} : \forall c \in C \ \forall k \in K$$

3. Введём переменную σ , являющуюся максимальной разностью по координате в группах. Её необходимо минимизировать:

$$\sigma \rightarrow \min$$

Введём ограничение, связывающую σ и исходную целевую функцию:

$$|y_{c,k_1} - y_{c,k_2}| \leq \sigma : \forall k_1, k_2 \in K \ \forall c \in C \ k_1 < k_2$$

Модуль раскрывается через два неравенства:

$$y_{c,k_1} - y_{c,k_2} - \sigma \leq 0$$

$$-y_{c,k_1} + y_{c,k_2} - \sigma \leq 0$$

Всего в задаче $NV \cdot NK + 1$ переменных.

1.6 Weighted Minimize

ВХОДНЫЕ ДАННЫЕ:

1. S - множество векторов
2. NV - мощность множества S
3. NC - размерность вектора $s \in S$, то есть каждый вектор описывается NC числовыми координатами

$$C = \{1, \dots, NC\}$$

Множество S задаётся следующим образом:

$$S = \{s_{i,1}, s_{i,2}, \dots, s_{i,NC}\} \quad \forall i \in \{1, \dots, NV\}$$

4. NK - число групп:

$$K = \{1, \dots, NK\}$$

ДОПОЛНЕНИЕ К ВХОДНЫМ ДАННЫМ:

- Введём дополнительную переменную $y_{c,k}$ - суммарное значение координаты $c \in C$ для группы $k \in K$.
- Введём дополнительные ИДЕАЛЬНЫЕ константные значения следующим образом:

$$\hat{y}_c = \frac{\sum_{i=1}^{NV} s_{i,c}}{NK} : \forall c \in C$$

- Введём константы весов, отвечающие за значимость уравнивания по определенной координате - чем больше значение веса, тем важнее уравнивать множество по данной координате:

$$W = \{w_1, \dots, w_{NC}\}$$

ЦЕЛЕВАЯ ФУНКЦИЯ: минимизация взвешенной (W) суммы модулей относительных отклонений $y_{c,k}$ от \hat{y}_c по каждой из координат для каждой группы с учётом весов W :

$$\sum_{c=1}^{NC} w_c \cdot \sum_{k=1}^{NK} \left| 1 - \frac{y_{c,k}}{\hat{y}_c} \right| \rightarrow \min$$

ОГРАНИЧЕНИЯ:

1. Каждый вектор должен находиться строго в одной из групп:

$$\sum_{k=1}^{NK} x_{s,k} = 1 : \forall s \in S$$

$$x_{s,k} \in \{0, 1\}$$

2. Сумма по каждой координате в каждой группе:

$$y_{c,k} = \sum_{s \in S} s_c \cdot x_{s,k} : \forall c \in C \ \forall k \in K$$

3. Введём $NC \cdot NK$ переменных $\sigma[c, k]$, являющуюся верхней границей максимальной величины отклонения. Будем минимизировать сумму всех этих переменных:

$$\sum_{c=1}^{NC} w_c \cdot \sum_{k=1}^{NK} \sigma[c, k] \rightarrow \min$$

Введём следующие ограничения для относительных отклонений:

$$\left| 1 - \frac{y_{c,k}}{\hat{y}_c} \right| \leq \sigma[c, k] : \forall c \in C \ \forall k \in K$$

Модуль раскрывается через два неравенства:

$$1 - \frac{y_{c,k}}{\hat{y}_c} - \sigma[c, k] \leq 0$$

$$-1 + \frac{y_{c,k}}{\hat{y}_c} - \sigma[c, k] \leq 0$$

Всего в задаче $NV \cdot NK + NC \cdot NK = NK(NV + NC)$ переменных.

1.7 Weighted Choose Minimize

ВХОДНЫЕ ДАННЫЕ:

1. S - множество векторов
2. NV - мощность множества S
3. NC - размерность вектора $s \in S$, то есть каждый вектор описывается NC числовыми координатами

$$C = \{1, \dots, NC\}$$

4. NK - число групп:

$$K = \{1, \dots, NK\}$$

Координаты векторов могут отличаться в зависимости от попадания в подмножество, поэтому множество S задаётся следующим образом:

$$S = \{s_{i,k,1}, s_{i,k,2}, \dots, s_{i,k,NC}\} \quad \forall i \in \{1, \dots, NV\} \quad \forall k \in K$$

ДОПОЛНЕНИЕ К ВХОДНЫМ ДАННЫМ:

- Введём дополнительную переменную $y_{c,k}$ - суммарное значение координаты $c \in C$ для группы $k \in K$.
- Введём дополнительные ИДЕАЛЬНЫЕ константные значения следующим образом:

$$\hat{y}_{c,k} = \frac{\sum_{i=1}^{NV} s_{i,k,c}}{NK} : \forall c \in C \quad \forall k \in K$$

- Введём константы весов, отвечающие за значимость уравнивания по определенной координате - чем больше значение веса, тем важнее уравнивать множество по данной координате:

$$W = \{w_1, \dots, w_{NC}\}$$

ЦЕЛЕВАЯ ФУНКЦИЯ: минимизация взвешенной (W) суммы модулей относительных отклонений $y_{c,k}$ от $\hat{y}_{c,k}$ по каждой из координат для каждой группы с учётом весов W :

$$\sum_{c=1}^{NC} w_c \cdot \sum_{k=1}^{NK} \left| 1 - \frac{y_{c,k}}{\hat{y}_{c,k}} \right| \rightarrow \min$$

ОГРАНИЧЕНИЯ.:

1. Каждый вектор должен находиться строго в одной из групп и вектор из группы возможных векторов в зависимости от номера группы должен быть тоже один.

$$\sum_{i=1}^{NK} \sum_{k=1}^{NK} x_{s,i,k} = 1 : \forall s \in S$$

$$x_{s,i,k} \in \{0, 1\}$$

Количество переменных: $NV \cdot NK^2$

2. Сумма по каждой координате в каждой группе:

$$y_{c,k} = \sum_{i=1}^{NK} \sum_{s \in S} s_{i,c} \cdot x_{s,i,k} : \forall c \in C \ \forall k \in K$$

3. Введём $NC \cdot NK$ переменных $\sigma[c, k]$, являющуюся верхней границей максимальной величины отклонения. Будем минимизировать сумму всех этих переменных:

$$\sum_{c=1}^{NC} w_c \cdot \sum_{k=1}^{NK} \sigma[c, k] \rightarrow \min$$

Введём следующие ограничения для относительных отклонений:

$$\left| 1 - \frac{y_{c,k}}{\hat{y}_{c,k}} \right| \leq \sigma[c, k] : \forall c \in C \ \forall k \in K$$

Модуль раскрывается через два неравенства:

$$1 - \frac{y_{c,k}}{\hat{y}_{c,k}} - \sigma[c, k] \leq 0$$

$$-1 + \frac{y_{c,k}}{\hat{y}_{c,k}} - \sigma[c, k] \leq 0$$

Всего в задаче $NV \cdot NK^2 + NC \cdot NK = NK \cdot (NV \cdot NK + NC)$ переменных.

1.8 Натурные данные 16.09.2020

ВХОДНЫЕ ДАННЫЕ:

Книга - подмножество бортипроводников. Подмножества связей должны быть примерно одинаковым.

Пусть M - количество связей, выданных на месяц.

$F = \{f_1, \dots, f_m\}$ - множество оборотных рейсов (оборотный рейс и связи - одно и то же).

$M' < M$ - число ночных связей - ночной связкой считается та связь, 50 процентов её рейсов относится с 22 : 00 до 6 : 00.

ХАРАКТЕРИСТИКИ СВЯЗКИ $f \in F$.

- t_f - время воздушного судна в воздухе - время налёта
- p_f - размер экипажа - сколько бортипроводников должно назначиться на связь (3, 4)
- v_f - тип сообщения (ВВЛ, МВЛ, СНГ) - внутренняя воздушная линия, международная воздушная линия, союз независимых государств
- U_1 - множество связей ВВЛ, u'_1 - множество ночных связей ВВЛ
- U_2 - множество связей МВЛ, u'_2 - множество ночных связей МВЛ
- U_3 - множество связей СНГ, u'_3 - множество ночных связей СНГ
 $U'_\alpha < U_\alpha, \forall \alpha \in \{1, 2, 3\}$
- $d_f \in T$ - дата вылета (дата начала связи) - множество дней горизонта планирования
- $a_f \in L$ - тип воздушного судна (ВС), на котором осуществляется связь, L - множество типов ВС
- $A_l, l \in L$ - множество связей с типом воздушного судна - l
- s_f - направление связи - тот город, куда направляется из Санкт-Петербурга $s_f \in R$, R - множество всех направлений
- D_i - множество связей с вылетом в день i

КНИГИ.:

$B = \{B_1, \dots, B_k\}$ - K подмножеств бортипроводников, B - множество книг, каждая книга характеризуется 3 характеристиками:

$$c_{\alpha,k}; \forall \alpha \in \{1, 2, 3\}; \forall k \in \{1, \dots, K\}$$

$c_{1,k}$ - число доступных бортипроводников в группе K , из множества ВВЛ для планирования

$c_{2,k}$ - число доступных бортипроводников в группе K , из множества МВЛ для планирования

$c_{3,k}$ - число доступных бортипроводников в группе K , из множества СНГ для планирования

Необходимо разбить подмножество связей на K подмножеств, с учётом критериев равномерности.

КРИТЕРИЙ РАВНОМЕРНОСТИ.

1. Средний налет на одного бортипроводника по типу сообщения (включает в себя 3 характеристики по ВВЛ, МВЛ, СНГ)

Пусть $\hat{y}_{j,k}$ - это усреднённое значение j -ой характеристики k -ой группы. Введем три идеальных значения:

$$\hat{y}_{j,k} = \frac{\sum_{i \in U_j} p_i \cdot t_i}{\sum_{k'=1}^K c_{j,k'}}; \forall j \in \{1, 2, 3\}; \forall k \in \{1, \dots, K\}$$

равен сумме налета по каждой связке из соответствующего типа сообщения, деленное на суммарное число бортипроводников по каждому типу сообщения

2. Средний ночной налёт на одного бортипроводника по типу сообщения:

$$\hat{y}_{3+j,k} = \frac{\sum_{i \in U'_j} p_i \cdot t_i}{\sum_{k'=1}^K c_{j,k'}}; \forall j \in \{1, 2, 3\}; \forall k \in \{1, \dots, K\}$$

3. Общее число ночных связей на рабочих столах (РС) пропорционально числу бортипроводников на рабочем столе - чем больше бп на рабочем столе, тем больше ночных связей на рабочем столе:

$$\hat{y}_{7,k} = \frac{c_{1,k} \cdot M'}{\sum_{k'=1}^K c_{j,k'}}; \forall k \in \{1, \dots, K\}$$

4. Общее число связей по типу сообщения для рабочего стола (РС) должно быть пропорционально числу бортипроводников, доступных для планирования по типу сообщения:

$$\hat{y}_{7+j,k} = \frac{c_{j,k} \cdot |U_j|}{\sum_{k'=1}^K c_{j,k'}}; \forall j \in \{1, 2, 3\}; \forall k \in \{1, \dots, K\}$$

5. Равенство общего количества связей в день для рабочего стола (РС) (каждый день эксперты должны следить за примерно одинаковым количеством бортипроводником и не было перегруза в сторону какого-то эксперта):

$$\hat{y}_{10+j,k} = \frac{|D_j|}{K}; \forall j \in \{1, \dots, |T|\}; \forall k \in \{1, \dots, K\}$$

6. Общее количество связей по типу воздушных судов (ВС) для рабочего стола (РС):

$$\hat{y}_{10+|T|+j,k} = \frac{|A_j|}{K}; \forall j \in \{1, \dots, |L|\}; \forall k \in \{1, \dots, K\}$$

7. Общее количество связей по направлению для РС. S_i - множество связей по направлению:

$$\hat{y}_{10+|T|+|L|+j,k} = \frac{|S_j|}{K}; \forall j \in \{1, \dots, |R|\}; \forall k \in \{1, \dots, K\}$$

Размерность идеального вектора для группы:

$$N = 10 + |T| + |L| + |R|$$

Для любой связи $f \in F$ введем следующие матрицы:

$$\Delta f = \{\delta_{f,j,k}\}_{j \in \{1, \dots, N\}; k \in \{1, \dots, K\}}$$

где $\delta_{f,j,k}$ - приращение j -ой характеристики при распределении f -ой связи в группу k - описание вектора для описания в предыдущей задаче.

$\Delta = \{\Delta_1, \Delta_2, \dots, \Delta_m\}$ - тензор приращений.

$\Delta_{f,k}$ - вектор (столбец) значений приращений при добавлении f -ой связи в k -ую подгруппу.

Целевая функция.

$$\sum_{k=1}^K \sum_{j=1}^N w_j \left(1 - \frac{y_{j,k}}{\hat{y}_{j,k}}\right)^2 \rightarrow \min$$

Веса характеристик, принадлежащим одному критерию, равны (веса для первых трех характеристик равны). 7 критериев, описывается 3-мя характеристиками, веса для этих характеристик равны.

Решение задачи.

2135 связей и 6 групп, то мы не можем решить целочисленным программированием. Решим задачу с помощью жадного алгоритма:

Алгоритм 1.

Пусть L - функция работает от k аргументов. Каждый аргумент описывает характеристики k -го подмножества $j = \{1, \dots, N\}$

$$L(y_1, \dots, y_K) = \sum_{k=1}^K \sum_{j=1}^N w_j \cdot \left(1 - \frac{y_{j,k}}{\hat{y}_{j,k}}\right)^2$$

где $y_i = (y_{1,i}, \dots, y_{N,i})$.

На шаге СТЕР мы ищем такую пару k и f , чтобы минимизировать значение целевой функции. На каждом шаге распределяем одну связь в

одно из подмножеств.

$$L_k(y_1, \dots, y_K, f) = L(y'_1, y'_2, \dots, y'_k)$$

где

$$y'_i = \begin{cases} y_i, & \text{if } i \neq k \\ y_i + \Delta_{f,i} & \end{cases}; \forall i \in \{1, K\}$$

будет искать такую связку, при котором значения функции минимально.

Введем $y_k = (0, \dots, 0)$, $k \in \{1, \dots, K\}$:

Пока $F \neq 0$ ищем $(k, f) = \underset{k \in \{1, \dots, K\}, f \in F}{\operatorname{argmin}} L_k(y_1, \dots, y_K, f)$

Мы нашли связку f маленькую, поэтому мы можем вычеркнуть ее из связок:

$$F = F \setminus \{f\}$$

$$y_k = y_k + \Delta_{f,k}$$

и сохраняет, что f в k . Цикл заканчивается и выдается распределение.

- Выбираем $K \cdot M$ вариантов - в какую группу положить связку
- Выбираем $K \cdot (M - 1)$

Алгоритм 2. С сортировкой связок nb document.

1. Импорт исходных файлов + эксперт
2. Позволяет вводить веса критериев
3. Позволяет проводить расчеты по A1 и A2 (2 типа сортировки)
4. Создавать отчет по результатам по результате работы алгоритма
 - Средний налет на бортпроводника - Последняя строка - максимальная разность по модулю между значениями в столбце - максимум по минимум
5. Функционал для сравнения результатов работ алгоритмов + решение эксперта

1.9 Метод Сугиямы

- 1) Поиск такого максимального ациклического подграфа. Дан $G = (V, A) \rightarrow G' \leq G : v' = (V, A'), |A'| \rightarrow \max$
- 2) Минимальный feedback arc set:

$$G = (V, A) \rightarrow FASCA, G'' = (V, A \setminus FAS)$$

3) Минимальный Feedback Set:

$$G = (V, A) \rightarrow FS \subset A, G'' = (V, (A \setminus FS) \cap rev(FS))$$

ациклическое, $|FS| \rightarrow \min$

1.10 Set Covering Formulation

Дан ориентированный граф. Нужно минимальный взвешенный Feedback Set, максимальный вес ациклического подграфа.

$$y_{i,j} = \begin{cases} 1, (i,j) \in FAS \\ 0, \text{если не удаляем} \end{cases}$$

Матрица $M(c \times n)$, где $m_{i,j} = 1$, если дуга под номером j в цикле i .

Мощность $|A| = n$, C - количество циклов в графе. Использовать FindCycles

$$\sum_{(i,j) \in A} w_{i,j} \cdot y_{i,j} \rightarrow \min$$

Из цикла нужно удалить как минимум одну дугу. Мы проходимся по всем циклам от $\forall i \in \{1, \dots, c\}$

$$\sum_{j=1}^n m_{i,j} y_{A(j)} \geq 1$$

где $A(j)$ - это j -ая дуга из A .

Формируем количество переменных y по количеству дуг. Формируем матрицу M . Результатом является минимальный FAS \rightarrow FS.

1.11 Неравенство Треугольника

Найти такой порядок вершин, чтобы при линейной укладке как можно меньше дуг смотрело справа налево, суммарный вес дуг наименьший. Дуги с наименьшим весом - менее важны.

Пусть дан граф $G = (V, E)$, n - количество рёбер: $\#E$

Пусть π - перестановка в лексикографическом порядке вершин (по возрастанию от $1, \dots, n$). Сформируем матрицу $C \in \mathbb{R}^{n \times n}$, элементами которого будут $c_{i,j}$, такие, что:

$$c_{i,j} = \begin{cases} w_{i,j}, & (i,j) \in A \\ 0 & \end{cases}$$

что означает, что мы заполняем вес ребра в матрицу, если оно есть в рёбрах графа.

Введём переменные задачи:

$$x_{i,j} = \begin{cases} 1, & \pi^*(i) < \pi^*(j) : \forall i, j \in V, \pi(i) < \pi(j) \\ 0 & \end{cases}$$

что означает, что мы будем брать те переменные, которые равны 1, то есть стоят левее в линейной укладке. Всего у нас порядка $O(n^2)$ переменных, а именно $\frac{n(n-1)}{2}$.

Целевая функция задачи разбивается на две суммы:

$$\sum_{\substack{j,i \in E \\ \pi(j) > \pi(i)}} c_{i,j} x_{i,j} + \sum_{\substack{i,j \in E \\ \pi(i) < \pi(j)}} c_{i,j} (1 - x_{i,j}) \rightarrow \min$$

что означает, что в первой сумме суммируются все дуги у которых первая вершина больше второй в лексикографическом порядке, а во второй - те дуги, у которых первая вершина меньше второй в лексикографическом порядке.

Ограничения:

$$0 \leq x_{i,j} + x_{j,k} - x_{i,k} \leq 1, \forall i, j, k \in V, \pi(i) < \pi(j) < \pi(k)$$

$$x_{i,j} \in \{0, 1\}, 1 \leq i, j \leq n$$

1.12 Layering

Пусть дан ациклический ориентированный граф $G = (V, A)$, необходимо разбить на слои V_1, V_2, \dots, V_k , чтобы $\forall (u, v) \in A : u \in V_i, v \in V_j, i < j$. Введём переменные $\lambda(u)$. Целевая функция:

$$\sum_{(u,v) \in A} (\lambda(v) - \lambda(u)) \rightarrow \min$$

Ограничения:

$$\lambda(v) - \lambda(u) \geq 1 \quad \forall (u, v) \in A$$

$$\lambda(v) \geq 1 \quad \forall v \in V$$

$$\lambda(u) \in \mathbb{Z} \quad \forall u \in V$$

1.13 Распределение вершин по слоям с ограничением на ширину слоя

Пусть дан $G(V, A)$ - DAG (Directed Acyclic Graph). Решается задача распределения по слоям. Введём следующие переменные:

$$X_{i,k} = \begin{cases} 1, & \text{если вершина } i \text{ назначена на слой } k \\ 0, & \text{в обратном случае} \end{cases}$$

Обозначим за W - максимальную ширину слоя, за n_v - число вершин в графе. Тогда ограничение:

$$\sum_{i=1}^{n_v} X_{i,k} \leq W \quad \forall k \in \{1, \dots, h_{\max}\}$$

где h_{\max} - длина наибольшего пути в графе. Данное ограничение означает, что количество вершин на каждом слое не может превышать W .

Для каждой вершины $i \in V$ определим минимальный и максимальный номер слоя, на котором может располагаться вершина i :

$$[\underline{h}_i, \overline{h}_i] \quad \forall i \in V$$

тогда количество переменных уменьшится:

$$X_{i,k} = \begin{cases} 1, & \text{если вершина } i \text{ назначена на слой } k \\ 0, & \text{в обратном случае} \end{cases}, \underline{h}_i \leq k \leq \overline{h}_i$$

$$X_{i,k} \in \{0, 1\}$$

$$\forall i \in V \quad \forall k \in \{\underline{h}_i, \overline{h}_i\}$$

Номер слоя вершины $\forall i \in V$ вычисляется следующим образом:

$$\lambda(i) = \sum_{k=\underline{h}_i}^{\overline{h}_i} k \cdot X_{i,k}$$

Ограничение: каждая вершина назначается ровно на один слой:

$$\sum_{k=\underline{h}_i}^{\overline{h}_i} X_{i,k} = 1$$

Ограничение, что направление дуг - сверху вниз:

$$\lambda(v) - \lambda(u) \geq 1 \quad \forall (u, v) \in A$$

$$\sum_{k=\underline{h}_i}^{\overline{h}_i} k \cdot X_{v,k} - \sum_{k=\underline{h}_i}^{\overline{h}_i} k \cdot X_{u,k} \geq 1 \quad \forall (u, v) \in A$$

Теперь переходим к главному: целевая функция. Необходимо минимизировать число фиктивных вершин. Для этого воспользуемся логическим рассуждением: чем меньше слоёв нам потребуется для расстановки вершин по слоям, тем меньшее количество фиктивных вершин будет в Layering графе. Поэтому введём переменную Φ и будем её минимизировать при дополнительном ограничении:

$$\Phi \rightarrow \min$$

$$\lambda(i) = \sum_{k=\underline{h_i}}^{\overline{h_i}} k \cdot X_{i,k} \leq \Phi \quad \forall i \in V$$

Если W - переменная, то введём веса и целевая функция примет вид:

$$w_1 \cdot \Phi + w_2 \cdot H \rightarrow \min$$

1.14 Гиперграф

Для начала разберёмся, что такое гиперребро. Гиперребро задаётся с помощью двух понятий: ИСТОЧНИК и МНОЖЕСТВО СТОКОВ - вершин, соединённых с источником. Так же у нас есть понятие ГОРИЗОНТАЛЬНОГО СЕГМЕНТА - прямая, соединяющая стоки.

Далее у нас есть 3 вида пересечений:

1. горизонтальный сегмент пересекает линию от источника
2. горизонтальный сегмент пересекает сток
3. стоки накладываются друг на друга

Все это представлено на картинке ниже:

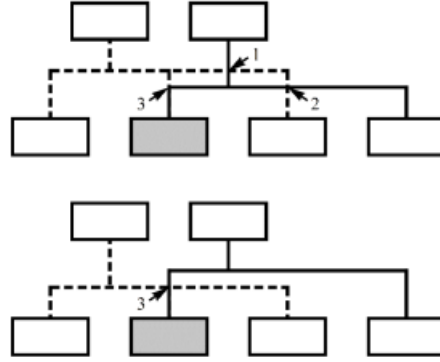


Рис.1 Три вида пересечений

Дан двухслойный гиперграф $H_2 = (V_1, V_2, E_{H_2})$, где $V_1 = \{u_1, \dots, u_{N_1}\}$ и $V_2 = \{v_1, \dots, v_{N_2}\}$ - множество вершин верхнего и нижнего слоя, а E_{H_2} - множество одноисточных рёбер. Для каждой вершины $v \in V_1 \cup V_2$ двух-слойного гиперграфа известны координаты $x(v), y(v)$.

Алгоритмы минимизации числа пересечений гиперрёбер

Введём 8 видов целочисленных переменных:

1. Переменные Z'_i и Z''_j характеризуют количество единичных сдвигов вершин верхнего $u_i \in V_1$ и нижнего слоя $v_j \in V_2$, знаки определяют направление сдвигов.

Новые абсциссы вершин $u_i \in V_1, v_j \in V_2$ будут равны:

$$x'(u_i) = x(u_i) + \Delta \times Z'_i \quad (1.1.1)$$

$$x'(v_j) = x(v_j) + \Delta \times Z''_j \quad (1.1.2)$$

2. Переменные AZ'_i и AZ''_j - это количество единичных сдвигов вершин u_i и v_j , соответственно, то есть:

$$AZ'_i = |Z'_i| \quad (1.2.1)$$

$$AZ_j'' = |Z_j''| \quad (1.2.2)$$

3. Для каждой пары гиперрёбер (e_n, e_m) , где $e_n = (u_\eta, T_\eta)$, $e_m = (u_\mu, T_\mu)$ и первый источник находится левее второго источника гиперребра $(x(u_\eta) < x(u_\mu))$ задаются следующие переменные:

$$HH_{n,m} = \begin{cases} 0, & \text{если горизонтальный сегмент } e_m \text{ **выше**, чем } e_n \\ 1, & \text{если горизонтальный сегмент } e_m \text{ **ниже**, чем } e_n \end{cases} \quad (1.3.1)$$

$$CT1_{n,m} = \begin{cases} 0, & \text{горизонтальный сегмент **не пересекает** линию от источника} \\ 1, & \text{горизонтальный сегмент **пересекает** линию от источника} \end{cases} \quad (1.3.2)$$

$$CT2_{n,m} = \begin{cases} c_{e_n, e_m}, & \text{если горизонтальный сегмент } e_m \text{ **выше**, чем } e_n \\ c_{e_m, e_n}, & \text{если горизонтальный сегмент } e_m \text{ **ниже**, чем } e_n \end{cases} \quad (1.3.3)$$

где c_{e_n, e_m} - количество пересечений второго типа между гиперрёбрами e_n и e_m в зависимости от взаимного расположения горизонтальных сегментов.

4. Переменные $BV_{\eta,j}$ - это булевы фиктивные переменные, которые сводятся для каждого гиперребра $e_n = (u_\eta, T_\eta)$ и для таких вершин v_j нижнего слоя, не смежных вершине u_η .

Целевая функция минимизации пересечений гиперрёбер

Выбор целевой функции задачи целочисленного программирования обусловлен целями:

- минимизация пересечений первого и второго типа
- минимизация сдвигов вершин

В терминах нашей задачи:

- Количество единичных сдвигов для вершин первого уровня должно быть как можно меньше

$$\sum_{i=1}^{|V_1|} AZ_i'$$

- Количество единичных сдвигов для вершин второго уровня должно быть как можно меньше

$$\sum_{j=1}^{|V_2|} AZ_j''$$

- Число пересечений и второго типа:

$$\sum_{n=1}^{|E_{H_2}|-1} \sum_{m=n+1}^{|E_{H_2}|} CT1_{n,m} + CT2_{n,m}$$

Очевидно, что все эти критерии берутся с какими-то весами, поэтому

целевая функция будет выглядеть следующим образом:

$$L = w_z \cdot \left(\sum_{i=1}^{|V_1|} AZ'_i + \sum_{j=1}^{|V_2|} AZ''_j \right) + \sum_{n=1}^{|E_{H_2}|-1} \sum_{m=n+1}^{|E_{H_2}|} w_{c_1} \cdot CT1_{n,m} + w_{c_2} \cdot CT2_{n,m} \rightarrow \min \quad (1)$$

Теперь запишем ограничения:

ОГРАНИЧЕНИЕ 2-3:

$$\begin{aligned} CT1_{n,m} &\geq \frac{x(u_\eta) + \Delta Z'_\eta - (x(v_j) + \Delta Z''_j(v_j))}{K} - K HH_{n,m}, \\ e_n &= (u_\eta, T_\eta) \in E_{H_2}, \quad e_m = (u_\mu, T_\mu) \in E_{H_2}, \\ x(u_\eta) &< x(u_\mu), \quad v_j \in T_\mu, \quad x(v_j) = \min_{v \in T_\mu} x(v) \end{aligned} \quad (2)$$

$$\begin{aligned} CT1_{n,m} &\geq \frac{x(v_j) + \Delta Z''_j(v_j) - (x(u_\mu) + \Delta Z'_\mu)}{K} - K(1 - HH_{n,m}), \\ e_n &= (u_\eta, T_\eta) \in E_{H_2}, \quad e_m = (u_\mu, T_\mu) \in E_{H_2}, \\ x(u_\eta) &< x(u_\mu), \quad v_j \in T_\eta, \quad x(v_j) = \max_{v \in T_\eta} x(v) \end{aligned} \quad (3)$$

Объяснение: было объяснено на паре, но суть следующая: $x(v_j) = \min_{v \in T_\mu} x(v)$ означает, что данная вершина наиболее левая вершина для первого слоя, а в третьем ограничении - что наиболее правая для второго слоя. Тогда если горизонтальный сегмент e_m **выше**, чем e_n , то $HH_{n,m} = 0$, то второе утверждение говорит нам, что вершина u_θ имеет новые координаты абсцисс левее, чем вершина v_j . При подстановке $HH_{n,m} = 1$ все в точности наоборот.

ОГРАНИЧЕНИЕ 4-5:

$$\begin{aligned} CT2_{n,m} &\geq c_{e_n, e_m} - K(1 - HH_{n,m}), \\ e_n &= (u_\eta, T_\eta) \in E_{H_2}, \quad e_m = (u_\mu, T_\mu) \in E_{H_2}, \quad x(u_\eta) < x(u_\mu) \end{aligned} \quad (4)$$

$$\begin{aligned} CT2_{n,m} &\geq c_{e_m, e_n} - K HH_{n,m}, \\ e_n &= (u_\eta, T_\eta) \in E_{H_2}, \quad e_m = (u_\mu, T_\mu) \in E_{H_2}, \quad x(u_\eta) < x(u_\mu) \end{aligned} \quad (5)$$

Объяснение: если горизонтальный сегмент e_m **выше**, чем e_n , то $HH_{n,m} = 0$, $CT2 = c_{e_n, e_m}$, ограничение 4 принимает вид: $0 \geq -K$, а (5) - $CT2_{n,m} \geq c_{e_m, e_n}$ - устанавливаем ограничение на число пересечений второго типа а при обратном $HH_{n,m} = 1$ и 4-е ограничение просто выполняется всегда.

ОГРАНИЧЕНИЕ 6-9:

$$\begin{aligned}
x(v_j) + \Delta Z_j''(v_j) &\leq x(u_\eta) + \Delta Z_\eta' + K BV_{\eta,j} + K HH_{n,m} - \varepsilon, \\
e_n &= (u_\eta, T_\eta) \in E_{H_2}, \quad e_m = (u_\mu, T_\mu) \in E_{H_2},
\end{aligned} \tag{6}$$

$$\begin{aligned}
x(u_\eta) &< x(u_\mu), \quad v_j \in T_\mu \\
x(v_j) + \Delta Z_j''(v_j) &\leq x(u_\mu) + \Delta Z_\mu' + K BV_{\mu,j} + K(1 - HH_{n,m}) - \varepsilon, \\
e_n &= (u_\eta, T_\eta) \in E_{H_2}, \quad e_m = (u_\mu, T_\mu) \in E_{H_2},
\end{aligned} \tag{7}$$

$$\begin{aligned}
x(u_\eta) + \Delta Z_\eta' &\leq x(v_j) + \Delta Z_j''(v_j) + K(1 - BV_{\eta,j}) + K HH_{n,m} - \varepsilon, \\
e_n &= (u_\eta, T_\eta) \in E_{H_2}, \quad e_m = (u_\mu, T_\mu) \in E_{H_2}, \\
x(u_\eta) &< x(u_\mu), \quad v_j \in T_\mu
\end{aligned} \tag{8}$$

$$\begin{aligned}
x(u_\mu) + \Delta Z_\mu' &\leq x(v_j) + \Delta Z_j''(v_j) + K(1 - BV_{\mu,j}) + K(1 - HH_{n,m}) - \varepsilon, \\
e_n &= (u_\eta, T_\eta) \in E_{H_2}, \quad e_m = (u_\mu, T_\mu) \in E_{H_2}, \\
x(u_\eta) &< x(u_\mu), \quad v_j \in T_\eta
\end{aligned} \tag{9}$$

Объяснение: в данных ограничениях у нас есть координаты абсцисс новых вершин v_j и μ_η . При подстановке различных значений переменных $HH_{n,m}$, если горизонтальный сегмент e_m **выше**, чем e_n , то мы пытаемся задать расположение рёбер и всё зависит от булевых фиктивных вершин, не смежных u_η . Если фиктивная переменная равна единице, то для ограничения 6 это означает, что v_j находится правее v_j , так как мы должны добавить какую-то константу K , а если 0, то левее. Далее с другими ограничениями аналогично. Но вообще я запутался с этим, мне не очень понятно. С переменными понимаю вроде смысл, а ограничений нет.

ОГРАНИЧЕНИЕ 10-11:

$$AZ_i' \geq Z_i', AZ_i' \geq -Z_i', \quad u_i \in V_1 \tag{10}$$

$$AZ_j'' \geq Z_j'', AZ_j'' \geq -Z_j'', \quad v_j \in V_2 \tag{11}$$

Объяснение: это просто линеаризация модуля из формул (1.2.1) и (1.2.2) и для целевой функции (1).

ОГРАНИЧЕНИЕ 12-13:

$$x(u_{i+1}) + \Delta Z_{i+1}' - (x(u_i) + \Delta Z_i') \geq \delta_w, \quad i = 1, \dots, |V_1| - 1 \tag{12}$$

$$x(v_{j+1}) + \Delta Z_{j+1}'' - (x(v_j) + \Delta Z_j'') \geq \delta_w, \quad j = 1, \dots, |V_2| - 1 \tag{13}$$

Объяснение: для каждой вершины разница в новых абсциссах между правым её соседом и самой вершиной на первом слое больше какого-то значения δ_w . То же самое и для вершин второго слоя.

ОГРАНИЧЕНИЕ 14:

$$\begin{aligned}
0 &\leq HH_{l,m} - HH_{l,n} + HH_{n,m} \leq 1, \\
e_l &= (u_\lambda, T_\lambda) \in E_{H_2}, \quad e_m = (u_\mu, T_\mu) \in E_{H_2}, \quad e_n = (u_\eta, T_\eta) \in E_{H_2}, \\
x(u_\lambda) &< x(u_\mu) < x(u_\eta)
\end{aligned} \tag{14}$$

Объяснение: пусть у нас есть 3 гиперребра и первое гиперребро левее всех остальных ребер, а второе левее третьего. Тогда по сути это неравенство треугольника для горизонтальных сегментов в высоту - задает порядок для каждой тройки гиперрёбер в высоту: либо горизонтальный сегмент выше, чем другие и тогда все нули, либо выше тогда, все единицы и т.д.

Алгоритм решения задачи минимизации числа пересечений гиперребер через задачу поиска линейной укладки

Одним из подходов к решению задачи минимизации числа пересечений гиперребер является ее формулировка в виде задачи устранения циклов в некотором взвешенном направленном графе. Каждая вершина этого графа соответствует горизонтальному сегменту гиперребра, и каждая пара вершин соединена двумя противоположенными взвешенными дугами. Направление дуги показывает, какая вершина (т.е. соответствующий ей горизонтальный сегмент) расположена выше, а ее вес определяется взвешенной суммой числа пересечений и недопустимых наложений гиперребер, соответствующих вершинам.

Буду ссылаться на алгоритм неравенства треугольника из пункта 1.11 в линейной укладке графа, который мы проходили со страницы 19.

Сформируем матрицу $C \in R^{n \times n}$, элементами которого будут $c_{i,j}$, такие, что:

$$\begin{cases} w_{i,j}, (i,j) \in E \\ 0 \end{cases}$$

где $w_{i,j}$ - суммарное количество пересечений первого второго и третьего типа и 0, если данная дуга не может быть проведена. Вводим переменные задачи:

$$x_{i,j} = \begin{cases} 1, i \mapsto j \text{ were selected} \\ 0 \end{cases}$$

По аналогии с линейной укладкой графа из 1.11 целевая функция должна разбиваться на две суммы, но я не понимаю каких:

$$\sum_{i,j} c_{i,j} \cdot x_{i,j} + \sum_{i,j} c_{i,j} (1 - x_{i,j}) \rightarrow \min$$

И добавляем ограничения и неравенства треугольник (опять же из того алгоритма):

$$0 \leq x_{i,j} + x_{j,k} - x_{i,k} \leq 1 \quad \forall i, j, k \in V$$

$$x_{i,j} \in \{0, 1\}, 1 \leq i, j \leq n$$

2 Построение расписания для цеха

3 Задача маршрутизации транспорта с ограничением на грузоподъёмность

Условие задачи:

Пусть задан полный неориентированный граф $G = (V, E)$, в котором $V = \{0, 1, \dots, n\}$ - множество всех вершин, 0 - начальная точка маршрута, в котором пути будут начинаться и заканчиваться. Расстояние между клиентами (стоимость проезда) между вершинами (клиентами) i, j обозначим за w_{ij} (причём $w_{ij} = w_{ji}$). Далее у нас есть некое множество *транспортных средств* мощностью T , у каждого из средств есть собственная грузоподъёмность c_j (*capacity*). Каждая вершина (клиент) выдвигает свой целочисленный спрос d_i (*demand*). Клиент обслуживается только одним транспортным средством. Транспортное средство не может обслужить множество клиентов, чей спрос превышает вместимость транспортного средства.

Цель: Необходимо посетить каждого клиента ровно 1 раз и построить маршруты с минимальным суммарным расстоянием, которые начинаются и заканчиваются в начальной вершине 0 и каждая вершина должна быть включена в маршрут только одного транспортного средства (кроме вершины 0).

Постановка задачи: целочисленное программирование:

В качестве переменных возьмём переменные отвечающие за то, было ли ребро (i, j) включено в итоговый маршрут:

$$x_{i,j} = \begin{cases} 1, \text{ ребро } (i, j) \text{ включено в маршрут} \\ 0, \text{ иначе} \end{cases}$$

Целевая функция - функция, минимизирующая суммарную длину маршрутов:

$$\sum_{i=0}^n \sum_{j=0}^n x_{ij} w_{ij} \rightarrow \min, i \neq j$$

Теперь запишем ограничения на въезд и выезд вершин:

$$\sum_{i=0}^n x_{ij} = 1 \quad \forall j \in \{1, \dots, n\}, i \neq j$$
$$\sum_{j=0}^n x_{ij} = 1 \quad \forall i \in \{1, \dots, n\}, i \neq j$$

Ограничения на выезд из начальной вершины:

$$\sum_{i=1}^n x_{0i} = T$$

Нужно ввести ограничение на веса..

$$\begin{aligned}
\lim_{x \rightarrow \infty} (2x-7)(\ln(3x+4)-\ln(3x)) &= \lim_{x \rightarrow \infty} (2x-7) \ln \left(\frac{3x+4}{3x} \right) = \lim_{x \rightarrow \infty} \ln \left(\left(1 + \frac{4}{3x} \right)^{2x} \right) = \\
&= \lim_{x \rightarrow \infty} \ln \left(\left(1 + \frac{4 \cdot 2}{3x} \right)^x \right) = \ln e^{\frac{8}{3}} = \frac{8}{3}
\end{aligned}$$