Планирование расписаний и управление доходами

Александр Широков ПМ-1701

Преподаватель:

Васильев Юрий Михайлович

Санкт-Петербург
2020 г., 7 семестр

Список литературы

[1]

Содержание

1 02.09.2020

1.1 Задача из авиакомпании Россия

В задачах планирования авиаперелетов:

- расписание судов
- маршутизация
- построение графика полета летного состава

Мы поговорим о построении графика полета летного состава. Зарплата бортпроводника зависит от навыков и от некоторыз 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 борпроводников разбито на 4 подмножеств с примерно одинаковыми характеристиками. Каждое подмножество называется **книга**.

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

Множесто 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 $S = \{s_1, \ldots, s_n\}$ на два подмножества, каждое подмножество характерирузет 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, кладём 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 уровень - число из сортированного 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 заменяется на их разницу и кладём 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 межде 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 $S = \{s_1, \ldots, s_n\}$ 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

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

$$Z - W \to \min$$

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

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

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

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

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

Жадный алгоритм

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

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

На каждом шаге алгоритма мы ищем такое $j \in \{1, \ldots, k\}$ такое, что значение целевой функции $gr = argminL_j$ 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 - $lbound, ubound$. Последний - какому множеству чисел принадлежит тип.

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

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

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

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

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

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

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

- Модуль со знаком $\geq$
- IF

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

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

и мы превратили в третий пункт

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

- Умножение бинарных переменных

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

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

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

$$y \leq x_1$$

$$y \leq x_2$$

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

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.

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

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

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

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

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

Множество:

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

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:

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

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

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

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

Так мы делаем для:

1. $$\sigma \geq y_{c,k_1} - y_{c,k_2} \quad \forall k_1, k_2 \in \{1, \ldots, NK\}, k_1 < k_2$$
$$\sigma \geq -y_{c,k_1} + y_{c,k_2} \quad \forall k_1, k_2 \in \{1, \ldots, NK\}, k_1 < k_2$$
$$\sigma \to \min$$

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

3. $$y_{c,k} = \sum_{s \in S} S_c \cdot x_{s,k} \forall c \in \{1, \ldots, NC\}, k \in \{1, \ldots, NK\}$$
$$x_{s,k} \in \{0, 1\} \forall s \in S, \forall k \in \{1, \ldots, NK\}$$

Всего незивестынх: $NV * NK + 1$

1.3 Рассмотрим взвешенную сумму средних квадратов отклонений

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

$\hat{y_c}$ - суммируем покоординатно c и делим на NK - идеальное значение по характеристике c в подмножестве.

Чем больше w_c, тем больше значит тебя критерий равномерности - тем больше равны координатым 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 ночных связок
2. Среднее рабочее время на бортпроводника - берем подмножество связок, 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 Differencse

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 максимальную разность по координате в группах. Её необходимо минимизировать:

$$\sigma \to \min$$

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 расскрывается 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

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

Множество S задаётся следующим образом:

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

4. NK - число групп:

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

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:

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

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

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

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:

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

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:

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

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 расскрывается 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

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

4. NK - число групп:

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

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

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

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:

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

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

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

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:

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

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:

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

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 расскрывается 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 бортпроводников. Подмножества связок должны быть примерно одинаковым.

Пусть M - количество связок, выданных на месяц.

$F = \{f_1, \ldots, f_m\}$ - множество оборотных рейство (оборотный рейс и связки - одно и тоже).

$M' < M$ - число ночных связок - ночной связкой считается та связка, 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 борпроводников должно назначиться на связку $(3, 4)$
- v_f - тип сообщения (ВВЛ, МВЛ, СНГ) - внутренняя воздушая линия, международная воздушная линия, союз независимых государств
- U_1 - множество связок ВВЛ, u'_1 - множество ночных связок ВВЛ
- U_2 - множество связок МВЛ, u'_2 - множество ночных связок МВЛ
- U_3 - множество связок СНГ, u'_3 - множество ночных связок СНГ
  $U'_\alpha < U_\alpha, \forall\alpha \in \{1, 2, 3\}$
- $d_f \in T$ - дата вылета (дата начала связки) - множество дней горизонта планирования
- $a_f \in L$ - тип воздушего судна (ВС), на котором осуществляется связка, L - множество типов ВС
- $A_l, l \in L$ - множество связок с типом воздушного судна - l
- s_f - направление связки - тот город, куда направляется из Санкт-Петербурга $s_f \in R$, R - множество всех направлений
- D_i - множесво связок с вылетом в день i

Книги.:

$B = \{B_1, \ldots, B_k\}$ - K подмножеств бортпроводников, B - множество книг, каждая книга характеризуется 3 харакстристиками:

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

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

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

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

Необходим разбить подмножество связок на K подмножеств, с учётом критериев равномерности.

Критерий равномерности.

1. Средний налет на одного бортпроводника 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:

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

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

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

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

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

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

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

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

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

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

6. Общее количество связок по типу воздуных судов (ВС) для рабочего стола (РС):

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

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

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

Размерность идеального вектора для группы:

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

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

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

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

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

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

Целевая функция.

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

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 связок 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 $j = \{1, \ldots, N\}$

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

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

На шаге STEP мы ищем такую пару k и f, чтобы минимизировать значение целевой функции. На каждом шаге распределяем одну связку 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 значении 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:

$$F = F\{1\}$$

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

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 А1 и А2 (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 максмимального аиклического подграфа. Дан $G = (V, A) \to G' \leq G : v' = (V, A')$, $|A'| \to \max$

2) Минимальный feedback arc set:

$$G = (V, A) \to FASCA, G'' = (V, A \backslash FAS)$$

3) Минимальный Feedback Set:

$$G = (V, A) \to FS \subset A, G'' = (V, (A \backslash FS) \cap rev(FS))$$

ацикличнское, $|FS| \to \min$

1.10 Set Covering Formulation

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

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

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 yпо количеству дуг. Формируем матрицу M. Результатом является минимальный FAS -> 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 $1, \ldots, n$). 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:

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

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

Ограничения:

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

$$x_{i,j} = \{0, 1\}, 1 \leqslant i, j \leqslant n$$

1.12 Layering

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

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

Ограничения:

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

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

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

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:

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

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

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

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:

$$X_{i,k} = \begin{cases} 1, \text{если вершина } \mathbf{i} \text{ назначена на слой } \mathbf{k} \\ 0, \text{в обратном случае} \end{cases} \quad , \underline{h_i} \leqslant k \leqslant \overline{h_i}$$

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

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

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:

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

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

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:

$$\Phi \to \min$$

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

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

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