

Комбинаторная оптимизация

Задача об открытии складов

5 июня 2019 г.

1 Формулировка

Межгалактическая торговая корпорация Vozon проводит конкурс по оптимизации расходов. Для обслуживания своих клиентов компания содержит n складов на разных планетах. Каждый склад характеризуется своей *вместимостью* $capacity_w$, а также *стоимостью содержания* $openCost_w$.

Кроме того, у компании есть m постоянных клиентов, которым надо поставлять товары со складов. Каждый клиент характеризуется величиной своего *потребления* $demand_c$.

Каждый клиент может потенциально обслуживаться несколькими складами одновременно. В такой ситуации потребление клиентом ресурсов складов делится в какой-то пропорции между несколькими складами. Формально говоря, назовём планом потребления X такой набор неотрицательных вещественных чисел $x_{w,c}$ по всем складам w и клиентам c , что $\sum_w x_{w,c} = 1$, а также, что $\sum_c x_{w,c} demand_c \leq capacity_w$ для любого склада w (потребление ресурсов склада не должно превосходить вместимости склада). Для каждой пары из клиента c и склада w также известна величина $useCost_{w,c}$, задающая стоимость полного обслуживания клиента c складом w ; в случае, если клиент c потребляет ресурсы склада w только частично, затраты уменьшаются пропорционально степени потребления.

С целью минимизации расходов, компания решила оставить только некоторое подмножество W из имеющихся складов, тем самым сократив расходы на их содержание. Разумеется, закрытые склады не могут больше обслуживать клиентов. Более конкретно, компания каждый месяц тратит деньги на два вида расходов:

1. $\sum_{w \in W} openCost_w$ — затраты на поддержание открытых складов;
2. $\sum_{w,c} x_{w,c} useCost_{w,c}$ — затраты на использование ресурсов складов клиентами.

Вы участвуете в конкурсе по оптимизации расходов, проводимом компанией. Результатом вашей работы должна быть пара (W, X) , состоящая из множества оставленных открытыми складов и плана потребления ресурсов клиентами. Ваша задача — найти любую оптимальную пару (W, X) .

Для решения данной задачи следует воспользоваться методом вариацией метода ветвей и границ под названием «Branch & Cut». Постройте формулировку данной задачи в терминах смешанного программирования (линейного программирования с дополнительными условиями целочисленности некоторых переменных), после чего воспользуйтесь средствами библиотеки GLPK, описанными в разделе «Branch & Cut» документации. Обратите внимание на дополнительные настройки, влияющие на ход работы алгоритма.

2 Входные данные

В первой строке входных данных находятся два целых числа n и m ($1 \leq n, m \leq 150$) — количество складов и клиентов соответственно.

Далее будем считать, что w пробегает диапазон от 0 до $n - 1$, а c пробегает диапазон от 0 до $m - 1$.

В последующих n строках находятся пары чисел $capacity_w, openCost_w$ ($0 \leq capacity_w \leq 10^4, 0 \leq openCost_w \leq 10^5$) — соответственно вместимость и стоимость содержания склада w .

В следующей строке находятся m чисел $demand_c$ ($0 \leq demand_c \leq 10^4$) — величины потреблений клиентов.

Далее следует матрица $n \times m$ из чисел $useCost_{w,c}$ ($0 \leq useCost_{w,c} \leq 10^5$) — стоимости полного обслуживания каждого из клиентов на каждом из складов.

Все величины вместимостей, потреблений и стоимостей — вещественные числа, заданные с не более, чем 5 знаками после запятой.

3 Формат вывода

В первой строке выведите целое число k ($1 \leq k \leq n$) — количество оставленных открытыми складов.

Во второй строке выведите k целых различных чисел от 1 до n в возрастающем порядке — номера оставленных складов в 1-индексации в порядке следования во входных данных.

В последующих k строках выведите матрицу размера $k \times m$, задающую план использования (матрицу из чисел $x_{w,c}$ по всем оставленным складам и всем клиентам).

Выполнение условий $\sum_w x_{w,c} = 1$ будет проверяться с абсолютной точностью 10^{-3} . Ваш ответ будет засчитан, если его относительная ошибка относительно ответа жюри не превосходит 10^{-3} .

4 Контест

При сдаче решения в Яндекс.Контест необходимо использовать `make2` в качестве компилятора. Использовать GLPK надо с помощью `#include "glpk.h"`.

5 Оценивание

Данная задача стоит 3.5 балла.

Вам будет доступно три открытых теста, которые можно будет скачать на странице курса, первый из которых приведён в условии. В первом тесте оптимальная стоимость составляет 70, во втором — 4816.70455, а в третьем — 96931.85081.

Оценка будет состоять из двух частей:

- **Качество вашего решения.** Оценка будет ставиться на основании количества закрытых тестов (с 4 теста и далее), пройденных вашим решением. Тесты с большими номерами соответствуют большим ограничениям.
- **Отчёт.** Требуется написать достаточно подробный отчет, описывающий ваше решение. За качественно написанный отчет будет выставляться 0.5 балла.