Заметки

1 марта 2023 г.

Содержание

1 Представление данных

В общем виде задача ЛП имеет вид

$$C^T[N] \cdot x[N] \to \min$$

$$\begin{cases} A[M_1, N] \cdot x[N] \geqslant b[M_1] \\ A[M_2, N] \cdot x[N] = b[M_2] \\ x[N_1] \geqslant 0 \end{cases}$$

1.1 Чтение из файла.

1. В первой строчке размерность задачи: кол-во переменных и кол-во ограничений m, n
2. Строка в которой записаны разделённые запятой коэфициенты функции цели (предполагается минимизация)
3. m строк, в которых n разделённых запятой коэфициентов, одно из `{GT,LT,EQ}`, означающее $\geq, \leq, =$, свободный член.
4. до n индексов, означающие переменные без ограничения на знак

1.1.1 Пример

Пример в отчёт включать скорее всего не стоит, но про представление я бы написал, в частности то, что задачи в канонической форме генерируются из общей

Задача (если пойдёт в отчёт, это будет с задачей, которую мы используем)

$$x_1 + x_2 + x_3 \to \min$$

$$\begin{cases} 3x_1 - 5x_2 + x_3 \geqslant 3 \\ x_1 + x_2 - x_3 \leqslant 5 \\ 8x_1 + x_2 + 5x_3 = 11 \\ x_1 \geqslant 0, x_2 \geqslant 0 \end{cases}$$

Будет записана как

```
3,3
1,1,1
3,-5,1,GT,3
1,1,-1,LT,5
8,1,5,EQ,11
3
```

1.2 Предобработка

По общей форме задачи генерируется каноническая форма прямой задачи и общая $\to$ каноническая форма двойственной задачи

Каноническая форма хранится в виде вектора коэфициентов функции цели, матрицы коэфициентов ограничений и столбца свободных членов.

2 Симплекс-метод

Решение СЛАУ и преобразование матриц к верхнетреугольному виду для облегчения вычисления определителя («наивный» рекурсивный алгоритм очевидно не подходит для задач большой размерности) использует метод отражений, так как он является точным и уже был реализован в 3м семестре. При решении СЛАУ к ответу приходим, используя обратный ход метода Гаусса. Матрица преобразований сохраняется для того, чтобы на последующих итерациях можно было не решать СЛАУ заново, перейти простыми преобразованиями.

2.1 Начальное приближение

Начальное приближение находится с помощью решения симплекс-методом вспомогательной задачи (метод иск. базиса)

Теория об этом. Последние пол страницы методички.

2.2 Алгоритм симплекс-метода

1. Для начального приближения находим N_k^+, при необходимости L_k^+, N_k (если $|N_k| < n$).
   a) Решим уравнение $A[M, N_k]y_k[M] = C[N_k]$.
   Решение: $y_k^T[M] = C^T[N_k] \cdot B[N_k, M]$, где $B[N_k, M] = A^{-1}[M, N_k]$.
2. Вычисление $d_k^T[L_k] = C^T[N_k] - y_k^T[M] \cdot A[M, N]$
3. Построение $u_k[N]$
   a) Поиск $j_k \in L_k : d_k[j_k] < 0$. Если j_k не найден, $d_k[L_k] \geqslant 0$, следовательно решение оптимальное. Иначе:
   b) $u_k[N_k] = B[N_k, M] \cdot A[M, j_k], u_k[j_k] = -1$, по остальным индексам нули.
4. Построение x_{k+1}
   a) Найдём $P = \{i \in N | u_k[i] > 0\}$. Если $P = \emptyset$, целевая функция не ограничена снизу и алгоритм заканчивается.
   b) $\theta_k = \min_P \frac{x_k[i]}{u_k[i]} = \frac{x_k[i_k]}{u_k[i_k]}$
   c) $x_{k+1} = x_k[N] - \theta_k u_k[N]$.

5. Построение обратной матрицы. Есть два возможных случая:

    a) $\theta_k! = 0$. Тогда можем построить $B[N_{k+1}, M]$ как

    $$B[N_{k+1}, M] = F[N_{k+1}, N_k] \cdot B[N_k, M]$$

    где матрица $F[N_{k+1}, N_k]$ — обратная к $G[N_k, N_{k+1}] = B[N_k, M] \cdot A[M, N_k + 1]$. Матрица G будет отличаться от единичной только столбцом, стоящим на i_k -м месте.

    Этот столбец — u_k. Очевидно $\det G[N_k, N_{k+1}] = u_k[i_k] > 0$, поэтому обратную матрицу очень легко найти

    $$F[N_{k+1}, N_k] = \begin{pmatrix} 1 & \cdots & -u_k[1]/u_k[i_k] & \cdots & 0 \\ \vdots & \ddots & . & . & \vdots \\ 0 & . & 1/u_k[i_k] & . & 0 \\ \vdots & . & . & \ddots & \vdots \\ 0 & \cdots & -u_k[m]/u_k[i_k] & \cdots & 1 \end{pmatrix}$$

    $$B[N_{k+1}, M] = F[N_{k+1}, N_k] \cdot B[N_k, M] \quad y_{k+1}^T[M] = C^T[N_k] \cdot B[N_{k+1}, M]$$

    Исключим из N_k, N_k^+ j_k и добавим туда i_k

    b) Если $x_k[N]$ — вырожденный опорный вектор, может получиться так, что $\theta_k = 0$, если $\exists i \in N_k \setminus N_k^+$. В таком случае пытаемся изменить базис $x_k[N]$.

    Используемый для перестановок базиса алгоритм меняет векторы не по одному, поэтому обратную матрицу сложнее построить. Для упрощения алгоритма (хотя это медленнее) в этом случае решается СЛАУ $A[M, N_{k+1}]y_{k+1}[M] = C[N_{k+1}]$, и обратная матрица получается как побочный эффект решения. После этого вернёмся на шаг 2 алгоритма.

Ещё раз перечислю возможные выходы из цикла:

1. $d_k[N] \geqslant 0$ (3.1) — найдено оптимальное решение
2. $u_k[N] < 0$ (4.1) — целевая функция не ограничена
3. при переходе к следующему базису в (4.2) можем перебрать все возможные комбинации индексов, которыми дополняем N_k^+. В таком случае программа также завершается.

3 Перебор крайних точек