ПРИМЕНЕНИЕ МЕТОДА МАКА В РЕШЕНИИ ЗАДАЧ О НАЗНАЧЕНИЯХ

Россолов С. Ю.

Россолов Сергей Юрьевич / Rossolov Sergey Yurievich – магистрант,
кафедра логистики и управления цепями поставок, факультет торгового и таможенного дела,
Санкт-Петербургский государственный экономический университет, г. Санкт-Петербург

Аннотация: *в статье представлен подробный алгоритм решения задачи о назначениях, которая представляет собой частный случай более общей транспортной задачи, посредством применения метода Мака. Раскрыты суть и практическое значение задачи о назначениях, обозначен основной принцип метода Мака. Текст статьи может быть использован в качестве методического пособия для выполнения практических, лабораторных и самостоятельных работ из раздела методов оптимизации, а также для решения соответствующих задач на предприятиях.*
Ключевые слова: *метод Мака, задачи о назначениях, методы оптимизации, транспортная задача, алгоритм решения.*

При решении некоторых задач менеджмента необходимо назначать исполнителей (людей, механизмы и т.п.) для выполнения однотипных работ. Метод Мака применяется для распределения функций между элементами системы, когда каждый из этих элементов способен выполнять любую из функций с разной эффективностью [1, с. 240]. Предполагается выполнение одним элементом одной функции, и на выходе системы необходимо получить максимальную эффективность. К примеру, в организации n работников, между которыми требуется распределить n заданий так, чтобы все задания были выполнены за суммарно минимальное время. Время выполнения заданий напрямую зависит от свойств характера исполнителей, их квалификации. Следовательно, эффективность труда исполнителей, в силу этих и иных причин, будет различна.

Данная задача носит название задачи о назначениях и представляет собой частный случай более общей транспортной задачи. Существует определенный алгоритм решения задачи о назначениях методом Мака. В первую очередь, составляется матрица: допустим, по вертикали указываются номера работников (Р) организации, а по горизонтали – номера заданий (З), которые соответствуют разным должностям. Таблицу заполняем значениями времени выполнения заданий. Требуется найти минимальное суммарное время выполнения всех n заданий. Ниже приведен пример таблицы исходных данных.

Таблица 1. Исходные данные для решения задачи о назначениях

	$З_1$	$З_2$	$З_3$	$З_4$
P_1	1	5	3	4
P_2	12	2	4	7
P_3	13	3	5	6
P_4	1	4	5	8

Далее необходимо составить систему ограничений по строкам и столбцам. Так как каждый работник должен быть задействован и выполнять одно задание, которое, в свою очередь, должно быть выполнено, сумма элементов в каждой строке (так же и со столбцами) должна быть равна 1 [2, с. 97].

Полное время выполнения заданий будет являться целевой минимизируемой функцией, где в качестве коэффициентов при X_ij будут выступать соответствующие значения времен.

В основе методики решения данной задачи методом Мака лежит принцип, заключающийся в том, что положения оптимального выбора не меняются, если к каждому элементу какой-либо строки или столбца добавить одно и то же число или вычесть его [3, с. 148].

Перейдем непосредственно к алгоритму решения задачи.

Шаг 1. Находим и подчеркиваем минимальный элемент в каждой строке. Если минимальных элементов несколько, помечаем любой из них.

Шаг 2. Множество всех столбцов матрицы делим на два подмножества: A и B. A - множество выбранных столбцов, B – невыбранных. На начальном этапе итерации множеству B принадлежат все столбцы, множество A является пустым. Выбираем из множества B столбец с наибольшим количеством подчеркнутых элементов и переводим его в A.

Шаг 3. Напротив каждой строки (справа от матрицы) выписываем все подчеркнутые элементы множества A. Слева от матрицы только для строк, содержащих подчеркнутые элементы из A, выписываем разности между минимальным элементом множества B для строки и подчеркнутым элементом из множества A той же строки. Далее находим минимальное значение этих разностей.

Шаг 4. Выделяем столбец из A и ячейку с минимальной разностью. В случае если имеется несколько одинаковых минимальных разностей, можно выбрать любую из них, но предпочтительнее ту, которая наиболее выгодна для наделения каждого из столбцов одним подчеркнутым элементом.

Шаг 5. Полученную минимальную разность добавляем ко всем элементам столбца (выделенного A). Все подчеркивания в таблице сохраняются.

Шаг 6. Далее составляется новая таблица, в которой тот элемент из B, отнимая от которого мы получили минимальную разность, помечаем тремя точками внизу.

Шаг 7. Столбец, содержащий элемент с тремя точками, обозначим *.

Далее возможны два варианта:

1) если столбец * не содержит других подчеркнутых элементов: подчеркиваем элемент с тремя точками полностью, убирая при этом другое подчеркивание в одной строке с ним. Это новый элемент, вошедший в базис взамен «старого». Затем переходим к шагу 8;

2) если столбец * содержит еще подчеркнутые элементы: переводим этот столбец в множество A, «забываем» про три точки и возвращаемся к шагу 3.

Шаг 8. Если не все столбцы содержат по одному подчеркнутому элементу, то оптимум еще не найден. Возвращаемся к шагу 2 и начинаем новую итерацию.

Если же оптимальное решение получено, на исходную матрицу наносим в соответствующих местах подчеркивания оптимального расположения элементов. Затем суммируем все подчеркнутые элементы и получаем минимальное время выполнения всех заданий. Задача решена.

Список литературы

1. *Галеев Э. М.* Теория. Примеры. Задачи. / В. В. Алексеев, Э. М. Галеев, В. М. Тихомиров. М.: Физматлит, 2005. 240 с.
2. *Банди Б.* Основы линейного программирования: Пер. с англ. М.: Радио и связь, 2011. С. 96-98.
3. *Большакова И. В.* Линейное программирование / И. В. Большакова, М. В. Кураленко. Мн.: БНТУ, 2004. 148 с.