UNIVERSITY NAME (IN BLOCK CAPITALS)

Исследование алгоритмов

by

Козловский Никита

A thesis submitted in partial fulfillment for the
degree of Doctor of Philosophy

in the
Faculty Name
Department or School Name

Оглавление

Глава 1

Введение

Одной из фундаментальных задач комбинаторной оптимизации является задача о назначениях (ЗОН). В своей классической постановке эта задача звучит так:

Имеется некоторое число работ и некоторое число исполнителей. Любой исполнитель может быть назначен на выполнение любой (но только одной) работы, но с неодинаковыми затратами. Нужно распределить работы так, чтобы выполнить работы с минимальными затратами.

Так как в данной форме рассматривается 2 множества – работников X и работ Y, затраты могут быть выражены ввиде $(c_ij) \in$ A, где A матрица из $Matr_{n\times n}$ и такая задача называется двухиндекской.

В 1955 Куном был опубликовано решение этой задачи [link] в виде Венгерского алгоритма. В 1957 Манкрес определил, что алгорим является строго полиномианльным, а Карп улучшил его, добившись временной сложности $O(n^3)$

Естественно обобщить эту задачу, рассмотрев многоиндексную задачу о назначениях. Однако, уже для трехиндексной ЗОН было показано [кем?], что она принадлежит к классу нп-полных, т.е. не может быть решена за полиномиальное время.

Соответсвенно возникает проблема выбора достаточно хорошего решения. Само собой, эта задача, как и любая задача дискретной оптимизации, может быть решена полным перебором. Однако, слишком большая (экспоненциальная?) временная сложность для такого метода не позволяет использовать его в реальной жизни. Однако, имеет место улучшенная версия этого алгоритма – метод ветвей и границ. В худшем случае он сводится к полному перебору, но чаще требует гораздо меньшего числа операций [для получения *приближенного* решения – а не точный ли он?].

Б’ольшую практическую ценность представляют т.н. эвристические алгоритмы. Они за приемлимое время позволяют получить приближенное решение. Цель данной работы состоит в изучении одного из таких методов, для корого Гимади в [] было показано, что решения, полученные с помощью такого алгоритма сходятся при $n \to \inf$. Для достижения этих целей необходимо решить следующие задачи:

- Изучение математической модели 3-АЗОН
- Изучить метод, предложенный Гимади
- Программно реализовать этот метод
- И провести его анализ

Глава 2

Обзор задачи

Задача о назначениях в самом общем виде формулируется следующим образом: есть некоторое количество *работ* и некоторе количество *исполнителей*. Любой исполнитель может быть назначен на выполнение любой (но только одной) работы, но с неодинаковыми затратами. Нужно распределить работы так, чтобы выполнить работы с минимальными затратами.

Строгая математическая формулировка звучит так.

Пусть даны множества A, T и функционал стоимости $C : A \times T \to \mathbb{R}$. Необходимо найти биекцию $f : A \to T$, такую что $\sum_{a \in A} C(a, f(a))$ минимальна.

Рассмотрим матрицу $C \in Matr_{n \times m}$, тогда $c_i j$ – стоимость назначения i работника на j работу, соотвественно целевая фунция переписывается ввиде $\sum_{a \in A} C_{a,f(a)}$.

Если число исполнителей и работ совпадает, $n = m$, то задачу называют линейной.

Эту задачу также можно переписать в виде задачи линейного программирования.

$$\sum_{i \in A} \sum_{j \in T} C(i,j) x_{ij}$$

и ограничениями

$$\sum_{j \in T} x_{ij} = 1 \text{ для } i \in A$$

$$\sum_{i \in A} x_{ij} = 1 \text{ для } j \in T$$

$$x_{ij} \geq 0 \text{ для } i, j \in A, T$$

В терминах общей формулировки первое ограничение означает, что на каждый работник может быть назначен лишь на одну работу, а второе означает, что каждая работа может быть отдана лишь одному работнику.

Рассмотрим теперь трехиндексную задачу о назначениях.

2.1 Трехиндексная акиальная задача о назначениях

Введем понятие назначения. Мы можем представлять назначение как некое биективное отображение ϕ, которое ставит элементы конечного множества U в соотвествие элементам конечного множества V. В тоже время назначение является перестановкой, которая записывается в виде

$$\begin{pmatrix} 1 & 2 & \dots & \mathrm{n} \\ \varphi(1) & \varphi(2) & \dots & \varphi(n) \end{pmatrix}$$

Каждой перестановке множества $\{1, 2, \dots, n\}$ соответсвует единственная матрица перестановок $\mathrm{X}_\varphi \in \mathrm{Matrix}_{n\times n}$, элементы котороый определяются как

$$x_{ij} = \begin{cases} 1 & \text{если } j = \varphi(i) \\ 0 & \text{иначе} \end{cases}$$

Обозначим множество S_n как множество всех возможных перестановок множества $\{1, 2, \dots, n\}$. Мощность этого множества $n!$.

2.1.1 Линейная задача о назначениях

Пусть дана матрица $n \times n$ весовых коэфициентов $C = (c_{ij})$. Требуется минимизировать линейную форму

$$\sum_{i=1}^{n} c_{i\varphi(i)}$$

то есть линейная задача о назначениях может быть поставлена в виде

$$\min_{\varphi \in S_n} \sum_{i=1}^{n} c_{i\varphi(i)}$$

.

При этом, если перестановки задаются матрицей перестановок X = (x_{ij}), линейная задача о назначениях может быть записана как задача линейной оптимизации

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} \tag{2.1}$$

$$\text{ограничения:} \sum_{i=1}^{n} x_{ij} = 1 \qquad (j = 1, \ldots, n) \tag{2.2}$$

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad (i = 1, \ldots, n) \tag{2.3}$$

$$x_{ij} \in \{0, 1\} \qquad (i, j = 1, \ldots, n) \tag{2.4}$$

Ограничения (2) – (4) задают допустимое множество

В дальнейшем будем называть X матрицей назначений.

Глава 3

Алгоритм решения

3.1 Алгоритм

3.2 Блок-схема

3.3 Комментарии к алгоритму

3.4 Вводимые модификации

3.4.1 Генерация нескольких начальных перестановок

3.4.2 Выбор лучшей перестановки

3.5 Итеративный алгоритм

Глава 4

Программная реализация и вычислительный эксперемент

4.1 Описание

4.2 Эксперимент