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

1 Введение **1**

2 Обзор задачи **3**
2.1 Перестановки 4
2.1.1 Линейная задача о назначениях в терминах перестановок 4
2.1.2 Трехиндексная задача о назначениях 5

3 Алгоритм решения **7**
3.1 Алгоритм 7
3.2 Блок-схема 8
3.3 Комментарии к алгоритму 8
3.4 Вводимые модификации 8
3.4.1 Генерация нескольких начальных перестановок 8
3.4.2 Выбор лучшей перестановки 8
3.5 Итеративный алгоритм 8

4 Программная реализация и вычислительный эксперемент **10**
4.1 Описание 10
4.2 Эксперимент 10

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?].

Большую 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.

2.1 Перестановки

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

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

Каждой перестановке множества $\{1, 2, \dots, n\}$ соответсвует единственная матрица перестановок $\text{X}_\varphi \in \text{Matrix}_{n \times n}$, 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 в терминах перестановок

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.

2.1.2 Трехиндексная задача о назначениях

Аксиальная трехиндесная задача о назначениях может быть определена следующим образом. Пусть даны n^3 весовых коэфициентов $c_{ijk}, (i, j, k = 1 \ldots n)$. Необходимо найти такие перестановки φ и ξ, что

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

где S_n множество всех перестановок целых чисел от $1 \ldots n$.

Так же задача может быть переписана как задача целочисленного линейного программирования.

$$\begin{aligned}
&\min \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} c_{ijk} x_{ijk} \\
\text{ограничения} \quad &\sum_{i=1}^{n} \sum_{k=1}^{n} x_{ijk} = 1 \qquad (j = 1, \ldots, n) \\
&\sum_{j=1}^{n} \sum_{k=1}^{n} x_{ijk} = 1 \qquad (i = 1, \ldots, n) \\
&\sum_{i=1}^{n} \sum_{j=1}^{n} x_{ijk} = 1 \qquad (k = 1, \ldots, n) \\
&x_{ijk} \in \{0, 1\} \qquad (i, j, k = 1, \ldots, n)
\end{aligned}$$

3-АЗОН состоит в том, чтобы выбрать среди элементов трехмерной матрицы $C = c_{ijk}$ такие n^2 элементов, что сумма в каждом выбраном сечении (при фиксированных i, j, k) минимальной.

Трехиндесная задача о назначениях также известна в планарной постановке. Содержательно она отличается от аксиальной тем, что что сумма всех выбранных элементов должна быть минимальна.

Глава 3

Алгоритм решения

Обозначим через f_{A} и $f*$ приближенное (полученное при помощи алгоритма А) и оптимальное значение целевой функции функции в некоторой конкретной задаче соответсвенно.

Будем говорить, что алгоритм А имеет оценки $(\epsilon_{\mathrm{A}}, \delta_{\mathrm{A}})$ если выполнено неравентво

$$Pr\{f_{\mathrm{A}} > (1 + \epsilon_{\mathrm{A}} f*\} \leq \delta_{\mathrm{A}}$$

, где ϵ_{A} есть оценка относительной погрешности решния, получаемого алгоритмом А, δ_{A} – вероятность несрабатывания алгоритма А, что можно трактовать как долю случаев, когда алгоритм А не гарантирует точность в пределах ϵ_{A}.

Алгоритм А назвается асимптотически оптимальным если существую оценки $(\epsilon_{\mathrm{A}}, \delta_{\mathrm{A}})$ стремящиеся к нулю с ростом размерности.

3.1 Алгоритм

Через ϕ обозначим любую целочислено значащую функцию, при этом $1 < \phi_n < n$

1. Берем произвольную подстановку $\pi \in S_n$. Пусть (d_{jk}) - $n \times n$ матрица, содержащая элементы исходной матрицы (c_{ijk}), где индекс $j = \pi(i)$ такой, что

   $$d_{ij} = c_{\pi^{-1}(j)jk}$$

   для любых $1 \leq j$,$n \leq n$ Положим $f = 0; j = 1; \mathrm{K} = 1, 2, \ldots, \phi_n$.

2. Выберем номер $\sigma(j)$ минимального элемента из множества $\operatorname{argmin} d_{jk} | k \in K$.

3. Полагаем $f = f + d_{j\sigma(j)}; \mathrm{K} = \mathrm{K} \setminus \sigma(j); k = j + \phi_n$

4. Если $k \leq n$, то $K = K \bigcap k$.
5. $j = j + 1$
6. Повторяем п.2, пока j<n. В противном случае идем к п.7
7. Результатом работы алгоритма A(ϕ_n) является значение функции f целевой функции $f_{\mathrm{A}(\phi_n)}$.

3.2 Блок-схема

3.3 Комментарии к алгоритму

Асимптотическое поведение трехиндексной аксиальной задачи о назначениях значительно отличается от поведения классической задачи о назначениях. В ряде статей Grundel Krohmal Oleveira было изучено поведение ожидаемого значения оптимальной функции. Ими было доказано, что оно стремится к левой границе распределения весовых коэфициентов.

Лемма

Алгоритм A находит решение 3-АЗН за время $O(n\phi_n)$

Несмотря на то, что алгоритм показывает полиномиальное время выполения, возможен ряд улучшений Например, заметно что алгоритм неустойчив относительно выбора начальной перестановки

3.4 Вводимые модификации

3.4.1 Генерация нескольких начальных перестановок

Изменим алгортим следующим образом

Пусть на начальном этапе дается не одна случайная перестановка, а m. Тогда на выходе можем выбрать лучшую ... Бред

3.4.2 Выбор лучшей перестановки

Введем функционал вида

При известном точном решении будем говорить, что одна перестановка лучше другой если она за тоже число шагов будет ближе сходится к точному решению

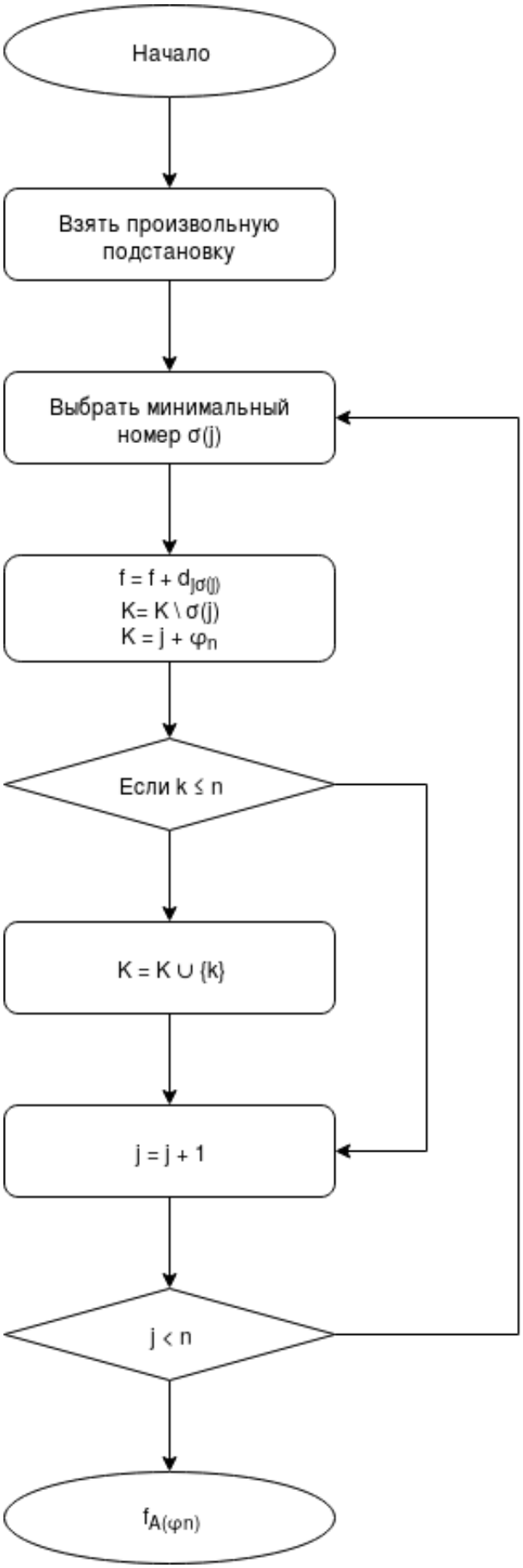
Начало
Взять произвольную подстановку
Выбрать минимальный номер σ(j)
f = f + d_{jσ(j)}
K= K \ σ(j)
K = j + φn
Если k ≤ n
K = K ∪ {k}
j = j + 1
j < n
f_{A(φn)}

Рис. 3.1: A boat.

3.5 Итеративный алгоритм

Добавим следующие шаги После последнего шага сохраним результат, пойдем в начало и запустим алгоритм еще раз Повторим M раз Получим M выводов, в качестве ответа выберем устредненное значение.

Глава 4

Программная реализация и вычислительный эксперемент

4.1 Описание

4.2 Эксперимент