МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Институт Математики, физики и информационных технологий
(наименование института полностью)

Кафедра «Прикладная математика и информатика»
(наименование)

01.03.02 Прикладная математика и информатика
(код и наименование направления подготовки, специальности)

Компьютерные технологии и математическое моделирование
(направленность (профиль)/специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Решение проблемы оптимизации в многоиндексной транспортной задаче»

Обучающийся С.С. Елисеев
(И.О. Фамилия) (личная подпись)

Руководитель к.т.н., доцент, Н.А. Сосина
(ученая степень, звание, И.О. Фамилия)

Консультант к.п.н., доцент, Т.С. Якушева
(ученая степень, звание, И.О. Фамилия)

Тольятти 2022

АННОТАЦИЯ

Выпускная квалификационная работа посвящена решению проблемы оптимизации в многоиндексной транспортной задачи.

Ключевые слова: многоиндексная, транспортная задача, трипланарная, метод потенциалов.

Огромное количество возможных вариантов грузоперевозок делает практически невозможным получение оптимального плана путем простого перебора. Применение математических методов для составления плана перевозок позволяет получить оптимальный план за конечное число итераций. Получение оптимального плана перевозок можно условно поделить на два шага. На первом шаге определяется, с помощью одного из методов нахождения начального опорного плана, например, методом северо-западного угла, опорный план. На втором шаге полученный опорный план последовательно улучшают методом потенциалов или симплекс методом. Объектом исследования данной бакалаврской работы являются методы определения опорного плана в задаче линейного программирования и методы последовательного улучшения опорного плана.

Предметом исследования является процесс решения многоиндексной транспортной задачи.

Целью бакалаврской работы является решение проблемы оптимизации в многоиндексной транспортной задаче и ее программная реализация.

В ходе выполнения выпускной работы было проведено исследование методов определения опорного план и методов последовательного улучшения опорного плана, а также разработан программный продукт на языке C++.

Выпускная квалификационная работа представлена на 56 страницах, включает 20 иллюстраций, 8 таблиц, 59 формул, и список используемой литературы, состоящий из 25 источников.

ABSTRACT

The title of the graduation work is «Solving the optimization problem in a multi-index transport task».

The senior paper consists of an introduction, three parts, a conclusion, list of references including foreign sources.

The key issue of the thesis is the implementation of the method of potentials for optimizing the reference plan of the triplanar transport problem, as well as the preparation of the initial reference plan of the triplanar transport problem by the method of sequential distribution.

The aim of the work is to solve the optimization problem in a multi-index transport problem and its software implementation.

The graduation work may be divided into the following logically interrelated parts: analysis of the method of potentials and the method of sequential distribution for solving a multi-index transport problem; analytical solution of this problem; implementation of the software module of algorithms for solving a multi-index transport problem; testing of the software module based on the problem solved analytically.

Finally, we present the work on the successful implementation of the software module of the method of potentials for solving a multi-index transport problem, the use of which allowed us to obtain an optimal cargo transportation plan.

In conclusion we'd like to stress this work is relevant not only in solving the optimization problem in a triplanar transport problem, but also similar technological solutions can be applied to solve the optimization problem in many multi-index transport problems.

Оглавление

Введение

Любая организация, занимающаяся грузоперевозками, сталкивается с проблемой минимизации себестоимости транспортировки различных грузов. Продумывание логистики доставки грузов, как правило, сводится к транспортной задаче линейного программирования. Из постановки транспортной задачи можно сделать вывод, что количество издержек транспортной компании, при доставке грузов, напрямую зависит от успешности решения транспортной задачи. Решением транспортной задачи является оптимальный план перевозок.

Огромное количество возможных вариантов грузоперевозок делает практически невозможным получение оптимального плана путем простого перебора. Применение математических методов для составления плана перевозок позволяет получить оптимальный план за конечное число итераций. Получение оптимального плана перевозок можно условно поделить на два шага. На первом шаге определяется, с помощью одного из методов нахождения начального опорного плана, например, методом северо-западного угла, опорный план. На втором шаге полученный опорный план последовательно улучшают методом потенциалов или симплекс методом.

Актуальность работы заключается в том, что менеджеру логистической организации необходимо выбрать из большого количества возможных вариантов перевозок такой, который даст наименьшее количество издержек на транспортировку грузов.

Целью бакалаврской работы является решение проблемы оптимизации в многоиндексной транспортной задаче и ее программная реализация.

Объектом исследования данной бакалаврской работы являются методы определения опорного плана и методы последовательного улучшения опорного плана.

Предметом исследования является процесс решения многоиндексной транспортной задачи.

Задачи, которые необходимо решить для достижения указанной цели:

- Изучить и проанализировать методы решения задач линейного программирования транспортного типа, а также метод последовательного распределения, метод потенциалов для решения многоиндексной транспортной задачи;
- Решить аналитически многоиндексную транспортную задачу с использованием оптимизационных методов;
- Реализовать программный модуль алгоритма решения многоиндексной транспортной задачи;
- Протестировать программный модуль на основе задачи, решенной аналитически с использованием оптимизационных методов.

Структура работы включает в себя введение, три главы, заключение, список литературы.

В первой главе рассматривается общая постановка задачи линейного программирования, симплекс метод. Постановка, нахождение опорного плана и метод потенциалов в общем виде для двухиндексной биаксиальной транспортной задачи.

Во второй главе рассматриваются формулировка, модель, свойства трипланарной транспортной задачи. Метод построения начального опорного плана трипланарной транспортной задачи методом последовательного распределения. Метод потенциалов, для последовательного улучшения полученного опорного плана. Аналитическое решение трипланарной транспортной задачи.

В третьей главе описана программная реализация алгоритмов на языке C++.

Глава 1. Анализ задач линейного программирования транспортного типа

1.1 Общая постановка задачи линейного программирования

Математическую формулировку задачи линейного программирования (ЗЛП) транспортного типа можно записать так: найти набор $X = \{x_j\}$, минимизирующий линейную функцию (1)

$$L(X) = \sum_{j=1}^{n} c_j x_j \tag{1}$$

и удовлетворяющий ограничениям (2), (3)

$$\sum_{j=1}^{n} a_{ij} x_{ij} = b_j, i \in \{1,2,\dots,m\} = I, \tag{2}$$

$$x_j \geq 0, j \in \{1,2,\dots,n\} = J. \tag{3}$$

Известно огромное множество публикаций, освещающих свойства ЗЛП, которыми также обладают другие задачи линейного программирования, например, двухиндексная биаксиальная задача.

Задачу линейного программирования удобно формулировать в векторно-матричной форме. Введем векторы $X = (x_1, x_2, \dots, x_n)^{\text{т}}$, $B = (b_1, b_2, \dots, b_m)^{\text{т}}$, $C = (c_1, c_2, \dots, c_n)^{\text{т}}$, где т – знак транспонирования. И матрицу

$$A = (A_1 A_2 \dots A_n) = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}. \tag{4}$$

Векторы-столбцы $A_j, j \in J$, в матрице А называются векторами условий.

Тогда задача (1) – (3) может быть сформулирована следующим образом: найти вектор X*, минимизирующий (5)

$$L(X) = C^{\text{т}} X \tag{5}$$

и удовлетворяющий ограничениям (6)

$$AX = B, X \geq 0. \tag{6}$$

«Набор чисел $X = \{x_1, x_2, \dots, x_n\}$, удовлетворяющий ограничениям задачи (2), (3), называется планом. Оптимальным планом или решением задачи линейного программирования называется план, доставляющий функции (1) экстремальное значение. Задача линейного программирования называется разрешимой, если она имеет хотя бы один оптимальный план. План $X = \{x_j\}$ задачи (1) – (3) называется опорным, если векторы условий, отвечающие его положительным составляющим, линейно-независимы.» [1]

Опираясь на теорию двойственности, можно сформулировать задачу, которая будет двойственной задаче (1) – (3): найти набор $Y^* = \{y_i^*\}$, максимизирующий линейную функцию (7)

$$L(Y) = \sum_{i=1}^{m} b_i y_i \tag{7}$$

и удовлетворяющий ограничениям (8)

$$\sum_{i=1}^{m} a_{ij}\, y_i \leq c_j, j \in J. \tag{8}$$

Исходя из теории двойственности, можно сделать вывод, что для любых двух планов $X = \{x_j\}$ и $Y = \{y_i\}$ задач (1) – (3) и (7), (8) имеет место соотношение (9)

$$\sum_{j=1}^{n} c_j x_j \leq \sum_{i=1}^{m} b_i y_i. \tag{9}$$

«Если одна из задач двойственной пары (1) – (3) и (7), (8) имеет решение, то сопряженная к ней задача также разрешима. При этом для любых оптимальных планов X* и Y* этих задач, соотношение (9) выполняется как строгое равенство» [2] (10)

$$\sum_{j=1}^{n} c_j x_j = \sum_{i=1}^{m} b_i y_i. \tag{10}$$

Критерий оптимальности для некоторого опорного плана, можно сформулировать следующим образом: необходимо и достаточно существование набора $\Lambda = \{\lambda_1, \lambda_2, \dots, \lambda_m\}$, удовлетворяющего условиям (11)

$$c_j^0 = c_j - \sum_{i=1}^{m} a_{ij}\lambda_i \geq 0, j \in J; \sum_{i=1}^{m} a_{ij}\lambda_i = c_j, j \in \{j: x_j > 0\}. \tag{11}$$

Рассмотрим один из методов последовательного улучшения опорного плана – симплекс-метод. Данный метод состоит из конечного числа итераций, которые можно описать следующим образом.

«Пусть в $(q-1)$-й итерации получен опорный план» [6] $X^{(q-1)} = \{x_j^{(q-1)}\}$. Рассмотрим содержание q-й итерации.

«Для проверки оптимальности плана $X^{(q-1)}$ решают систему линейных уравнений» [3] (12)

$$\sum_{i=1}^{m} a_{ij}\lambda_i = c_j, j \in R_{q-1}, \tag{12}$$

где $R_{q-1} = \{j: x_j > 0\}$ – множество «существенных индексов.

Система состоит из m уравнений и m неизвестных и в силу линейной независимости векторов-столбцов матрицы» [1] $A = \{a_{ij}\}, j \in R_{q-1}$, имеет единственное решение. Пусть $\{\tilde{\lambda}_1, \tilde{\lambda}_2, \dots, \tilde{\lambda}_m\}$ – решение этой системы.

«План $X^{(q-1)}$ является оптимальным, если выполняется критерий оптимальности (11), в противном случае план неоптимален.» [4]

Переход к следующему шагу осуществляется, если на первом шаге выявлена неоптимальность опорного плана $X^{(q-1)}$.

«Для получения нового опорного плана, доставляющего целевой функции (1) меньшее значение, в план $\{x_j^{(q-1)}\}$ вводят новую положительную компоненту x_j^*, для которой $c_{j*} = \min\{c_j\}, j \in J$. С этой целью решают систему линейных уравнений» [5] (13)

$$\sum_{j \in R_{q-1}} A_j \Theta_j = -A_{j*} \tag{13}$$

относительно $\Theta_j, j \in R_{q-1}$.

В скалярной форме (13) записывается так (14):

$$\sum_{j \in R_{q-1}} a_{ij}\theta_j = -a_{ij*}, i = 1,2, \dots, m. \quad (14)$$

Пусть $\theta'_j, j \in R_{q-1}$ – решение системы (14). Найдем (15)

$$\frac{x_{j1}^{(q-1)}}{\theta'_{j1}} = \theta_0^{(q)} = \max\left\{\frac{x_j^{(q-1)}}{\theta'_j}\right\}, j \in R_{q-1}^{-}. \quad (15)$$

Значения компонент нового опорного плана $X^{(q)} = \{x_j^{(q)}\}$ определяются по формуле (16)

$$x_j^{(q)} = x_j^{(q-1)} - \theta_j^{(q)}\theta_0^{(q)}, j \in J, \quad (16)$$

где $\theta_j^{(q)} = \begin{cases} 1, j = j^*; \\ \theta'_j, j \in R_{q-1}; \\ 0, j \notin R_{q-1} \cup j^*. \end{cases}$

«В результате выполненных операций в новом опорном плане $X^{(q)} = \left\{x^{(q)}\right\}$ появляется положительная компонента $x_{j*} = \theta_0^{(q)}$ и одновременно становится равной нулю переменная x_{j1}». [4] При этом $R_q = j^* \cup R_{q-1}/j_1$.

Задачи линейного программирования, которые будут рассмотрены далее можно решать симплекс-методом, но учет специфики данных задач позволяет использовать более эффективные алгоритмы.

1.2 Двухиндексная биаксиальная транспортная задача

Постановку двухиндексной биаксиальной транспортной задачи или классической транспортной задачи можно описать следующим образом: найти набор $X = \{x_{ij}\}$, минимизирующий функцию (17)

$$L(X) = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \quad (17)$$

и удовлетворяющий ограничениям (18) – (20)

$$\sum_{j=1}^{n} x_{ij} = a_i, i \in I, \quad (18)$$

$$\sum_{i=1}^{m} x_{ij} = b_j, j \in J, \tag{19}$$

$$x_{ij} \geq 0, (ij) \in E. \tag{20}$$

Задача (17) – (20) называется двухиндексной биаксиальной транспортной. Сформулируем основные свойства этой задачи.

Число переменных задачи равно $m * n$, число условий – равенств – $m + n$. В векторно-матричной форме данную задачу можно записать следующим образом: найти вектор X, минимизирующий линейную форму (21)

$$L(X) = C^{\mathrm{т}} X \tag{21}$$

и удовлетворяющий ограничениям (22), (23)

$$\sum_{i=1}^{m} \sum_{j=1}^{n} P_{ij} x_{ij} = P, \tag{22}$$

$$X \geq 0, \tag{23}$$

где $C = \{c_{11} c_{12} \dots c_{1n} c_{21} c_{22} \dots c_{mn}\}^{\mathrm{т}}; X = \{x_{11} x_{12} \dots x_{1n} x_{21} x_{22} \dots x_{mn}\}^{\mathrm{т}}$; $P = \{a_1 a_2 \dots a_m b_1 b_2 \dots b_n\}$ – вектор ограничений; $P_{ij} = \{0\ 0 \dots 1\ 0 \dots 0\ 1\ 00 \dots 00\}^{\mathrm{т}}$ –

вектор условий размера $m + n$, i-я и $(m + j)$-я компоненты которого равны единице, а остальные – нулю.

Сформулируем условие баланса (24), опираясь на условие разрешимости.

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j = S. \tag{24}$$

Если это условие выполняется, то такая задача называется сбалансированной.

Как и всякой задаче линейного программирования, данной задаче соответствует сопряженная ей двойственная задача: найти наборы $\{u_i\}, \{v_j\}$, максимизирующие функцию (25)

$$L(U, V) = \sum_{i=1}^{m} a_i u_i + \sum_{j=1}^{n} b_j v_j \tag{25}$$

и удовлетворяющие ограничениям (26)

$$u_i + v_j \le c_{ij}, (ij) \in I \times J. \tag{26}$$

Из первой теоремы двойственности можно сделать вывод, что задача (25) – (26) всегда имеет решение, причем если $X = \{x_{ij}\}$ и $W = \{u_1, u_2, \dots, u_m, v_1, v_2, \dots, v_n\}$ – оптимальные планы задач, то (27)

$$\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} = \sum_{i=1}^{m} a_i u_i + \sum_{j=1}^{n} b_i v_j \tag{27}$$

Из теории двойственности можно сформулировать «критерий оптимальности. Для оптимальности плана $X = \{x_{ij}\}$ задачи (17) – (20) необходимо и достаточно существование чисел $u_1, u_2, \dots, u_m, v_1, v_2, \dots, v_n$ таких, что

$$\begin{cases} u_i + v_j \le c_{ij}, (ij) \in I \times J, \\ u_i + v_j = c_{ij}, \text{если } x_{ij} > 0. \end{cases} \tag{28}$$

Числа $u_i, i \in I,$ и $v_j, j \in J,$ обычно называют потенциалами.» [8]

Опорный план задачи «называется невырожденным, если он содержит ровно $m + n - 1$ положительных компонент. Опорный план, не удовлетворяющий этому условию, называется вырожденным.» [9]

«Алгоритм «северо-западного угла» состоит из $m + n - 1$ однотипных шагов. На каждом шаге определяются значения переменных плана, расположенных в одном ряду (строке или столбце) матрицы $\{x_{ij}\}$. Пусть проведено $q - 1$ шагов алгоритма, в результате» [10] которых получена матрица $\{x_{ij}\}$, содержащая подматрицу $\{x_{ij}\}_{i_q j_q}, i \in \{i_q, i_q + 1, \dots, m\}, j \in \{j_q, j_q + 1, \dots, n\}$, с неизвестными значениями переменных x_{ij}. На очередном, q-м шаге алгоритма выполняются следующие операции.

Если $a_{i_q} \le b_{j_q}$, то $x_{i_q j_q} = a_{i_q}, x_{i j_q} = 0, i \in \{i_q + 1, i_q + 2, \dots, m\}, b_{j_q} = b_{j_q} - a_{i_q}, i_{q+1} = i_q + 1.$

Если $a_{i_q} > b_{j_q}$, то $x_{i_q j_q} = b_{j_q}, x_{i_q j} = 0, j \in \{j_q + 1, j_q + 2, \dots, n\}, a_{i_q} = a_{i_q} - b_{j_q}, j_{q+1} = j_q + 1.$

В результате выполнения данного алгоритма будет получен начальный опорный план транспортной задачи.

«Рассмотрим формальную схему алгоритма метода потенциалов. Пусть в $(q-1)$-й итерации получен опорный план $X^{(q-1)} = \{x_{ij}^{(q-1)}\}$.

Шаг 1. Для проверки оптимальности решают систему линейных уравнений (29)» [11]

$$u_i + v_j = c_{ij}, (ij) \in \left\{(ij) :\ x_{ij}^{(q-1)} > 0\right\} = R_{(q-1)}. \tag{29}$$

Пусть $\tilde{V} = \{\tilde{u}_1, \tilde{u}_2, \ldots, \tilde{u}_m, \tilde{v}_1, \tilde{v}_2, \ldots, \tilde{v}_m\}$ – решение системы (29). План $= \{x_{ij}^{(q-1)}\}$ является оптимальным, если $c_{ij}^0 = c_{ij} - \left(u_i + v_j\right) \geq 0$ для всех $(ij) \in I \times J$.

Иначе осуществляется переход к следующему шагу.

Шаг 2. «Для получения нового опорного плана, доставляющего целевой функции (17) меньшее значения, в план $\{x_{ij}^{(q-1)}\}$ вводят новую положительную компоненту» [12] $x_{i^*j^*}$, для которой $c_{i^*j^*} = \min\left\{c_{ij}^0\right\}, (ij) \in I \times J$.

Для этого решается система линейных уравнений (30)

$$\sum_{(ij)\in R_{q-1}} P_{ij}\Theta_{ij} = -P_{i^*j^*} \tag{30}$$

относительно $\Theta_{ij}, (ij) \in R_{q-1}$.

Можно показать, что система (30) имеет единственное решение, причем $\Theta_{ij} \in \{-1; 0; 1\}$. Решение системы (30), сводится к поиску среди базисных компонент опорного плана $x_{ij}^{(q-1)}$замкнутой цепочки и приписыванию каждой из неизвестных Θ_{ij} значения +1 или -1 в зависимости от ее расположения в данной последовательности.

Пусть $\left\{\Theta_{ij}'\right\}, (ij) \in R_{q-1}$, – решение системы (3). Найдем

$$\frac{x_{i_1 j_q}^{(q-1)}}{\Theta_{i_1 j_1}'} = \Theta_0^{(q)} = \max\left\{\frac{x_{ij}^{(q-1)}}{\Theta_{ij}'}\right\}, (ij) \in R_{q-1}^- \tag{31}$$

где $R_{q-1}^- = \left\{(ij) :\ \Theta_{ij}' < 0\right\}$.

«Теперь значения компонент нового опорного плана» [11] $X^{(q)} = \{x_{ij}^{(q)}\}$ определяются по формуле (32)

$$x_{ij}^{(q)} = x_{ij}^{(q-1)} - \Theta_{ij}^{(q)}\Theta_0^{(q)}, (ij) \in I \times J, \tag{32}$$

где

$$\Theta_{ij}^{(q)} = \begin{cases} 1, (ij) = (i^* j^*); \\ \Theta'_{ij}, (ij) \in R_{q-1}; \\ 0, (ij) \notin R_{q-1} \cup (i^* j^*). \end{cases} \tag{33}$$

В результате в $X^{(q)} = \{x_{ij}^{(q)}\}$ «появляется новая положительная компонента» [13] $x_{i^* j^*} = \Theta_0^{(q)} > 0$ и обнуляется переменная $x_{i_1 j_1}$. При этом

$$R_q = (i^* j^*) \cup R_{q-1} / (i_1 j_1) \tag{34}$$

Итерации повторяются до выполнения критерия оптимальности (28).

Выводы по 1 главе.

В первой главе сформулирована постановка задачи линейного программирования и ее свойства. Решением данной задачи является оптимальный план, найденный с помощью метода последовательного улучшения опорного плана – симплекс-метода.

В главе также была рассмотрена постановка двухиндексной биаксиальной транспортной задачи, сформулированы условие разрешимости и критерий оптимальности. Для решения биаксиальной транспортной задачи приводится построение начального опорного плана методом «северо-западного угла», а также метод потенциалов, для последовательного улучшения данного опорного плана.

Глава 2. Многоиндексная транспортная задача

2.1 Формулировка и модель трипланарной транспортной задачи

«На практике часто возникает задача составления плана транспортировки некоторого однородного продукта от центров производства к центрам потребления с использованием транспортных средств различных типов, реализация которого обеспечила бы минимальные транспортные издержки. Понятно, что решить такую задачу, опираясь на классическую транспортную теорию, невозможно, так как в рассматриваемом случае стоимость транспортировке единицы продукта зависит не только от взаимного расположения центров производства и потребления, но и от типа транспорта, а к обычным транспортным ограничениям добавляются дополнительные ограничения на количество продукта, перевозимого средствами данного типа.» [14]

«Будем считать известными следующие неотрицательные величины:

$a_i, i \in I,$ – количество (запас) продукции, находящейся в распоряжении i-го центра производства;

$b_j, j \in J,$ – количество продукции, необходимое j-му центру потребления;

$c_k, k \in K,$ – количество продукции, которое может быть перевезено всеми единицами транспорта k-го типа;

$c_{ijk}, i \in I, j \in J,\ k \in K,$ – стоимость транспортировки единицы продукта из i-го центра производства в j-й центр потребления транспортным средством k-го типа.» [15]

«Совокупность чисел $\{c_{ijk}\}, i \in I, j \in J,\ k \in K,$ как правило, называют трехиндексной матрицей» [1] размера $m \times n \times p$, которую можно представить на рисунке 1.

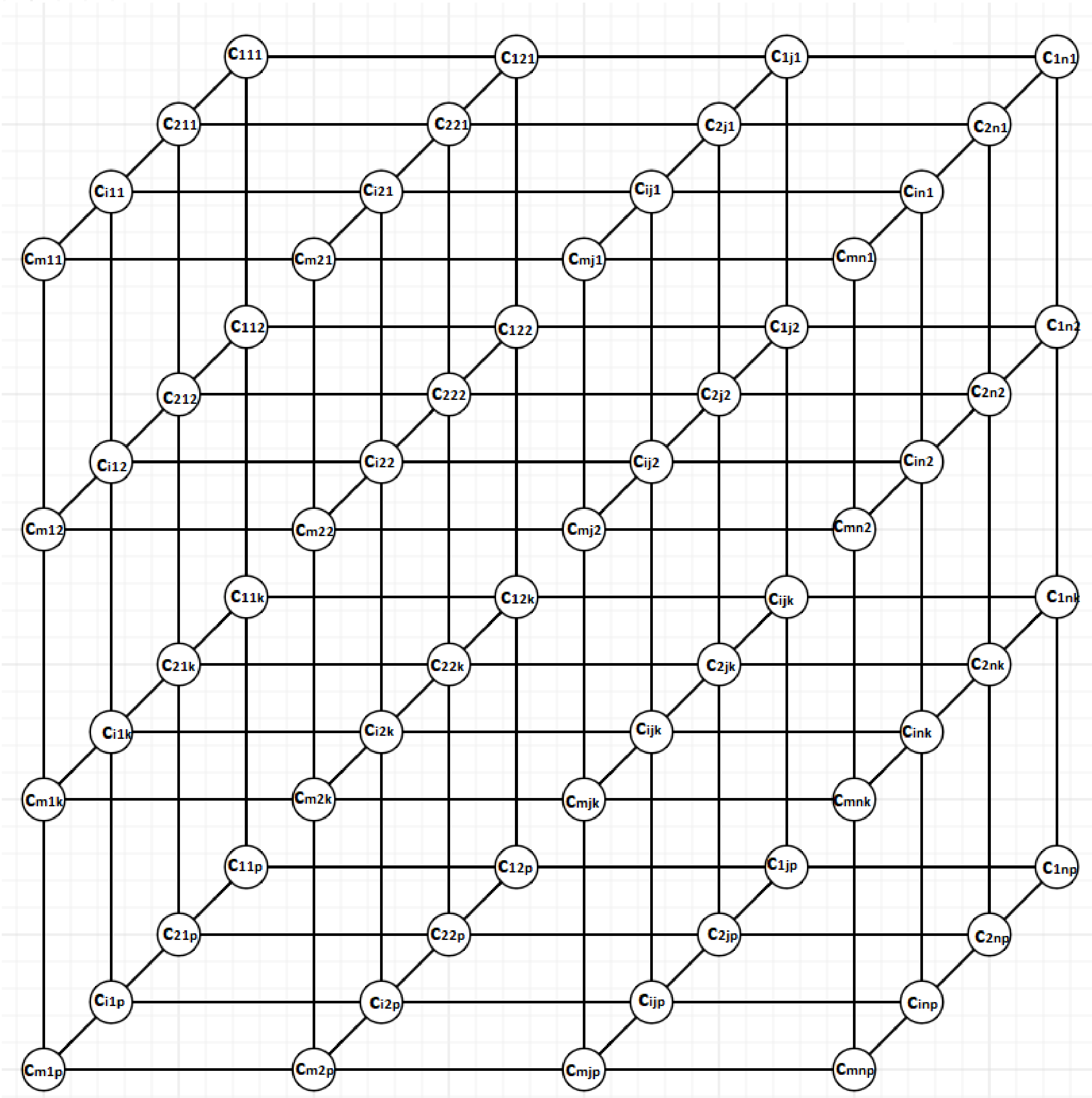

Рисунок 1 – трехмерная числовая решетка $\{c_{ijk}\}$

«Пусть $x_{ijk}, i \in I, j \in J, k \in K,$ есть количество продукции, планируемое для перевозки из i-го центра производства в j-й центр потребления транспортными средствами k-го типа. Тогда совокупность чисел $\{x_{ijk}\}, i \in I, j \in J, k \in K$ естественно назвать планом транспортировки, который также представляет собой трехиндексную матрицу. Очевидно, что размеры матриц $\{c_{ijk}\}$ и $\{x_{ijk}\}$ совпадают.» [16]

«Элемент x_{ijk} матрицы $\{x_{ijk}\}$ назовем компонентой плана. Заметим, что суммы компонент плана по каждому из сечений матрицы $\{x_{ijk}\}$ имеют вполне определенный смысл. Так, например, сумма элементов матрицы $\{x_{ijk}\}$ по одномерному сечению $X_{jk}^{i}, x_{jk} = \sum_{i=1}^{m} x_{ijk}$ – это общее количество продукции, доставляемое j-му потребителю транспортными средствами k-го типа.» [14]

Сумма компонент плана по двумерному сечению $X_{k}^{ij}, x_k = \sum_{i=1}^{m}\sum_{j=1}^{n} x_{ijk} = \sum_{j=1}^{n} x_{ij} = \sum_{i=1}^{m} x_{ik}$ – это общее количество продукции, перевозимое транспортными средствами k-го типа.

«План транспортировки должен удовлетворять определенным условиям:

- количество продукции, планируемое для вывоза из каждого центра производства, не должно превышать соответствующий запас,
- количество продукции, планируемое для ввоза каждому из потребителей, должно быть не меньше его потребности,
- общее количество продукции, планируемое для перевозки транспортным средством k-го типа, не должно превышать возможности данных средств.» [17]

Формальная запись указанных условий имеет следующий вид (35):

$$\begin{cases} \sum_{j=1}^{n}\sum_{k=1}^{p} x_{ijk} \le a_i, i \in I; \\ \sum_{i=1}^{m}\sum_{k=1}^{p} x_{ijk} \ge b_j, j \in J; \\ \sum_{i=1}^{m}\sum_{j=1}^{n} x_{ijk} \le c_k, k \in K. \end{cases} \qquad (35)$$

В большинстве «случаев требования к плану транспортировки формулируют более жестко:

– из центров производства должен быть вывезен весь наличный запас продукции,

– спрос всех потребителей должен быть удовлетворен в точном соответствии с их потребностями,

– количество продукции, планируемое для перевозки средствами k-го типа, должно быть равно возможностям средств этого типа.

В этом случае формульная запись ограничений имеет вид» [18] (36) – (38):

$$\sum_{j=1}^{n}\sum_{k=1}^{p} x_{ijk} = a_i, i \in I, \tag{36}$$

$$\sum_{i=1}^{m}\sum_{k=1}^{p} x_{ijk} = b_j, j \in J, \tag{37}$$

$$\sum_{i=1}^{m}\sum_{j=1}^{n} x_{ijk} = c_k, k \in K. \tag{38}$$

К ограничениям (36) – (38) добавляется условие неотрицательности каждой компоненты плана (39):

$$x_{ijk} \geq 0 \text{ для всех } i \in I, j \in J, k \in K. \tag{39}$$

«Стоимость перевозки x_{ijk} единиц продукта равна $\text{c}_{ijk}x_{ijk}$. Суммируя такие произведения по всем трем индексам, получаем общую величину транспортных издержек» [19] (40)

$$L(X) = \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{p} c_{ijk}x_{ijk} \tag{40}$$

«Теперь задача составления плана перевозок может быть сформулирована как задача линейного программирования: найти набор $X^* = \{x_{ijk}^*\}$, минимизирующий функцию (40) и удовлетворяющий условиям (36) – (39).» [17]

«Задача (36) – (40) называется трехиндексной с планарными суммами или трипланарной транспортной задачей и обозначается T-3P. Величины

a_i, b_j, c_k, c_{ijk} – параметры задачи Т-ЗР. Набор $\{x_{ijk}\}$ – набор переменных задачи. Параметры полностью определяют задачу.» [20]

Представим план задачи Т-ЗР с ограничениями (36) – (39) в трехмерном пространстве на рисунке 2.

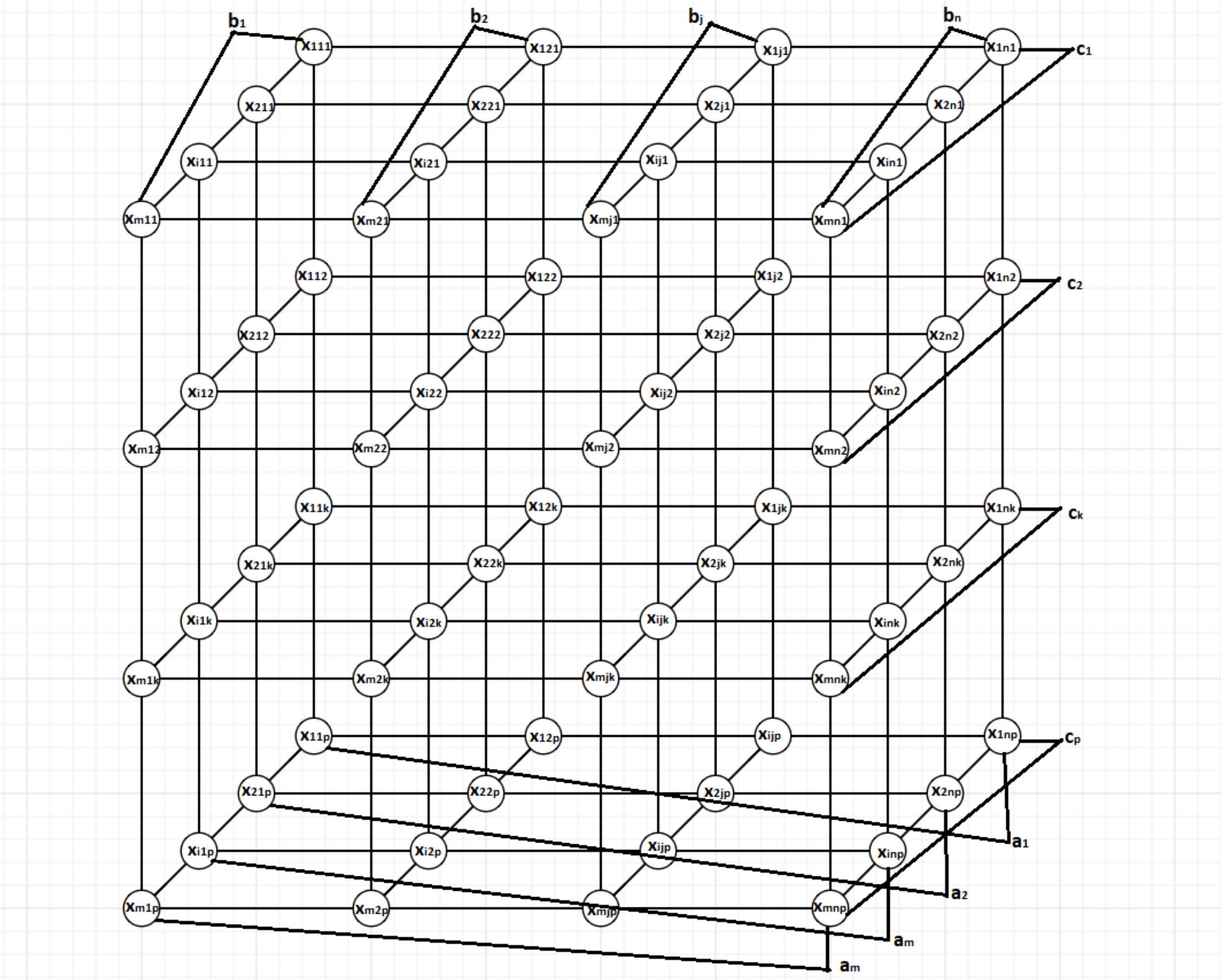

Рисунок 2 – план задачи Т-ЗР с ограничениями

Данная схематическая запись, конечно, дает возможность наглядно представить задачу Т-ЗР, но гораздо удобнее записать параметры и переменные в виде таблицы 1.

Таблица 1 – таблица переменных

x_{111}				x_{121}				x_{1n1}				x_{1*1}				a_1
	x_{112}				x_{122}				x_{1n2}				x_{1*2}			
		…				…				…				…		
			x_{11p}				x_{12p}				x_{1np}				x_{1*p}	
x_{211}				x_{221}				x_{2n1}				x_{2*1}				a_2
	x_{212}				x_{222}				x_{2n2}				x_{2*2}			
		…				…				…				…		
			x_{21p}				x_{22p}				x_{2np}				x_{2*p}	
x_{m11}				x_{m21}				x_{mn1}				x_{m*1}				a_m
	x_{m12}				x_{m22}				x_{mn2}				x_{m*2}			
		…				…				…				…		
			x_{m1p}				x_{m2p}				x_{mnp}				x_{m*p}	
b_1				b_2				b_n				c_1	c_2	…	c_p	

«Рассмотрим структуру таблицы 1, называемой таблицей переменных и обозначаемой T_x. Значения переменных x_{ijk} записываются по диагонали каждого блока таблицы. «*», поставленная вместо индекса, означает, что по этому индексу произведено суммирование. Таким образом, $x_{i*k} = \sum_{j=1}^{n} x_{ijk}$. В нижней строке таблицы записываются свободные члены уравнений (37), в крайнем правом столбце – свободные члены условий (36), в правом нижнем углу – свободные члены условий (37).» [21]

Рассмотренную выше интерпретацию задачи Т-ЗР обычно называют канонической.

2.2 Свойства задачи Т-ЗР

Сформулируем некоторые свойства задачи Т-ЗР. «Набор переменных $\{x_{ijk}\}$, удовлетворяющий условиям (36) – (39) называется допустимым планом, или просто планом задачи. Каждому плану соответствует определенное значение целевой функции (40). Допустимый план, который обеспечивает минимальное значение среди значений целевой функции на множестве всех допустимых планов, называется оптимальным планом задачи. Задача линейного программирования является разрешимой, если существует оптимальный план. Поэтому оптимальный план иногда называют решением задачи.» [22]

«Для разрешимости задачи Т-3Р необходимо и достаточно выполнения условия (41)» [23]

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j = \sum_{k=1}^{p} c_k = S. \tag{41}$$

«Условие совместимости (41) системы ограничений (36) – (39) называется условием баланса, а задачи, для которых это условие выполняется, – сбалансированными транспортными задачами.» [24]

Рассмотрим более подробно структуру системы ограничений задачи Т-3Р. Запишем в развернутом виде систему уравнений (36) – (38) для задачи, схематически изображенной на рисунке 3, где $m = 3, n = 3, p = 2$.

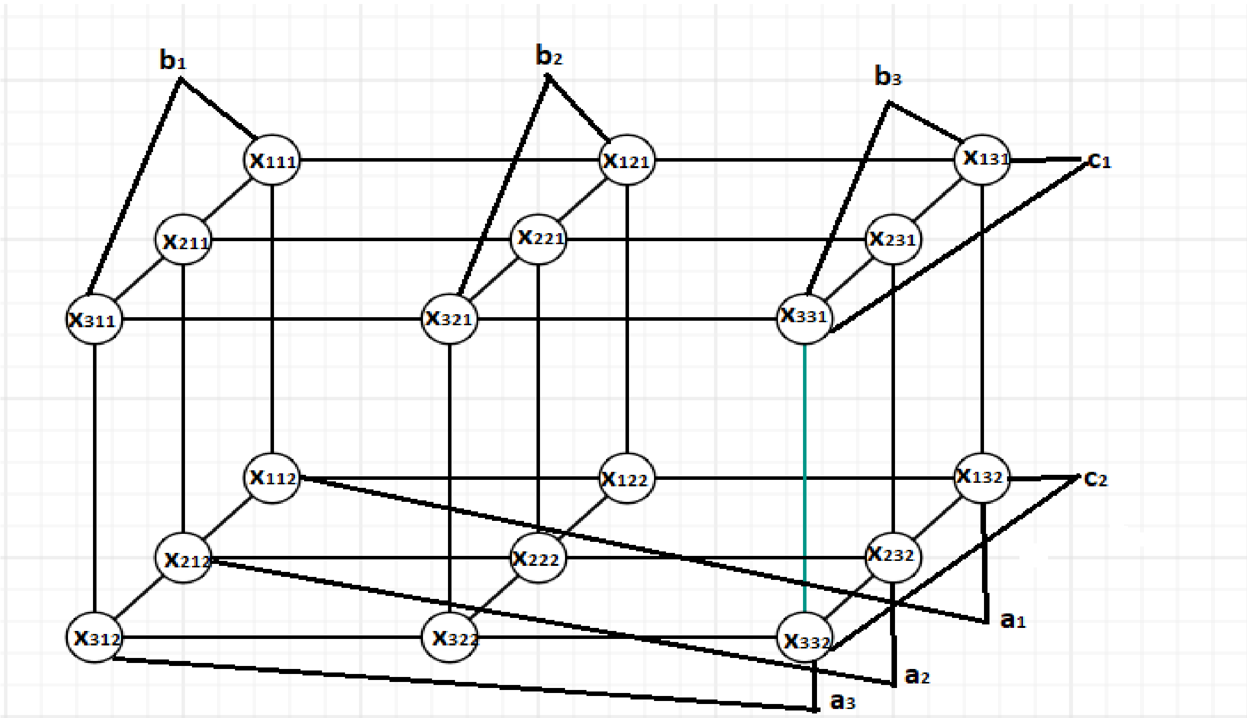

Рисунок 3 – задача размером 3 × 3 × 2

$$
\begin{cases}
x_{111}+x_{112}+x_{121}+x_{122}+x_{131}+x_{132}=a_1\\
x_{211}+x_{212}+x_{221}+x_{222}+x_{231}+x_{232}=a_2\\
x_{311}+x_{312}+x_{321}+x_{322}+x_{331}+x_{332}=a_3\\
x_{111}+x_{112}+x_{211}+x_{212}+x_{311}+x_{312}=b_1\\
x_{121}+x_{122}+x_{221}+x_{222}+x_{321}+x_{322}=b_2\\
x_{131}+x_{132}+x_{231}+x_{232}+x_{331}+x_{332}=b_3\\
x_{111}+x_{121}+x_{131}+x_{211}+x_{221}+x_{231}+x_{311}+x_{321}+x_{331}=c_1\\
x_{112}+x_{122}+x_{132}+x_{212}+x_{222}+x_{232}+x_{312}+x_{322}+x_{332}=c_2
\end{cases}
\tag{42}
$$

Коэффициенты системы (42) можно представить в виде таблицы 2.

Таблица 2 – коэффициенты системы

	i	1	1	1	1	1	1	2	2	2	2	2	2	3	3	3	3	3	3	R
	j	1	1	2	2	3	3	1	1	2	2	3	3	1	1	2	2	3	3	
	k	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	
1	E_1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	a_1
2		0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	a_2
3		0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	a_3
4	E_2	1	1	0	0	0	0	1	1	0	0	0	0	1	1	0	0	0	0	b_1
5		0	0	1	1	0	0	0	0	1	1	0	0	0	0	1	1	0	0	b_2
6		0	0	0	0	1	1	0	0	0	0	1	1	0	0	0	0	1	1	b_3
7	E_3	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	c_1
8		0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	c_2
λ / G		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	

Обозначим систему «коэффициентов (42) через П, а вектор-столбец свободных членов – через R.» [25]

Матрица П представляет собой совокупность векторов-столбцов $P_\sigma, \sigma = 1,2,\ldots,mnp$, т.е. $\Pi = (P_1, P_2, \ldots, P_{mnp})$ или в трехиндексной нумерации $\Pi = \left(P_{111}, P_{112}, \ldots, P_{mnp}\right)$.

Система ограничений (36) – (38) задачи Т-3Р состоит из группы уравнений (36), (37) и (38), которые называются соответственно группам типа i, j и k.

Матрица $\Pi = (\pi_{\lambda\sigma})$, $\lambda \in \{1,2,\ldots,m+n+p\}$, $\sigma \in \{1,2,\ldots,mnp\}$, коэффициентов системы (36) – (38) имеет размер $(l \times s)$, где « $l = m+n+p, s = mnp$, и состоит из нулей и единиц. Множество E_0

номеров строк матрицы П можно разделить на три подмножества E_1, E_2 и E_3 ($\bigcup_{i=1}^{3} E_i = E_0, E_1 \cap E_2 = E_1 \cap E_3 = E_2 \cap E_3 = \emptyset$), порождаемых соответственно группами условий типа i, j и k. Подматрицы Π_1, Π_2 и Π_3, образованные из строк указанных подмножеств, имеют размеры, соответственно» [19] $m \times s, n \times s$ и $p \times s$, причем

$$\sum_{\sigma=1}^{s} \text{п}_{\lambda\sigma} = \begin{cases} np, \lambda \in E_1, \\ mp, \lambda \in E_2, \\ mn, \lambda \in E_3, \end{cases} \quad (43)$$

$$\sum_{\lambda \in E_p} \text{п}_{\lambda\sigma} = 1, \sigma \in \{1,2,\dots,s\}, p \in \{1,2,3\}. \quad (44)$$

Также, «опорный план задачи Т-3Р содержит не более чем $m + n + p - 2$ положительных компонент. Если опорный план содержит ровно $m + n + p - 2$ положительных компонент, то такая задача называется невырожденной.» [20]

Трехиндексная матрица $\{c_{ijk}\}$ и план задачи $\{x_{ijk}\}$ могут быть записаны в виде mnp-мерных векторов $C = (c_{111}, \dots, c_{mnp})$, $X = (x_{111}, \dots, x_{mnp})$. Тогда задачу (36) – (40) можно представить в следующей форме: отыскать вектор X^*, минимизирующий (45)

$$L(X) = CX \quad (45)$$

и удовлетворяющий ограничениям (46), (47)

$$\sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{p} P_{ijk} x_{ijk} = B, \quad (46)$$

$$X \geq 0. \quad (47)$$

«Основной метод решения транспортных задач – метод потенциалов – основан на теории двойственности линейного программирования. Сформулируем задачу, двойственную задаче Т-3Р. Введем вектор двойственных переменных $V = \{u_1, u_2, \dots, u_m, v_1, v_2, \dots, v_n, w_1, w_2, \dots, w_p\}$. Двойственная задача формулируется следующим образом: найти набор» [20] $V = \{u_1, u_2, \dots, u_m, v_1, v_2, \dots, v_n, w_1, w_2, \dots, w_p\}$, максимизирующий функцию (48)

$$L(V)=\sum_{i=1}^{m}a_iu_i+\sum_{j=1}^{n}b_jv_j+\sum_{k=1}^{p}c_kw_k \tag{48}$$

и удовлетворяющий ограничениям (49)

$$u_i+v_j+w_k\le c_{ijk}, i\in I, j\in J, k\in K. \tag{49}$$

Величины $u_i, i\in I, v_j, j\in J, w_k, k\in K,$ называют потенциалами задачи Т-ЗР.

Двойственную задачу (48), (49) обозначим $\tilde{T}$-ЗР. Из теории двойственности можно сделать некоторые наблюдения, пусть наборы $X=\{x_{ijk}\}$ и $V=\{u_i, v_j, w_k\}$ допустимые планы задач Т-Зр и $\tilde{T}$-ЗР. Справедливо следующее соотношение (50):

$$\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{p}c_{ijk}x_{ijk}\ge\sum_{i=1}^{m}a_iu_i+\sum_{j=1}^{n}b_jv_j+\sum_{k=1}^{p}c_kw_k, \tag{50}$$

которое обращается в равенство тогда и только тогда, когда планы X и V являются оптимальными.

«Для оптимальности плана X задачи Т-ЗР необходимо и достаточно существования чисел $u_i, i\in I, v_j, j\in J, w_k, k\in K,$ таких, что (51)

$$u_i+v_j+w_k\le c_{ijk}, i\in I, j\in J, k\in K, \tag{51}$$

причем (52)

$$u_i+v_j+w_k=c_{ijk}, (ijk)\in\{(ijk):x_{ijk}>0\}. \tag{52}$$

Введем следующее преобразование матрицы коэффициентов целевой функции» [20] (53):

$$a_{ijk}=c_{ijk}+\alpha_i+\beta_j+\gamma_k, i\in I, j\in J, k\in K, \tag{53}$$

преобразование (53) называется эквивалентным и будет эффективно использовано для построения ряда вычислительных алгоритмов.

2.3 Построение начального опорного плана

Ранее был рассмотрен метод «северо-западного угла» для нахождения начального опорного плана двухиндексной транспортной задачи. На основе данного метода был разработан другой метод, с помощью которого возможно построить начальный опорный план как для трехиндексной транспортной задачи, так и для транспортной задачи, обладающей большей индексностью.

Метод построения начального опорного плана трипланарной транспортной задачи называется метод последовательного распределения. В основе метода «последовательного распределения лежит следующее соображение. Выберем произвольный элемент $x_{i_1 j_1 k_1}$ матрицы $\{x_{ijk}\}$ и положим его равным $x_{i_1 j_1 k_1} = \min\{a_{i_1}, b_{j_1}, c_{k_1}\}$. Элемент $x_{i_1 j_1 k_1}$входит в три ограничения»[7] $\sum_{j=1}^{n}\sum_{k=1}^{p} x_{i_1 jk} = a_{i_1}$; $\sum_{i=1}^{m}\sum_{k=1}^{p} x_{ij_1 k} = b_{j_1}$;$\sum_{i=1}^{m}\sum_{j=1}^{n} x_{ijk_1} = c_{k_1}$. «Введем $b'_{j_1} = b_{j_1} - a_{i_1}$ и $c'_{k_1} = c_{k_1} - a_{i_1}$. Теперь, приняв значения остальных переменных в сечении $X_{i_1}^{jk}$ равными нулю, можно исключить это сечение из матрицы $\{x_{ijk}\}$, уменьшив таким образом размерность матрицы на единицу. В результате выполненных операций получаем новую усеченную матрицу $\{x_{ijk}\}$ размера $(m-1)\times n\times p$ и новый вектор ограничений $B' = \{a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_m, b_1, \dots, b_j, \dots, b_n, c_1, \dots, c_{k1}, \dots, c_p\}$.» [21]

«Указанная процедура может быть продолжена. Элемент x_{ijk} матрицы назначений, который на очередном шаге вводится в множество ненулевых компонент плана, называется ведущим. С целью упорядочения выбора очередного ведущего элемента удобно в качестве ведущего на первом шаге выбирать элемент x_{111}, а на каждом из последующих шагов – элемент x_{ijk} с минимальными значениями индексов i,j,k.» [19]

2.4 Метод потенциалов для решения трипланарной транспортной задачи

Начальный опорный план задачи Т-3Р не всегда является оптимальным. Для последовательного улучшения данного плана обычно применяется метод потенциалов, итерации которого можно поделить на два этапа. Во время выполнения очередной итерации, на первом этапе, применяется критерий оптимальности для проверки полученного опорного плана. Если данный опорный план прошел проверку, т.е. является оптимальным, то решение закачивается и на основе полученного опорного плана высчитывается целевая функция задачи. Очевидно, что значение целевой функции при оптимальном плане будет меньше, чем при начальном опорном плане, полученном методом последовательного распределения. Если на первом этапе итерации выясняется, что опорный план не является оптимальным, то идет переход ко второму этапу итерации, на котором данный опорный план улучшается, т.е. становится близким к оптимальному.

Пусть в $(q-1)$-й итерации получен опорный план $X^{(q-1)} = \{x_{ijk}^{(q-1)}\}$ и известна матрица коэффициентов $C^{(q-1)} = \left\{C_{ijk}^{(q-1)}\right\}$. Тройку значений индексов i, j, k, называют индексным элементом и обозначают (i,j,k). Прямое произведение множеств значений индексов $I = \{1,2,\dots,m\}$, $J = \{1,2,\dots,n\}$, $K = \{1,2,\dots,p\}$, составляющее множество индексных элементов, обозначим через E: $E = I \times J \times K$. Пусть R_{q-1} – множество индексных элементов, соответствующих положительным компонентам плана $X^{(q-1)} = \left\{x_{ijk}^{(q-1)}\right\}$. Таким образом, $R_{q-1} = \{(i,j,k) :\ x_{ijk}^{(q-1)} > 0\}$. Будем называть R_{q-1} множеством существенных индексных элементов. Для невырожденных задач число элементов множества R_{q-1} равно $m+n+p-2$. Рассмотрим содержание q-й итерации.

Шаг 1. Для проверки оптимальности плана $X^{(q-1)}$ решим систему линейных уравнений (54)

$$u_i + v_j + w_k = c_{ijk}^{(q-1)}, (i,j,k) \in R_{q-1}. \tag{54}$$

Система (54) состоит из $m+n+p-2$ уравнений и содержит $m+n+p$ неизвестных и, следовательно, является недоопределенной. Для получения одного из решений системы приравняем нулю два произвольных неизвестных с различными индексами, например u_{i_0} и v_{j_0}. Пусть $\tilde{V} = \left\{\widetilde{u_1}, \widetilde{u_2}, \dots, \widetilde{u_m}, \widetilde{v_1}, \widetilde{v_2}, \dots, \widetilde{v_n}, \widetilde{w_1}, \widetilde{w_2}, \dots, \widetilde{w_p},\right\}$ – решение системы (54). Числа

$\widetilde{u_i}, i \in I, \widetilde{v_j}, j \in J, \widetilde{w_k}, k \in K,$ называются предварительными потенциалами задачи Т-3Р.

Признаком оптимальности опорного плана $X^{(q-1)} = \left\{x_{ijk}^{(q-1)}\right\}$ является выполнение условий (55)

$$c_{ijk}^{(q)} = c_{ijk}^{(q-1)} - \left(\widetilde{u_i} + \widetilde{v_j} + \widetilde{w_k}\right) \geq 0, (ijk) \in E. \tag{55}$$

Если хотя бы одно из условий (55) не выполняется, т.е. существует хотя бы один индексный элемент $(i^* j^* k^*)$, для которого $c_{i^* j^* k^*}^{(q)} < 0$, то план может быть улучшен и осуществляется переход ко второму шагу.

Шаг 2. Для получения нового опорного плана, доставляющего линейной форме (40) меньшее значение, введем в предыдущий план компоненту $x_{i^* j^* k^*}$, для которой (56)

$$c_{i^* j^* k^*}^{(q)} = \min\left\{c_{ijk}^{(q)}\right\}, (ijk) \in E. \tag{56}$$

Делается это следующим образом.

Решим систему линейных уравнений относительно неизвестных θ_{ijk}

$$\sum_{(ijk) \in R_{q-1}} P_{ijk}\, \theta_{ijk} = -P_{i^* j^* k^*}, \tag{57}$$

где P_{ijk} – векторы-столбцы матрицы условий П.

Система (57) содержит $m + n + p$ уравнений и $m + n + p - 2$ неизвестных. Однако она не является переопределенной, так как два уравнения из системы (57) являются следствиями остальных и могут быть отброшен. Тогда, в силу линейной независимости векторов условий P_{ijk}, соответствующих положительным компонентам опорного плана $X^{(q-1)}$, система имеет единственное решение.

Пусть $\left\{\theta'_{ijk}\right\}, (ijk) \in R_{q-1}$, решение системы (57).

Введем следующее подмножество индексных элементов: $R_{q-1}^{-} = \left\{(ijk) : \theta'_{ijk} < 0\right\}, R_{q-1}^{-} \subset R_{q-1}$.

Вычислим отношения $x_{ijk}^{(q-1)}/\Theta'_{ijk}$ для всех (ijk), для которых $\Theta'_{ijk} < 0$, найдем наибольшее среди этих отношений $x_{i_1 j_1 k_1}^{(q-1)}/\Theta'_{i_1 j_1 k_1}$ и обозначим его $\Theta_0^{(q)}$. Таким образом, получим (58)

$$\frac{x_{i_1 j_1 k_1}^{(q-1)}}{\Theta'_{i_1 j_1 k_1}} = \Theta_0^{(q)} = \max\left\{\frac{x_{ijk}^{(q-1)}}{\Theta'_{ijk}}\right\}, (ijk) \in R_{q-1} \tag{58}$$

Определим, далее, значения компонент нового опорного плана $X^{(q)} = \left\{x_{ijk}^{(q)}\right\}$ по формуле (59)

$$x_{ijk}^{(q)} = x_{ijk}^{(q-1)} - \Theta_{ijk}^{(q)}\Theta_0^{(q)}, \tag{59}$$

где $\Theta_{ijk}^{(q)} = \begin{cases} 1, (ijk) = (i^* j^* k^*); \\ \Theta'_{ijk}, (ijk) \in R_{q-1}; \\ 0, (ijk) \notin \left\{R_{q-1} \cup (i^* j^* k^*)\right\}. \end{cases}$

В соответствии с (59) компонента $x_{ijk}^{(q)}$ нового опорного плана $\{x_{ijk}^{(q)}\}$ равна соответствующие компоненте старого плана $\{x_{ijk}^{(q-1)}\}$, елси индексный элемент (ijk) не является существенным. Значение переменной $x_{i^* j^* k^*}^{(q)}$, вводимой в базис, положим равным $\Theta_0^{(q)}$. Значения остальным компонент изменяются (увеличиваются или уменьшаются в зависимости от знака Θ'_{ijk}) на величину $\Theta_0^{(q)}\left|\Theta'_{ijk}\right|$. В результате проделанных операций в новом опорном плане $X^{(q)} = \{x_{ijk}^{(q)}\}$ появляется новая положительная компонента $x_{i^* j^* k^*}^{(q)} = \left|\Theta_0^{(q)}\right| > 0$ и одновременно исключается переменная с индексным элементом $(i_1 j_1 k_1)$, для которого достигается максимум в соотношении (58), при этом $R_q = (i^* j^* k^*) \cup R_{q-1}/(i_1 j_1 k_1)$.

После выполнения второго шага осуществляется переход к следующей, $(q+1)$-й итерации, которая выполняется аналогично. Итерации повторяются до тех пор, пока не будет выполнен критерий оптимальности (55).

2.5 Аналитическое решение трипланарной транспортной задачи

Решим методом потенциалов следующую трипланарную транспортную задачу: определить набор

$$X^* = \{x_{111}^*, x_{112}^*, x_{121}^*, x_{122}^*, x_{131}^*, x_{132}^*, x_{211}^*, x_{212}^*, x_{221}^*, x_{222}^*, x_{231}^*, x_{232}^*, \},$$

минимизирующий целевую функцию

$$L(X) = 5x_{111} + 32x_{112} + 12x_{121} + 10x_{122} + 11x_{131} + 4x_{211} + 9x_{212} + 21x_{221} + 5x_{222} + 13x_{231} + 8x_{232}$$

и удовлетворяющий ограничениям:

$$\begin{cases} x_{111} + x_{112} + x_{121} + x_{122} + x_{131} + x_{132} = 1 \\ x_{211} + x_{212} + x_{221} + x_{222} + x_{231} + x_{232} = 8 \\ x_{111} + x_{112} + x_{211} + x_{212} = 3 \\ x_{121} + x_{122} + x_{221} + x_{222} = 2 \\ x_{131} + x_{132} + x_{231} + x_{232} = 4 \\ x_{111} + x_{121} + x_{131} + x_{211} + x_{221} + x_{231} = 2 \\ x_{112} + x_{122} + x_{132} + x_{212} + x_{222} + x_{232} = 7 \\ x_{ijk} \geq 0, i = 1,2; j = 1,2,3; k = 1,2 \end{cases}$$

Схематическая запись условий задачи показа на рисунке 4.

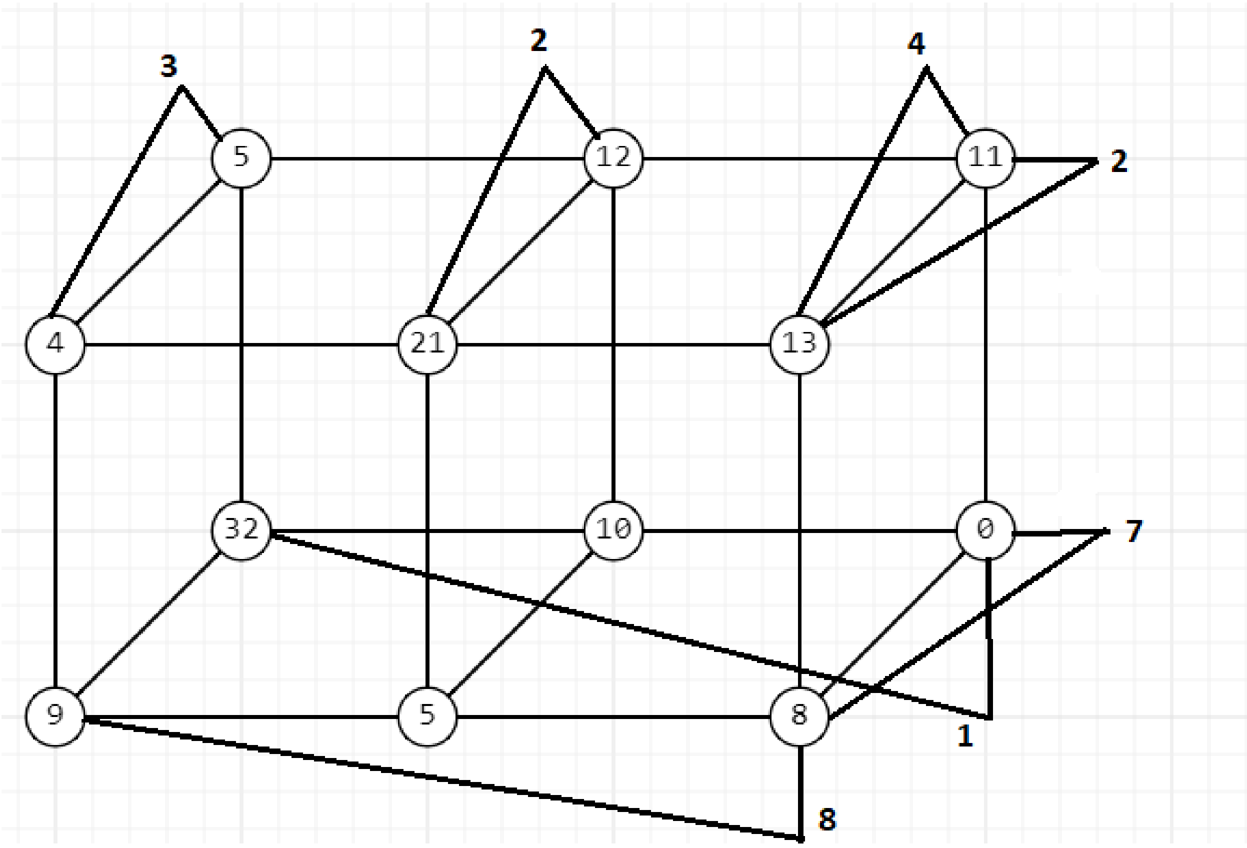

Рисунок 4 – схема условий задачи

Предварительный этап $(q = 0)$. Определим методом последовательного распределения начальный опорный план $X^{(0)}$. Шаг 1. Выбираем ведущий элемент x_{111} и определяем $x_{111} = d_1 = \min\{a_1, b_1, c_1\} = \min\{1,3,2\} = 1$.

Пересчитываем значения элементов вектора ограничений:

$$a_1^{(1)} = a_1 - d_1 = 1 - 1 = 0; b_1^{(1)} = b_1 - d_1 = 3 - 1 = 2; c_1^{(1)} = c_1 - d_1 = 2 - 1 = 1.$$

Остальные компоненты вектора ограничений переписываются без изменения. Так как $d = a_1$, то $i_2 = 2, j_2 = 1, k_2 = 1$.

Вводим элемент $x_{111} = 1$ в матрицу T_x, полагая остальные элементы сечения равными нулю.

Шаг 2. Выбираем новый ведущий элемент x_{211}. При этом

$$x_{211} = d_2 = \min\{a_2, b_1, c_1\} = \min\{8,2,1\} = 1.$$

$$a_2^{(2)} = 8 - 1 = 7;\ b_1^{(2)} = 2 - 1 = 1; c_1^{(2)} = 1 - 1 = 0;\ i_3 = 2; j_3 = 1; k_3 = 2.$$

Элемент $x_{211} = 1$ вводится в матрицу T_x, полагая остальные элементы сечения равными нулю.

Следующие расчеты производятся аналогично, поэтому ниже без пояснений.

Шаг 3.

$$x_{212} = \min\{a_2, b_1, c_2\} = \min\{7,1,7\} = 1;$$

$$a_2^{(3)} = 7 - 1 = 6;\ b_1^{(3)} = 1 - 1 = 0; c_2^{(3)} = 7 - 1 = 6;\ i_4 = 2; j_4 = 2; k_4 = 2.$$

Шаг 4.

$$x_{222} = \min\{a_2, b_2, c_2\} = \min\{6,2,6\} = 2;$$

$$a_2^{(4)} = 6 - 2 = 4;\ b_2^{(4)} = 2 - 2 = 0; c_2^{(4)} = 6 - 2 = 4;\ i_5 = 2; j_5 = 3; k_5 = 2.$$

Шаг 5.

$$x_{232} = \min\{a_2, b_3, c_2\} = \min\{4,4,4\} = 4;$$

$$a_2^{(5)} = 4 - 4 = 0;\ b_3^{(5)} = 4 - 4 = 0; c_3^{(5)} = 4 - 4 = 0.$$

Решение закончено. Получен план, содержащий ровно $m + n + p - 2 = 5$ положительных компонент. Процедура построения опорного плана изображена в таблице 3.

Таблица 3 – процедура построения опорного плана

j / i	1	2	3				$a_i^{(0)}$	$a_i^{(1)}$	$a_i^{(2)}$	$a_i^{(3)}$	$a_i^{(4)}$	$a_i^{(5)}$
1	1 / 0	0 / 0	0 / 0				1	0	0	0	0	0
2	1 / 1	0 / 2	0 / 4				8	8	7	6	4	0
$b_j^{(0)}$	3	2	4	$c_k^{(0)}$	2	7						
$b_j^{(1)}$	2	2	4	$c_k^{(1)}$	1	7	Шаг 1					
$b_j^{(2)}$	1	2	4	$c_k^{(2)}$	0	7	Шаг 2					
$b_j^{(3)}$	0	2	4	$c_k^{(3)}$	0	6	Шаг 3					
$b_j^{(4)}$	0	0	4	$c_k^{(4)}$	0	4	Шаг 4					
$b_j^{(5)}$	0	0	0	$c_k^{(5)}$	0	0	Шаг 5					

Методом последовательного распределения был определен начальный опорный план $X^{(0)}$, запишем его в таблицу 4 ($T_x^{(0)}$).

Таблица 4 – $T_x^{(0)}$

	1		2		3		1	2	a_i
1	1		0		0		1		1
		0		0		0		0	
2	1		0		0		1		8
		1		2		4		7	
b_j	3		2		4		2	7	c_k

В таблицу 5 ($T_c^{(0)}$) запишем значения коэффициентов целевой функции задачи.

Таблица 5 – $T_c^{(0)}$

									u_i
	5		12		11				
		32		10		0			
	4		21		13				
		9		5		8			
v_j									w_k

Отметим в таблице $T_c^{(0)}$ элементы $c_{ijk}^{(0)}$, соответствующие положительным переменных $x_{ijk}^{(0)}$. Значение целевой функции $L\left(X^{(0)}\right) = 1 * 5 + 1 * 4 + 1 * 9 + 2 * 5 + 4 * 8 = 60$. Множество существенных индексных элементов $R_0 = \{(1\ 1\ 1), (2\ 1\ 1), (2\ 1\ 2), (2\ 2\ 2), (2\ 3\ 2)\}$.

Основной этап. Итерация 1 $(q = 1)$.

Шаг 1. Проверим план $X^{(0)}$ на оптимальность, для чего решим систему уравнений:

$$\begin{cases} u_1 + v_1 + w_1 = 5; \\ u_2 + v_1 + w_1 = 4; \\ u_2 + v_1 + w_2 = 9; \\ u_2 + v_2 + w_2 = 5; \\ u_2 + v_3 + w_2 = 8. \end{cases}$$

Пусть $\widetilde{v_1} = 0$ и $\widetilde{w_1} = 0$. Тогда $\widetilde{u_1} = 5$; $\widetilde{u_2} = 4$; $\widetilde{w_2} = 9 - 4 = 5$; $\widetilde{v_2} = 5 - 4 - 5 = -4$; $\widetilde{v_3} = 8 - 4 - 5 = -1$. Занесем полученное решение в таблицу 6 ($T_c^{(0)}$)

Таблица 6 – решение системы.

									u_i
	5		12		11				5
		32		10		0			
	4		21		13				4
		9		5		8			
v_j	0		-4		-1		0	5	w_k

В соответствии с (55) вычислим разности

$$c_{111}^{(1)} = 5 - (5 + 0 + 0) = 0;\ c_{112}^{(1)} = 32 - (5 + 0 + 5) = 22;$$

$$c_{121}^{(1)} = 12 - (5 - 4 + 0) = 11;\ c_{122}^{(1)} = 10 - (5 - 4 + 5) = 4;$$

$$c_{131}^{(1)} = 11 - (5 - 1 + 0) = 7;\; c_{132}^{(1)} = 0 - (5 - 1 + 5) = -9;$$

$$c_{211}^{(1)} = 4 - (4 + 0 + 0) = 0;\; c_{212}^{(1)} = 9 - (4 + 0 + 5) = 0;$$

$$c_{221}^{(1)} = 21 - (4 - 4 + 0) = 21;\; c_{222}^{(1)} = 5 - (4 - 4 + 5) = 0;$$

$$c_{231}^{(1)} = 13 - (4 - 1 + 0) = 10;\; c_{232}^{(1)} = 8 - (4 - 1 + 5) = 0$$

и занесем их в таблицу 7 ($T_c^{(1)}$).

Таблица 7 – $T_c^{(1)}$

									u_i
	0		11		7				-9
		22		4		-9			
	0		21		10				0
		0		0		0			
v_j	0		0		0		0	0	w_k

План $X^{(0)}$ не является оптимальным, так как $c_{132}^{(1)} = -9 < 0$. Переходим ко второму шагу итерации.

Шаг 2. В соответствии с (56)

$$\min\left\{c_{ijk}^{(1)}\right\} = \min\{0,22,11,4,7,-9,0,0,21,0,10,0\} = -9 = c_{132}^{(1)}$$

Таким образом, $(i^* j^* k^*) = (1\ 3\ 2)$.

Решим систему уравнений (57), которая в данном случае имеет вид

$$\begin{cases} \theta_{111} = -1; \\ \theta_{111} + \theta_{212} + \theta_{222} + \theta_{232} = 0; \\ \theta_{111} + \theta_{211} + \theta_{212} = 0; \\ \theta_{222} = 0; \\ \theta_{232} = -1; \\ \theta_{111} + \theta_{211} = 0; \\ \theta_{212} + \theta_{222} + \theta_{232} = -1. \end{cases}$$

Решение системы выглядит так: $\theta'_{111} = -1; \theta'_{222} = 0; \theta'_{232} = -1; \theta'_{211} = 1; \theta'_{212} = 0$.

Определяем $\theta_0^{(1)}$ согласно (58), $R_0^- = \{(1\ 1\ 1), (2\ 3\ 2)\}$:

$$\theta_0^{(1)} = \max\left\{\frac{x_{111}^{(0)}}{\theta'_{111}}, \frac{x_{232}^{(0)}}{\theta'_{232}}\right\}, (ijk) \in R_0^- = \{-1, -4\} = -1.$$

Таким образом, $(i_1 j_1 k_1) = (1\ 1\ 1)$.

Вычислим значения компонент нового плана $X^{(1)}$:

$$\Theta_{ijk}^{(1)} = \begin{cases} 1, (ijk) = (1\ 3\ 2); \\ \Theta'_{ijk}, (ijk) \in R_0; \\ 0, (ijk) \notin R_0 \text{ и } (ijk) \neq (\ 1\ 3\ 2). \end{cases}$$

Изменению в старом плане $X^{(0)}$ подвергаются только следующие четыре компоненты:

$$x_{132}^{(1)} = x_{132}^{(0)} - \Theta_{132}^{(1)}\theta_0^{(1)} = 0 + 1 = 1; x_{111}^{(1)} = x_{111}^{(0)} - \Theta_{111}^{(1)}\theta_0^{(1)} = 1 - 1 = 0;$$

$$x_{211}^{(1)} = x_{211}^{(0)} - \Theta_{211}^{(1)}\theta_0^{(1)} = 1 + 1 = 2; x_{232}^{(1)} = x_{232}^{(0)} - \Theta_{232}^{(1)}\theta_0^{(1)} = 4 - 1 = 3.$$

Остальные компоненты плана остаются без изменения.

Впишем новый план в таблицу $T_x^{(1)}$ и вычислим соответствующее ему значение целевой функции $L(X^{(1)}) = 0 * 1 + 4 * 2 + 9 * 1 + 5 * 2 + 8 * 3 = 51$.

Как видно в таблице 8, при переходе к новому опорному плану ограничения не нарушились, а значение целевой функции уменьшилось:

Таблица 8 – $T_x^{(1)}$

	1		2		3		1	2	a_i
1	0		0		0		0		1
		0		0		1		1	
2	2		0		0		2		8
		1		2		3		6	
b_j	3		2		4		2	7	c_k

$$L(X^{(1)}) = 51 < L(X^{(0)}) = 60.$$

На этом итерация заканчивается.

Итерация 2 ($q = 2$).

Шаг 1. Проверим полученный опорный план $X^{(1)}$ на оптимальность, для чего решим систему уравнений:

$$\begin{cases} u_1 + v_3 + w_2 = -9; \\ u_2 + v_1 + w_1 = 0; \\ u_2 + v_1 + w_2 = 0; \\ u_2 + v_2 + w_2 = 0; \\ u_2 + v_3 + w_2 = 0. \end{cases}$$

Пусть $\widetilde{v_3} = 0$ и $\widetilde{w_2} = 0$. Тогда $\widetilde{u_1} = -9$; $\widetilde{u_2} = 0$; $\widetilde{v_2} = 0$; $\widetilde{v_1} = \widetilde{w_1} = 0$. Занесем полученное решение в таблицу $T_c^{(1)}$ и вычислим разности, как этого требует соотношение (55):

$$c_{111}^{(2)} = 0 - (-9 + 0 + 0) = 9 \geq 0; c_{112}^{(2)} = 22 + 9 = 31 \geq 0;$$

$$c_{121}^{(2)} = 11 + 9 = 20 \geq 0; c_{122}^{(2)} = 4 + 9 = 13 \geq 0; c_{131}^{(2)} = 7 + 9 = 16 \geq 0;$$

$$c_{132}^{(2)} = -9 + 9 = 0 \geq 0; c_{211}^{(2)} = 0 \geq 0; c_{212}^{(2)} = 0 \geq 0; c_{231}^{(2)} = 10 \geq 0; c_{232}^{(2)} = 0 \geq 0.$$

Убеждаемся, что все разности $c_{ijk}^{(2)}, (ijk) \in E$, неотрицательны и, следовательно, полученный план $X^{(1)}$ является оптимальным; $X^{(1)} = X^*$. Задача решена.

Выводы по 2 главе.

Во второй главе сформулирована и построена математическая модель трипланарной транспортной задачи. Математическая модель задачи включает в себя целевую функцию и ограничения. Решение задачи для полноты восприятия иллюстрируется графически: трехиндексная матрица представлена в виде трехмерной числовой решетки.

В главе также рассматриваются свойства задачи Т-3Р, определяется допустимый план открытой и сбалансированной транспортных задач, формулируется критерий оптимальности. Для решения трипланарной транспортной задачи приводится построение начального опорного плана, а также метод потенциалов, с помощью которого данный опорный план последовательно улучшается.

Принцип работы алгоритма построения начального опорного плана и алгоритма метода потенциалов продемонстрированы на конкретном примере решения задачи Т-3Р.

Глава 3. Программная реализация метода потенциалов для решения трипланарной транспортной задачи

3.1 Структура программы

В предыдущей главе был рассмотрен метод последовательного распределения, который позволяет определить опорный план трипланарной транспортной задачи и метод потенциалов, с помощью которого можно данный план последовательно улучшать. Процесс улучшения плана методом потенциалов состоит из конечного числа однотипных итераций, следовательно на основе алгоритма метода потенциалов, можно написать программу. Данный алгоритм реализован на языке С++ в среде Code::Blocks.

Структура программы включается в себя:

– подключенная библиотека iostream;
– функция вывода исходных данных;
– функция определения опорного плана;
– функция вывода опорного плана;
– функция вычисления значения целевой функции;
– функция решения системы u, v, w;
– функция вычисления разностей c_{ijk}^{q};
– функция проверки на оптимальность;
– функция поиска минимума;
– функция решения системы Θ_{ijk};
– функция составления нового плана $X^{(q)}$;
– функция main.

Работа программы начинается с обработки глобальных переменных, а именно количество поставщиков, количество потребителей и количество транспортных средств, которые вводятся вне функции main (рисунок 5).

```
//количество поставщиков
const int m = 2;
//количество потребителей
const int n = 3;
//количество транспортных средств
const int p = 2;
```

Рисунок 5 – глобальные переменные

Затем обрабатываются исходные данные, которые задаются в функции main (рисунок 6). В данном случае, исходные данные были взяты из задачи, аналитическое решение которой представлено ранее. Массив $a[m]$ представляет собой количество запасов, соответствующих номеру поставщика. Массив $b[n]$ – число потребностей, соответствующих номеру поставщика. Массив $c[p]$ – количество груза, которое может перевести определенное транспортное средство. Матрица $T[m][n][p]$ задающая тарифы грузоперевозок.

```
//запасы
    int *a = new int [m];
    a[0] = 1;
    a[1] = 8;

//потребности
    int *b = new int [n];
    b[0] = 3;
    b[1] = 2;
    b[3] = 4;

//возможность перевозки
    int *c = new int [p];
    c[0] = 2;
    c[1] = 7;

//стоимость перевозок
    int T[m][n][p];
    T[0][0][0] = 5;
    T[0][0][1] = 32;
    T[0][1][0] = 12;
    T[0][1][1] = 10;
    T[0][2][0] = 11;
    T[0][2][1] = 0;
    T[1][0][0] = 4;
    T[1][0][1] = 9;
    T[1][1][0] = 21;
    T[1][1][1] = 5;
    T[1][2][0] = 13;
    T[1][2][1] = 8;
```

Рисунок 6 – исходные данные

Далее производится вывод исходных данных в консоль с помощью функции show_source_data (рисунок 7).

```
void show_source_data(int T[m][n][p])
{
    cout<< "Исходные данные:" <<endl;
    cout<< "Количество поставщиков = "<<m<<endl;
    cout<< "Количество потребителей = "<<n<<endl;
    cout<< "Количество транспортных средств = "<<p<<endl;
    cout<< "Таблица стоимости грузоперевозок T:"<<endl;
    for (int i=0; i<m; i++)
    {
        for (int k=0; k<p; k++)
        {
            for(int j=0; j<n; j++)
            {
                cout<< T[i][j][k] << " ";
            }
            cout<<endl;
            if (k!=p-1)
                cout<<" ";
        }
    }
}
```

Рисунок 7 – функция show_source_data

После вывода исходных данных выполняется функция определения опорного плана. Рассмотрим реализацию метода последовательного распределения подробнее.

3.2 Реализация метода последовательного распределения

Метод последовательного распределения реализован в функции op_plan (рисунок 8).

```
void op_plan(int Z, int *i_z, int *j_z, int *k_z, int x[m][n][p],int a[m],int b[n],int c[p])
{
    int d[m*n*p-2];
    int i=0;
    int j=0;
    int k=0;
    int count = 0;
    while ( count < (Z-1) )
    {
        d[i+j+k]=a[i];
        i_z[count]=i;
        j_z[count]=j;
        k_z[count]=k;
        if (d[i+j+k]>b[j])
            d[i+j+k]=b[j];
        if (d[i+j+k]>c[k])
            d[i+j+k]=c[k];
        x[i][j][k]=d[i+j+k];
        a[i] = a[i]-d[i+j+k];
        b[j] = b[j]-d[i+j+k];
        c[k] = c[k]-d[i+j+k];

        if(a[i]==0)
            i++;
        if(b[j]==0)
            j++;
        if(c[k]==0)
            k++;
        count++;
    }
    //последняя итерация
    d[m+n+p-3] = a[m-1];
    i_z[Z-1]=m-1;
    j_z[Z-1]=n-1;
    k_z[Z-1]=p-1;
    x[m-1][n-1][p-1] = d[m+n+p-3];
    a[m-1]=0;
    b[n-1]=0;
    c[p-1]=0;
    cout<<"Множество существенных индексных элементов:"<<endl;
    for(int i=0; i<m+n+p-2; i++)
    {
        cout<<i_z[i]<<" "<<j_z[i]<<" "<<k_z[i]<<endl;
    }
}
```

Рисунок 8 – функция op_plan

Данная функция принимает на вход переменную $Z = m + n + p - 2$ – число положительных компонент плана. Массивы $i\_z[m]$, $j\_z[n]$, $k\_z[p]$, которые заполняются номерами существенных индексных элементов. Трехмерный массив $x[m][n][p]$, который, в результате работы функции, будет представлять опорный план. Массивы $a[m]$, $b[n]$, $c[p]$ – исходные данные задачи.

В цикле while производится выбор ведущего элемента, определяется минимум из трех элементов вектора ограничений, затем происходит пересчет значения элементов вектора ограничений. Ведущий элемент вводится в трехмерный массив $x[m][n][p]$. После цикла while производится вывод множества существенных индексных элементов.

Функция show_op_plan выводит значения опорного плана в консоль (рисунок 9).

```
void show_op_plan(int x[m][n][p])
{
    cout<< "Опорный план:" <<endl;
    for (int i=0; i<m; i++)
    {
        for (int k=0; k<p; k++)
        {
            for(int j=0; j<n; j++)
            {
                cout<< x[i][j][k] << " ";
            }
            cout<<endl;
            if (k!=p-1)
                cout<<" ";
        }
    }

}
```

Рисунок 9 – функция show_op_plan

Данная функция принимает на вход трехмерный массив $x[m][n][p]$, который был заполнен функцией op_plan и в цикле for выводит значения данного массива в консоль. Для того, чтобы значения массива $x[m][n][p]$ не выводились в консоль беспорядочно, были добавлены следующие строчки

$cout \ll endl;$

$if\ (k! = p - 1)$

$cout \ll "\ ";$

Данное решение позволяет выводить значения массива $x[m][n][p]$ в виде таблицы переменных x_{ijk}, структура которой была рассмотрена ранее.

Опорный план определен, теперь на его основе можно вычислить значение целевой функции $L(X)$. Вычисление происходит в функции L_foo (рисунок 10).

```
int L_foo(int x[m][n][p], int T[m][n][p])
{
    int L = 0;
    for (int k=0; k<p; k++)
    {
        for (int i=0; i<m; i++)
        {
            for(int j=0; j<n; j++)
            {
                L+=x[i][j][k]*T[i][j][k];
            }
        }
    }
    cout<< "Значение целевой функции при таком плане = "<<L<<endl<<endl;
    return L;
}
```

Рисунок 10 – функции L_foo

В данной функции происходит суммирование произведения массивов $x[i][j][k]$ и $T[i][j][k]$, т.е. произведение массива, в котором хранится опорный план на массив тарифов грузоперевозок. Итогом работы функции является значение функции $L(X)$ и вывод этого значения в консоль. Еще эта функция участвует в подсчете значения функции $L(X)$ после каждой итерации метода потенциалов, который направлен на улучшение опорного плана.

3.3 Реализация метода потенциалов

Для наглядности, представим реализацию метода потенциалов в виде блок-схемы (рисунок 11).

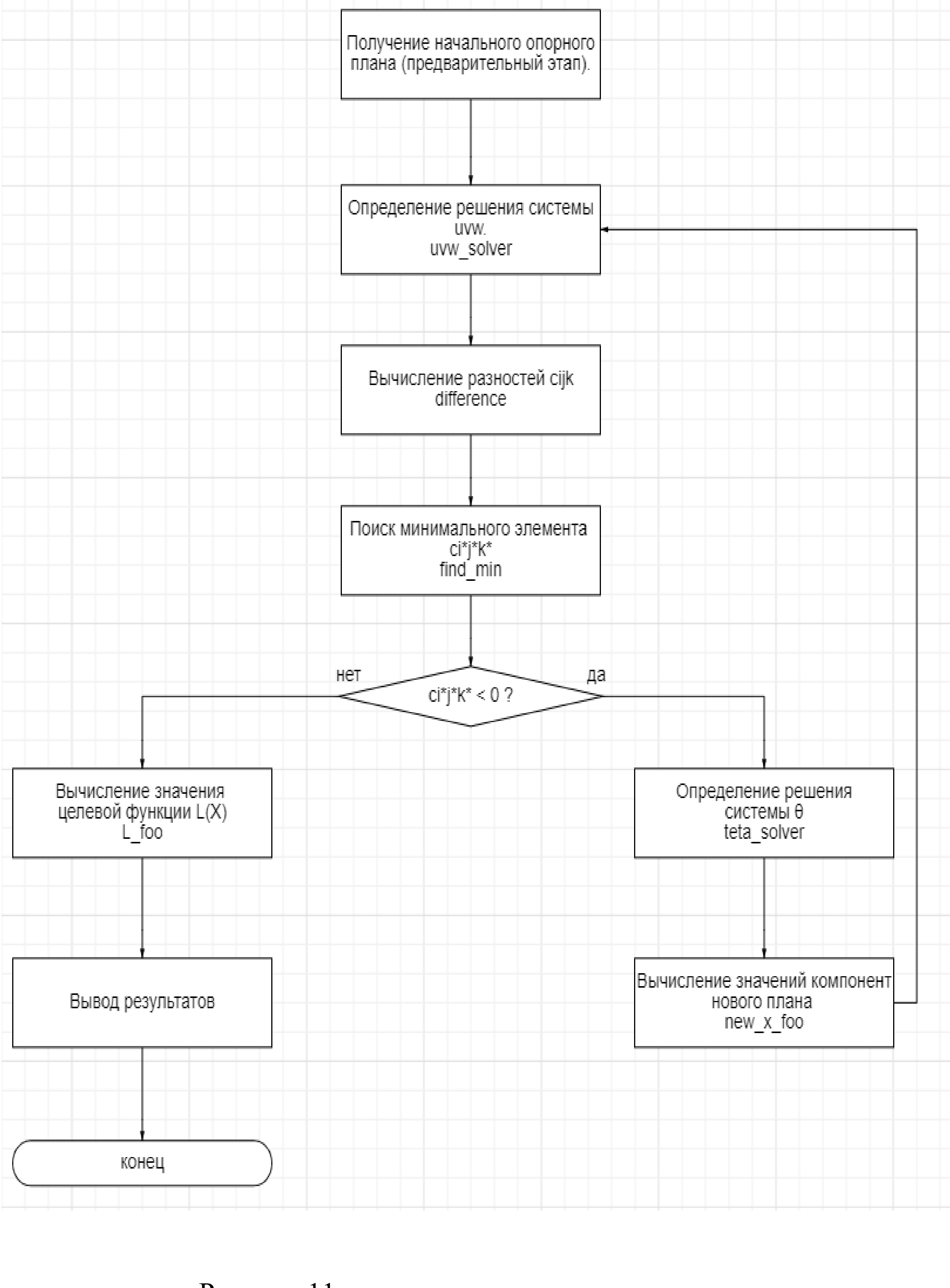

Рисунок 11 – реализация метода потенциалов

Для проверки оптимальности плана решается система линейных уравнений (54) в функции uvw_solver (рисунок 12).

```
void uvw_solver(int u[m], int v[n], int w[p], int T[m][n][p],int i_z[],int j_z[],int k_z[], int q,int Tc[m][n][p])
{
    int count = 0;
    int line=0;
    int col=0;
    int height=0;
    while(count<n+p+m-2)
    {
        while (count<m)
        {
            line = i_z[count];
            col = j_z[count];
            height = k_z[count];
            if(q == 1)
                u[line]=T[line][col][height];
            else
                u[line]=Tc[line][col][height];
            count++;
        }

        while(count<p+m-1)
        {
            line = i_z[count];
            col = j_z[count];
            height = k_z[count];
            if(q==1)
                w[height]=T[line][col][height]-u[line];
            else
                w[height]=Tc[line][col][height]-u[line];
            count++;
        }
        line = i_z[count];
        col = j_z[count];
        height = k_z[count];
        if (q == 1)
            v[col]=T[line][col][height]-u[line]-w[height];
        else
            v[col]=Tc[line][col][height]-u[line]-w[height];
        count++;
    }
```

Рисунок 12 – функция uvw_solver

Данная функция принимает на вход массивы $u[m], v[n], w[p]$, которые задаются в теле функции $main$, так как в данных массивах будут хранится вычисленные значения предварительных потенциалов. Массивы $i\_z[\,]$, $j\_z[\,]$, $k\_z[\,]$, которые задаются в теле функции $main$, они будут хранить номера существенных переменных, которые будут использоваться в дальнейшем. Переменная q, номер итерации. Если данная итерация не первая, то значения для вычисления предварительных потенциалов будут браться не из исходной матрицы $T[m][n][p]$, а из новой матрицы $Tc[m][n][p]$, вычисление которой произойдет позже.

В результате работы данной функции получаются предварительные потенциалы, которые необходимы для вычислений разностей (55).

Вычисление разностей (55) производится в функции difference (рисунок 13).

```
int difference(int T[m][n][p],int u[], int v[], int w[],int k_opt, int Tc[m][n][p],int q)
{
    cout<<"Вычисление разностей"<<endl;
    k_opt = 0;
    for (int i=0; i<m; i++)
    {
        for (int j=0; j<n; j++)
        {
            for (int k=0; k<p; k++)
            {
                if (q==1)
                    Tc[i][j][k] = T[i][j][k] - (u[i]+v[j]+w[k]);
                else
                    Tc[i][j][k] = Tc[i][j][k] - (u[i]+v[j]+w[k]);
                cout<<"Tc["<<i<<"]["<<j<<"]["<<k<<"] = "<<Tc[i][j][k]<<endl;
                if (Tc[i][j][k]<0)
                    k_opt++;
            }
        }
    }
    return k_opt;
}
```

Рисунок 13 – функция difference

На вход данной функции поступает переменная q, номер итерации. Если данная итерация первая, то производится вычисление матрицы $Tc[m][n][p]$, иначе матрица $Tc[m][n][p]$ обновляет свои значения. Также на вход поступают массивы $u[\,], v[\,], w[\,]$, вычисленные ранее. Переменная q_opt – критерий оптимальности, от значения которой зависит вывод из цикла в функции $main$.

Далее осуществляется поиск минимума среди всех разностей $c_{ijk}^{(q)}$ функцией find_min и вывод минимальной разности с записью существенного индекса (рисунок 14).

```
int find_min(int Tc[m][n][p], int min, int *l, int *c, int *h)
{
    for (int i=0; i<m; i++)
    {
        for (int j=0; j<n; j++)
        {
            for (int k=0; k<p; k++)
            {
                if (min > Tc[i][j][k])
                {
                    min = Tc[i][j][k];
                    (*l) = i;
                    (*c) = j;
                    (*h) = k;
                }

            }
        }
    }
    cout<<"Tc min = "<< min<<endl;
    cout<<"i*j*k* = "<<(*l)<<" "<<(*c)<<" "<<(*h)<<endl;
    return min;

}
```

Рисунок 14 – функция find_min

Затем проводится проверка на оптимальность функцией check_opt (рисунок 15).

```
void check_opt(int k_opt, int Tc[m][n][p])
{
    cout<<"Проверка на оптимальность "<<endl;
    int i_z2[k_opt];
    int j_z2[k_opt];
    int k_z2[k_opt];
    int line2;
    int col2;
    int height2;

    cout<<"План не является оптимальным, так как Tc[";
    for (int i=0; i<m; i++)
    {
        for (int j=0; j<n; j++)
        {
            for (int k=0; k<p; k++)
            {
                if (Tc[i][j][k] < 0)
                {
                    i_z2[i] = i;
                    line2 = i_z2[i];
                    j_z2[j] = j;
                    col2 = j_z2[j];
                    k_z2[k] = k;
                    height2 = k_z2[k];
                    cout<<i<<"]["<<j<<"]["<<k<<"]"<<" < 0"<<endl;
                }
            }
        }
    }
}
```

Рисунок 15 – функция check_opt

Функция проверки оптимальности достаточно проста в реализации, так как нужно всего лишь проверить весь трехмерный массив $Tc[i][j][k]$ на неотрицательность, если среди элементов данного массива найдется хотя бы один отрицательный, то план не является оптимальным.

Если план не оптимален, то начинает работу функция teta_solver (рисунок 16), которая определяет решение системы (57).

```
int teta_solver(int i_z[], int j_z[], int k_z[], int teta[m][n][p], int teta0, int col2, int line2, int x[m][n][p])
{
    string sign[m][n][p];
    int line;
    int col;
    int height;

    for (int i=0; i<m+n+p-2; i++)
    {
        line = i_z[i];
        col = j_z[i];
        height = k_z[i];
        if((i % 2) != 0 )
        {
            sign[line][col][height] = "+";
            if (sign[line-1][col][height] == "-")
            {
                teta[line][col][height] = teta[line-1][col][height] * (-1);
            }
        }
        if (((i%2)==0)&&((col == col2) || (line == line2) ) )
        {

            sign[line][col][height] = "-";
            teta[line][col][height] = -1;
            if(teta0 == 0)
                teta0 = teta[line][col][height] * x[line][col][height];
            if(teta0 <(teta[line][col][height] * x[line][col][height]))
                teta0 = teta[line][col][height] * x[line][col][height];
        }
    }
    return teta0;
}
```

Рисунок 16 – функция teta_solver

С помощью трехмерного массива $s[m][n][p]$, тип данных которого $string$, формируется цикл пересчета. Также вычисляется значение переменной $teta0$, заданной в функции $main$.

Далее вычисляются значения компонент нового плана функцией new_x_foo (рисунок 17). Новый план отправляется в функцию uvw_solver, тем самым начиная $q + 1$ итерацию.

```
void new_x_foo(int teta[m][n][p],int x[m][n][p], int teta0, int line2, int col2, int height2, int T[m][n][p], int i_z[m], int j_z[n], int k_z[p])
{
    int count = 0;
    for (int i=0; i<m; i++)
    {
        for (int k=0; k<p; k++)
        {
            for(int j=0; j<n; j++)
            {
                if (teta[i][j][k]!=0)
                {
                    x[i][j][k] = x[i][j][k] - teta[i][j][k]*teta0;
                }
                if ((i == line2) && (j == col2) && (k == height2))
                {
                    x[i][j][k] = x[i][j][k] - teta0;
                    i_z[0] = line2;
                    j_z[0] = col2;
                    k_z[0] = height2;
                }
                cout<<x[i][j][k];
                count++;
            }
            cout<<endl;
            if (k!=p-1)
                cout<<" ";
        }
    }
}
```

Рисунок 17 – функция new_x_foo

Итерационный процесс продолжается, пока план станет оптимальным. Если функция check_opt показала, что план оптимальный, то проводится вычисление значения целевой функции $L(X)$ функцией L_foo. В консоли выводится оптимальный план и на этом программа заканчивает работу.

Проверим работу программы на примере, решение которого было представлено ранее.

На рисунке 18 показана первая итерация.

```
Исходные данные:
Количество поставщиков = 2
Количество потребителей = 3
Количество транспортных средств = 2
Таблица стоимости грузоперевозок T:
5 12 11
 32 10 0
4 21 13
 9 5 8
Множество существенных индексных элементов:
0 0 0
1 0 0
1 0 1
1 1 1
1 2 1
Опорный план:
1 0 0
 0 0 0
1 0 0
 1 2 4

Оптимизация плана грузоперевозок методом потенциалов. Итерация (1):
Значение целевой функции при таком плане = 60

Решение системы uvw
u[0] = 5
u[1] = 4
w[0] = 0
w[1] = 5
v[0] = 0
v[1] = -4
v[2] = -1
Вычисление разностей
Tc[0][0][0] = 0
Tc[0][0][1] = 22
Tc[0][1][0] = 11
Tc[0][1][1] = 4
Tc[0][2][0] = 7
Tc[0][2][1] = -9
Tc[1][0][0] = 0
Tc[1][0][1] = 0
Tc[1][1][0] = 21
Tc[1][1][1] = 0
Tc[1][2][0] = 10
Tc[1][2][1] = 0
Проверка на оптимальность
План не является оптимальным, так как Tc[0][2][1] < 0
Tc min = -9
i*j*k* = 0 2 1
Новый план:
000
 001
200
 123
```

Рисунок 18 – первая итерация

На рисунке 19 показана вторая итерация, на которой программа завершила свою работу.

```
Оптимизация плана грузоперевозок методом потенциалов. Итерация (2):
Значение целевой функции при таком плане = 51

Решение системы uvw
u[0] = -9
u[1] = 0
w[0] = 0
w[1] = 0
v[0] = 0
v[1] = 0
v[2] = 0
Вычисление разностей
Tc[0][0][0] = 9
Tc[0][0][1] = 31
Tc[0][1][0] = 20
Tc[0][1][1] = 13
Tc[0][2][0] = 16
Tc[0][2][1] = 0
Tc[1][0][0] = 0
Tc[1][0][1] = 0
Tc[1][1][0] = 21
Tc[1][1][1] = 0
Tc[1][2][0] = 10
Tc[1][2][1] = 0
Все разности неотрицательны, следовательно, полученный план является оптимальным. Задача решена.
План:
0 0 0
 0 0 1
2 0 0
 1 2 3
является оптимальным. При таком плане значение целевой функции = 51

Process returned 0 (0x0)   execution time : 0.167 s
Press any key to continue.
```

Рисунок 19 – вторая итерация и конец работы программы

3.4 Решение трипланарной транспортной задачи в Microsoft Excel

Для решения небольших трипланарных транспортных задач можно использовать функцию «Поиск решения» в программе Microsoft Excel. На рисунке 20 продемонстрирован пример работы данной функции для решения трипланарной транспортной задачи, аналитическое решение которой представлено раннее.

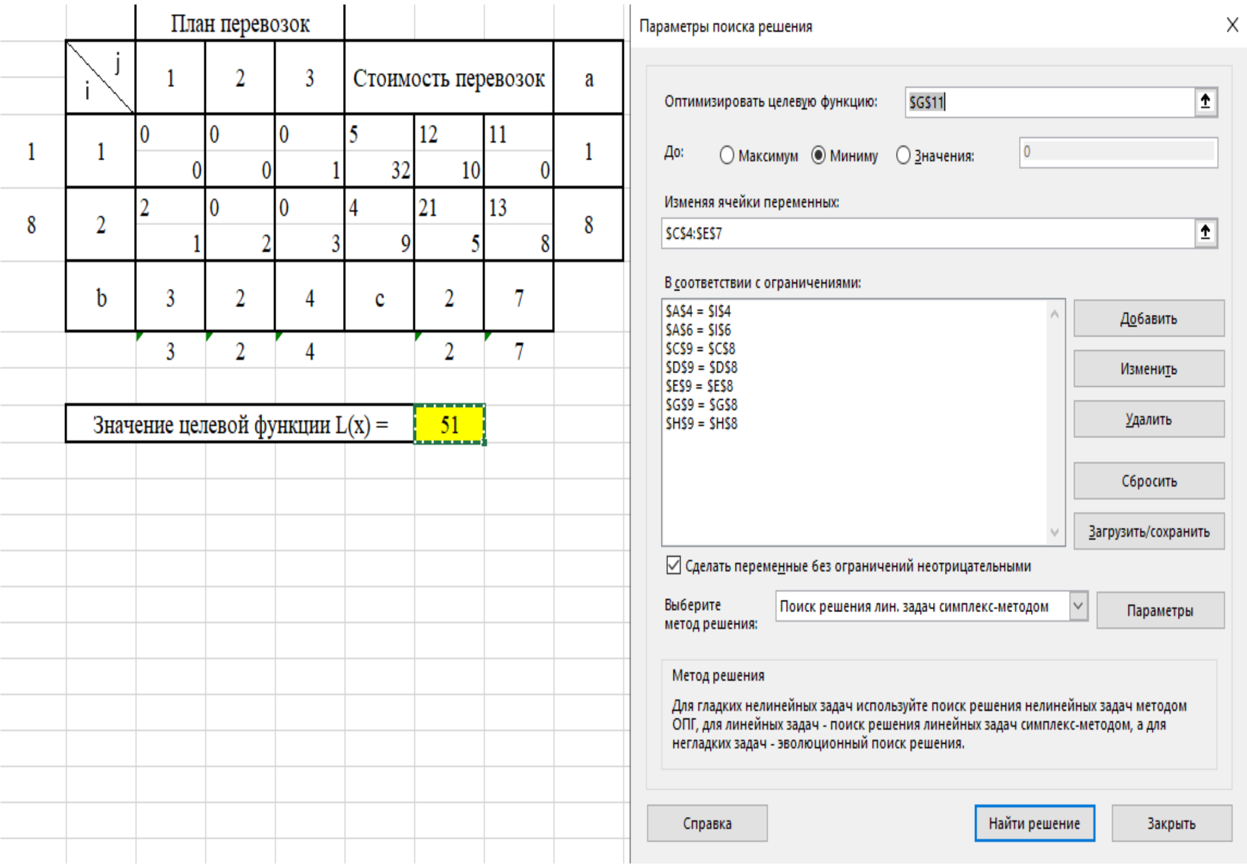

Рисунок 20 – пример работы функции «Поиск решения»

В параметрах функции «Поиск решения» задается ячейка, в которой минимизируется значение целевой функции $L(x)$. В данном случае, значение целевой функции описывается формулой =СУММПРОИЗВ(C4:E7;F4:H7), т.е. вычисляется сумма произведений матрицы опорного плана и матрицы, которая задает стоимость перевозок. Далее задается диапазон ячеек переменных опорного плана и соответствующие им ограничения.

Необходимое условие неотрицательности задается ниже, в пункте «Сделать переменные без ограничений неотрицательными». После нажатия кнопки «Найти решение» данная функция начнет подбирать значения в ячейках, диапазон которых задан раннее, симплекс-методом таким образом, чтобы получившееся значение целевой функции было минимально.

Безусловно такой метод решения не подходит для решения больших трипланарных транспортных задач, так как все данные необходимо заполнять вручную, а значит чтобы внести все данные и соответствующие ограничения для задач большой размерности потребуется огромное количество времени и внимательности, так как ошибка хотя бы в одной ячейке приведет к некорректному решению всей задачи.

Выводы по 3 главе.

Таким образом, в третьей главе рассмотрена программная реализация функций, написанных на основе алгоритмов метода последовательного распределения и метода потенциалов для решения трипланарной транспортной задачи. Структура метода потенциалов для полноты восприятия продемонстрирована в виде блок-схемы. Каждая функция программы снабжена подробными комментариями о работе данной функции, данных, поступающих на вход, и переменных, меняющих свое значение в результате выполнения функции.

В главе также продемонстрирована работа программы и функции «Поиск решения», на примере задачи, аналитическое решение которой представлено ранее.

Заключение

В ходе выполнения выпускной квалификационной работы проведено исследование, объектом которого являлись методы определения опорного плана и методы последовательного улучшения опорного плана.

Целью данной выпускной квалификационной работы было решение проблемы оптимизации в многоиндексной транспортной задаче и ее программная реализация. В ходе данной работы поставлены и выполнены следующие задачи:

- Изучены и проанализированы методы решения задач линейного программирования транспортного типа, а также метод последовательного распределения, метод потенциалов для решения многоиндексной транспортной задачи;
- Решена аналитически многоиндексная транспортная задача с использованием оптимизационных методов;
- Реализован программный модуль алгоритма решения многоиндексной транспортной задачи;
- Программный модуль протестирован на основе задачи, решенной аналитически с использованием оптимизационных методов.

В результате сделаны следующие выводы из проделанной работы:

- Для построения начального опорного плана трипланарной транспортной задачи можно использовать метод последовательного распределения;
- Используя алгоритм метода потенциалов, можно за ограниченное количество итераций улучшить опорный план до оптимального;
- Метод потенциалов для решения трипланарной транспортной задачи возможно реализовать на языке C++ с использованием стандартных библиотек.

Список используемой литературы

1. Афраймович Л.Г. Минимизация затрат при распределении однородного ресурса в иерархических системах с двусторонними ограничениями // КоГраф 2002. Материалы докладов всероссийской конференции. – Нижний Новгород. 2002. С. 100-110.

2. Булавский В.А., Звягина Р.А., Яковлева М.А. Численные методы линейного программирования. Специальные задачи. – М.: Наука. 1977. С. 50-55.

3. Афраймович Л.Г. Метод решения целочисленных многоиндексных транспортных задач с декомпозиционной структурой // Доклады Одесского семинара по дискретной математике. № 11. 2011. 210 с.

4. Герами, В. Д. Городская логистика. Грузовые перевозки: учебник для вузов / В. Д. Герами, А. В. Колик. – Москва: Издательство Юрайт, 2022. 228 с.

5. Григорьев, М. Н. Коммерческая логистика: теория и практика: учебник для вузов / М. Н. Григорьев, В. В. Ткач, С. А. Уваров. – 3-е изд., испр. и доп. – Москва: Издательство Юрайт, 2022. С. 220-230.

6. Лукинский, В. С. Логистика и управление цепями поставок : учебник и практикум для вузов / В. С. Лукинский, В. В. Лукинский, Н. Г. Плетнева. – Москва: Издательство Юрайт, 2022. 120 с.

7. Раскин Л.Г., Кириченко И.О. Многоиндексные задачи линейного программирования. – М.: Радио и связь. 1982. С. 12-17.

8. Рублев В.С., Смирнов А.В. Задача целочисленного сбалансирования трехмерной матрицы и алгоритмы ее решения // Моделирование и анализ информационных систем. 2010. Т. 17. № 2. С. 57-78.

9. Рублев В.С., Смирнов А.В. NP-полнота задачи сбалансирования трехмерной матрицы // Доклады Академии наук. 2010. Т. 435. № 3. С. 50–60.

10. Рублев В.С., Чаплыгина Н.Б. Выбор критерия оптимизации в задаче о равномерном назначении // Дискретная математика. 2005. Т. 17. Вып. 4. 29 с.

11. Сергеев С.И. Новые нижние границы для трипланарной задачи назначения. Использование классической модели // Автоматика и телемеханика. 2008. № 12. С. 105-120.

12. Сергеев С.И. Улучшенные нижние границы для решения квадратичной задачи назначения // Автоматика и телемеханика. 2004. № 11. С. 100-110.

13. Смирнов А.В. Задача целочисленного сбалансирования трехмерной матрицы и сетевая модель // Моделирование и анализ информационных систем. 2009. Т. 16. № 3. 100 с.

14. Сосина Н.А. Исследование операций. Электронное учебное пособие. В 2-х частях. Часть I/ Тольятти: ФГБОУ ВО «Тольяттинский государственный университет», 2022г. Электронный ресурс, 3,3 Мб. ISBN 978-5-8259-1045-1. С. 7-67.

15. Триус Е.Б. Задачи математического программирования транспортного типа. М.: Советское радио. 1967. С. 228 с.

16. Филлипс Д., Гарсиа-Диас А. Методы анализа сетей. – М.: Мир. 1984. С. 99-110.

17. Кравцов М.К., Кравцов В.М., Лукшин Е.В. О нецелочисленных вершинах многогранника трехиндексной аксиальной задачи о назначениях // Дискретная математика. 2001. Т. 13. Вып. 2. С. 57-64.

18. Литвак Б.Г., Раппопорт А.М. Задачи линейного программирования, допускающие сетевую постановку // Экономика и математические методы. 1970. Т. 6. Вып. 4. С. 100-110.

19. Малашенко Ю. Е., Новикова Н. М. Потоковые задачи анализа уязвимости многопродуктовых сетей. – М.: ВЦ АН СССР. 1989. С. 101-110.

20. Меламед И.И., Сигал И.Х. Вычислительное исследование алгоритмов решения бикритериальных задач дискретного программирования // Журнал вычислительной математики и математической физики. 2000. Т. 4. № 11. 30 с.

21. Ahuja R.K., Magnati T.L., Orlin J.B. Network flows: theory, algorithms, and applications. Prentice Hall. 1993. С. 122-128.

22. Alighanbari M., How J.P. Cooperative task assignment of unmanned aerial vehicles in adversarial environments // Proceedings of the American Control Conference. 2005. V. 7. P. 4661–4666. С. 32-38.

23. Amaldi E., Pfetsch M.E., Leslie E.T. On the maximum feasible subsystem problem, IISs and IIS-hypergraphs // Mathematical Programming, 2003. V. 95. N 3. 68 с.

24. Arbib C., Pacciarelli D., Smriglio S. A. A three-dimensional matching model for perishable production scheduling // Discrete Applied Mathematics. 1999. V. 92. С. 79-90.

25. Balas E., Saltzman M.J. An algorithm for the three-index assignment problem // Operations Research. 1991. V. 39. С. 123–134.