МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ 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 кафедры полностью)

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

Системное программирование и компьютерные технологии
(направленность (профиль)

БАКАЛАВРСКАЯ РАБОТА

на тему **Реализация алгоритма декомпозиции задач линейного программирования больших размерностей**

Студент А.С.Базаева
(И.О. Фамилия) (личная подпись)

Руководитель Г.А. Тырыгина
(И.О. Фамилия) (личная подпись)

Консультанты Е.В.Косс
(И.О. Фамилия) (личная подпись)

Допустить к защите

Заведующий кафедрой к.т.н., доцент, А.В. Очеповский _____
(ученая степень, звание, И.О. Фамилия) (личная подпись)

«_____»_____20_____г.

Тольятти 2018

АННОТАЦИЯ

Темой выпускной квалификационной работы является: «Реализация алгоритма декомпозиции для задач линейного программирования большой размерности»

Актуальностью данной работы являются математические модели реальных задач сводящийся к задачам линейным программирования которые характеризуются большим числом переменных.

Объект исследования: математическая модель задачи линейного программирования.

Предмет исследования: является математическая модель задачи линейного программирования с большим числом переменных.

Цель: разработка и реализация алгоритма декомпозиции для задачи линейного программирования.

Для реализации цели сформируем задачи:

1. Проанализировать теоретические основы симплекс-метода
2. Исследовать сущность декомпозиции для задачи линейного программирования.
3. Реализация декомпозиции для задачи линейного программирования.

Дипломная работа состоит из введения, трех глав и заключения.

В первой главе данной работы описана и проанализированы основы симплекс-метода.

Во второй главе изучены теоретические основы метода декомпозиции задач линейного программирования.

В третьей главе представлен алгоритм декомпозиции, рассмотрен и реализован пример.

В заключении представлены выводы и результаты проделанной работы.

Пояснительная записка выпускной квалификационной работы содержит 44 страницы, состоит из введения, трех глав, заключения и списка литературы, в который входит 35 литературных источников.

Результатом работы является реализация алгоритма декомпозиции для задач линейного программирования больших размерностей.

ABSTRACT

The title of the graduation work is "Implementation of the decomposition algorithm for linear programming problems of large dimension".

The relevance of this work is the mathematical model of real problems reduced to linear programming problems which are characterized by a large number of variables.The object of the graduation work is a mathematical model of the linear programming problem.The subject of the graduation work is a mathematical model of the linear programming problem with a large number of variables.

The aim of the work is to give some information about the development and implementation of the decomposition algorithm for the linear programming problem.

To achieve the goal, we will form the following tasks:

1. To analyze the theoretical basis of the simplex method.
2. To investigate the essence of the decomposition for the linear programming problem.
3. To Implement a decomposition algorithm for the linear programming problem.

The graduation work consists of an explanatory note on 44 pages, introduction, including 2 figures, 2 tables, the list of 35 references including 5 foreign sources and only appendice.

The graduation work consists of an introduction, three chapters and a conclusion. The first Chapter of this paper describes and analyzes the basics of the simplex method.In the second Chapter, the theoretical foundations of the method of the decomposition of linear programming problems are studied.The third

Chapter presents the decomposition algorithm, an example is considered and implemented. Finally, the conclusions and results of the work are presented.

The result of this work is the implementation of the decomposition algorithm for linear programming problems of large dimensions.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Большая размерность является исключительным свойством большинства практических задач линейного программирования. Например, решение задач оптимизированного проектирования на уровне страны матрицы ограничений достигают размерности $m=10^4$, $n=10^6$. При такой размерности классические методы линейного программирования, такие как симплекс-метод, двойственный симплекс-метод и др., оказываются малоэффективными. Таким образом, это было связано с необходимостью разработки специальных методов как приближенных, так и точных, предназначенных для решения задач высокой размерности.

Большинство этих методов используют принцип декомпозиции (принцип децентрализации). Его идея состоит в том, что большая задача разбивается на ряд задач меньшего размера, нахождении независимых оптимумов для каждой из них, а затем связывают эти частные решения в общее решение исходной задачи.

Методы блочного программирования могут быть использованы для построения эффективных алгоритмов решения специальных задач линейного программирования.

Актуальность данной работы математическая модель реальных задач сводящийся к задачам линейным программирования которые характеризуются большим числом переменных.

Объектом является математическая модель задачи линейного программирования.

Предметом является математическая модель задачи линейного программирования с большим числом переменных.

Целью является разработать и реализовать алгоритм декомпозиции для задачи линейного программирования.

Для реализации цели сформируем задачи:

1. Проанализировать теоретические основы симплекс-метода

2. Исследовать сущность декомпозиции для задачи линейного программирования.
3. Реализовать декомпозицию для задачи линейного программирования.

ГЛАВА 1 ТЕОРЕТИЧЕСКИЕ ОСНОВЫ СИМПЛЕКС-МЕТОДА

1.1. Основные понятия симплекс-метода

В отечественной литературе симплекс-метод известен как «метод последовательного улучшения плана», он был создан Данцигом в 1947 году. Непрерывному увеличению значения целевой функции способствует переход от одного допустимого базового решения к другому и все это позволяет данный метод. Оптимальное решение при данном условии лежит в конечном числе шагов. Алгоритмы симплекс-метода также позволяют установить, разрешима ли проблема линейного программирования [2].

Представим ограничения задачи линейного программирования в таком виде:

$$A_1 x_1 + A_2 x_2 + \cdots + A_{n+m} x_{n+m} = A_0.$$

Предположим, что $A_1, A_2, \ldots, A_m$ – множество линейно-независимых векторов [22].

В таком случае уравнение

$$A_1 x_1^* + A_2 x_2^* + \cdots + A_m x_m^* = A_0 \qquad (1)$$

находит базисное решение $x_1^*, x_2^*, \ldots, x_m^*$.

Положим, что данное решение допустимо, т.е. $x_1^* \geq 0, \ldots, x_m^* \geq 0$. Базис $A_1, A_2, \ldots, A_m$ образует m-мерное пространство, каждый из векторов $A_{m+1}, A_{m+2}, \ldots, A_{m+n}$ однозначно определяется эти базисом [7].

Небазисный вектор A_r определяется в виде:

$$A_r = x_{1r} A_1 + x_{2r} A_2 + \cdots + x_{mr} A_m, \qquad (2)$$

где x_{ir} это соответствующие коэффициенты ($i = 1, 2, \ldots, m$).

Дальше предположим такое, что хотя бы одна из величин $x_{ir} > 0$. Соответственно найденные решения уравнения

$$A_1 x_1 + A_2 x_2 + \cdots + A_m x_m + A_r x_r = A_0 \qquad (3)$$

будут обозначаться следующим образом, $x_1^{**}, x_2^{**}, \ldots, x_m^{**}, x_r$.

И тут очевидно, что уравнение будет выглядеть так:

$$A_1 x_1^{**} + A_2 x_2^{**} + \cdots + A_m x_m^{**} + A_r x_r = A_0. \quad (4)$$

Уравнение (2) умноженное на x_r нужно вычесть из уравнения (1) и тогда будет получено уравнение:

$$A_1 \ x_1^* - x_r x_{1r} \ + A_2 \ x_2^* - x_r x_{2r} \ + \cdots + A_m \ x_m^* - x_r x_{mr} \ = A_0 - x_r A_r. \quad (5)$$

Связь получившегося решения $x_1^{**}, \ x_2^{**}, \ldots, x_m^{**}, x_r$ со старым базисным $x_1^*, x_2^*, \ldots, x_m^*$ находим путем сравнения (5) и (4):

$$x_1^{**} = x_1^* - x_r x_{1r}, x_2^{**} = x_2^* - x_r x_{2r}, \ldots, x_m^{**} = x_m^* - x_r x_{mr}, x_r \quad (6)$$

Это решение (6), будет не базисным, затем что в ней содержится переменная m+1, и также, будет допустимым не для всех значение x_r.

Нужно выбрать x_r так, чтобы ни одна из величин $x_i^{**} = x_i^* - x_r x_{ir}$ $(i = 1, 2, \ldots, m)$ не оказалась меньше нуля, и чтобы найденное решение было допустимым [23]. Найдем соотношение по которому находится максимальное значение переменной x_r:

$$x_{imax} = \min_i \ \frac{x_i^*}{x_{ir}} \ , \quad (7)$$

и $x_{ir} > 0$.

Удалив одну переменную x_j и соответствующий вектор A_j из базиса, новое допустимое решение будет базисным. Также новое базисное решение будет включать в себя m векторов [3].

Выбираем x_r в соответствии с (7). Поэтому новое базисное решение имеет вид

$$x_1^* - x_{rmax} x_{1r};$$
$$x_2^* - x_{rmax} x_{2r};$$
$$x_j \ \text{опущен} \ ;$$

$$x_{rmax},$$

а новый базис $A_1, A_2, \ldots, A_{j+1} \ldots, A_m, A_r$.

Для большинства задач линейного программирования такой переход от одного базиса к другому дает возможность находить решения. Можно вычислить целевую функцию найдя все крайние точки, и тем самым найти

экстремальное решение. Однако при больших m и n это практически невозможно. Для перехода к таким вероятным базисным решениям, которым отвечают максимальные значения целевой функции, предложен критерий симплекс-метода [28]. Следующее значение целевой функции соответствует новому базисному решению $x_1^* - x_r x_{1r}, x_2^* - x_r x_{2r}, \dots, x_m^* - x_r x_{mr}, x_r$:

$$z_1 = c_1 \left(x_1^* - x_r x_{1r}\right) + c_2 \left(x_2^* - x_r x_{2r}\right) + \cdots + c_r x_r \qquad (8)$$
$$= c_1 x_1^* + c_2 x_2^* + \cdots + c_m x_m^*$$
$$+ x_r \left(c_r - x_{1r} c_1 - \cdots - c_m x_{mr}\right),$$

где $c_1 x_1^* + c_2 x_2^* + \cdots + c_m x_m^*$ - значение целевой функции, отвечающее исходному базисному решению;

$c_r - c_1 x_{1r} - c_r x_{2r} - \cdots - c_m x_{mr}$ – симплекс-разность для x_r.

У каждой переменной вычисляется симплекс-разность, но он не входит в базис, поэтому находится свободная переменная x_r имеющая максимально положительную симплекс-разность [8].

Теперь составим алгоритм симплекс-метода который будет состоять из следующих шагов:

1. Для начала требуется определить начальный базис и найти допустимое к нему решение.

2. Далее нужно определить для каждой переменной симплекс-разность которая не состоит в базисе [25].

3. В базис вводится переменная с максимальной симплекс-разностью, притом значение x_{rmax} находится из условия (7) для всех i, для которых $x_{ir} > 0$;

4. Затем, из базиса нужно вывести вектор A_j, а вот из базисного решение выводят переменную x_j, которая соответствует данному соотношению $min \frac{x_i^*}{x_{ir}} = \frac{x_j^*}{x_{jr}}$;

5. Критерием оптимальности базисного решения будут являться отрицательные переменные, не входящие в базис, и которые находятся с помощью симплекс-разности [1].

Итак, рассмотрим следующие теоремы симплекс-метода: Теорема 1.1. Итак, если либо основная задача, либо двойственная задача имеет конечное оптимальное решение, соответствующие значения объективных функций равны [24]. Теорема 1.2. Пусть основная задача имеет оптимальное допустимое базисное решение $x = (x_B, 0)^T$, и соответствующие столбцы в A образуют подматрицу B, а соответствующие коэффициенты затрат образуют вектор-столбец C_B. Тогда вектор $\lambda = (B-1)\,t \quad C_B$ является оптимальным решением двойственной задачи. Оптимальные значения обеих задач совпадают [29].

Метод полного исключения.

Матричная форма системы линейных уравнений

$$\sum_{j=1}^{n} a_{ij}x_j = a_{i0}, \qquad i = 1, \dots, m.$$

имеет вид:

$$Ax = A_0.$$

$A_p = [A, A_0]$ - расширенная матрица. Метод полного исключения Жордана-Гаусса приводящий матрицу А к единичному виду состоит из конечного числа однотипных шагов [27]. Метод основан на элементарных операциях Гаусса, приводящих к эквивалентной системе уравнений:

- Одна из строк расширенной матрицы перемножается на число, замечательное от нуля;

- Из каждой строки расширенной матрицы вычитается одна строка, умноженную на число.

Определим шаги метода:

1. Произвольный отличный от нуля из матрицы A – направляющий элемент шага. Строка и столбец в которых находится этот элемент – направляющий.

2. Все элементы направляющей строки расширенной матрицы делят на направляющий элемент. Затем из каждой строки матрицы А

вычитают новую направляющую строку, умноженную на элемент, расположенный на пересечении преобразуемой строки и направляющего столбца [9].

$A_p^{(1)}$ обозначим матрицу преобразованную после первого шага. В этой матрице элементы направляющего столбца которые были отлично от направляющего элемента обратились в нуль. Основной частью матрицы $A_p^{(1)}$ называется множество переменных первых n столбцов матрицы A_p, которые находятся вне направляющей строки и столбца предыдущего шага [4].

В главной части матрицы $A_p^{(1)}$среди ненулевых элементов исследуют направляющий элемент второго шага.

Остальные шаги метода также выполняются как и первый шаг, последующие шаги будут выполняться пока существует выбор направляющего элемента [30]. Процесс удаления элементов завершается, когда если после k-го шага у основной части матрицы $A_p^{(k)}$ отсутствуют элементы или она содержит нули.

К примеру, процесс будет заканчиваться после шага l. Будем предполагать, что в строках матрицы $A^{(l)}$ существуют такие элементы, которые не являлись направляющими ни в одном из шагов, допустим, строка с номером i. Из этого вывода, очевидно что $a_{ij} = 0, \quad j = 1,2,\dots,n.$

Для *i*-й строки уравнение:

$$0x_1 + 0x_2 + \cdots + 0x_n = a_{i0}^{l}. \tag{9}$$

так как

$$A^{(i)}x = a_{i0}, \tag{10}$$

и для любой строки.

При $a_{i0}^{(l)} \neq 0$, то (9) противоречиво, и неразрешима.

При $a_{i0}^{(l)} = 0$, то (9) обращается тождество, и отбрасывается *i*-я строка [33].

Когда происходит перебор строк матрицы $A^{(l)}$ которые не являются направляющими, с ними происходит следующее, либо система уравнений становится неопределенной, либо отбрасываются все нулевые строки [6].

Итак, таким образом в системе будет l уравнений. Будем полагать для точности, что это первые l уравнений.

В таком случае полученная система уравнений может быть записана в виде

$$\sum_{j=1}^{n} a_{ij}^{(l)} x_j = a_{i0}^{(l)}, \quad i = 1,2,\dots,l. \tag{11}$$

Положим, что i-й направляющий столбец соответствует i-й направляющей строке.

Тогда

$$a_{ij}^{(l)} = \begin{matrix} 0, i \neq j \\ 1, i = j \end{matrix} \quad j = 1,2,\dots,l. \tag{12}$$

Поэтому, (11) можно записать:

$$x_i = a_{i0}^{(l)} - \sum_{j=l+1}^{n} a_{ij}^{(l)} \cdot x_j, \quad i = 1,2,\dots,l. \tag{13}$$

$x_i (i = 1,2,\dots,l)$- базисным, $(x_j, j = l+1,\dots,n)$ – небазисными.

Получаем одно из базисных решений системы уравнений при $x_i = 0$ $(j = l+1,\dots,n)$:

$$x_i = a_{i_0}^{(l)}, i = 1,2,\dots,l;\ x_j = 0, \quad j = l+1,\dots,n.$$

Все множество решенией образующие $(n-1)$ – мерное подпространство решений получается, если задавать произвольные a_j для x_j.

Если x_i-i-я компонента, тогда

$$x_i = \begin{matrix} a_{i0} - \sum_{j=l+1}^{n} a_{ij}\alpha_j, \ \text{если } i = 1,\dots,l; \\ \alpha_i, \text{если } i = l+1, l+2,\dots,n. \end{matrix} \tag{14}$$

Если

$$x_0 = (a_{10}, a_{20}, \dots, a_{l0}, 0, \dots, 0),$$

$$x_j = \ -a_{1j}, -a_{2j}, \dots, -a_{ij}, 0, \dots, 1, 0, \dots, 0\ , \qquad l < j \le n,$$

то общее решение системы линейных уравнений (14) имеет вид:

$$x_{\text{общ}} = x_0 + \sum_{j=l+1}^{n} a_j x_j, \tag{15}$$

x_0 – базисное решение исходной системы; $\sum_{j=l+1}^{n} a_j x_j$, - полное решение соответствующей однородной системы уравнений, т.е. при $A_0 = 0$.

Расширенную матрицу условий после k-й итерации обозначим

$$A_p^{(k)} = \ a_{i0}^{k}, a_{i1}^{k}, \dots, a_{in}^{k}\ , \ i = 1,2,\dots,m.$$

Итак, предположим что, a_{ij}^{k} -направляющий элемент преобразования на $k+1$ $-$й итерации. Матрица, $A_p^{(k+1)}$ получающаяся после $k+1$ $-$й итерации «метода полного исключения Гаусса, элементы которые вычисляются по формулам:

1. для элементов направляющей строки

$$a_{il}^{(k+1)} = \frac{a_{il}^{(k)}}{a_{ij}^{(k)}}, l = 0,1,\dots,n; \tag{16}$$

2. для элементов направляющего столбца

$$a_{rj}^{(k+1)} = 0, \qquad r = 1,2,\dots,n, \qquad \text{причем } r \neq i;\ a_{ij}^{k+1} = 1; \tag{17}$$

3. для всех остальных элементов матрицы $A_p^{(k+1)}$

$$a_{rl}^{(k+1)} = a_{rl}^{(k)} - \frac{a_{il}^{k}}{a_{ij}^{k}} a_{rj}^{k}, \qquad l \neq j, r \neq i. \tag{18}$$

1.2. Алгоритмическая реализация симплекс-метода

Суть симплекс-метода заключается в переходе от допустимого базисного решения к другому, с целью увеличения значения целевой функции (задача максимизации) [31].

Если ограничения задачи привести к виду в котором матрице A есть единичная подматрица, тогда все свободные члены будут положительны [11].

$$A_1 x_1 + \cdots + A_n x_n + e_1 x_n + e_1 x_{n+1} + \cdots + e_m x_{n+m} = A_0 = \begin{matrix} a_{10} \\ \cdots \\ a_{m0} \end{matrix},$$

где $e_1 = \begin{matrix} 1 \\ 0 \\ \cdots \\ 0 \end{matrix}, \ldots, e_m = \begin{matrix} 0 \\ 0 \\ \cdots \\ 1 \end{matrix}$ является единичным базисом и $a_{i0} \geq 0$ для $i = 1,2,\ldots,m$.

К расширенной матрице ограничений $A_p = A_1, \ldots, A_n, e_1, \ldots, e_m, A_0$ применим шаги метода полного исключения.

Считаем что, a_{ij} – направляющий элемент преобразования. В процессе преобразований будут получены новые значения свободных членов [5].

$$a_{l0}^{(k+1)} = a_{l0}^{(k)} - \frac{a_{i0}^{k}\, a_{lj}^{k}}{a_{ij}^{k}}, \text{если } l \neq i, l = 1,2,\ldots,m, \tag{19}$$

$$a_{i0}^{(k+1)} = \frac{a_{i0}^{(k)}}{a_{ij}^{(k)}}.$$

Новое базисное решение считается допустимым, если будет исследовано выражение (19) и выяснены условия, при которых $a_{l0}^{(k+1)} > 0$ для всех l.

По предположению, $a_{l0}^{(k)} \geq 0, l = 1,2,\ldots,m$.

Предположим, что $a_{ij}^{(k)} > 0$, тогда $a_{i0}^{(k+1)} = \frac{a_{i0}^{(k)}}{a_{ij}^{(k)}} \geq 0$.

Если $a_{lj}^{k} < 0$, то, очевидно, $a_{l0}^{k+1} > 0$, так как $a_{i0}^{k} > 0$, $a_{ij}^{k} > 0$. Если $a_{lj}^{k} > 0$, то $a_{l0}^{k+1} = a_{lj}^{k} > 0$ при всех значениях $l = 1,2,\dots,m$ тогда и только тогда, когда

$$\frac{a_{i0}^{k}}{a_{ij}^{k}} = \min_{l}\{\frac{a_{l0}^{k}}{a_{lj}^{k}} \mid a_{lj}^{k} > 0\}. \tag{20}$$

Таблица 1. Симплекс-таблица

c	$-$	$-$	c_1	c_2	c_3	…	c_j	…	c_n
$-$	B_x	a_{i0}	A_1	A_2	A_3	…	A_j	…	A_n
c_1	x_1	a_{10}	a_{11}	a_{12}	a_{13}	…	a_{1j}	…	a_{1n}
c_2	x_2	a_{20}	a_{21}	a_{22}	a_{23}	…	a_{2j}	…	a_{2n}
…	…	…	…	…	…	…	…	…	…
c_i	x_i	a_{i0}	a_{i1}	a_{i2}	a_{i3}	…	a_{ij}	…	a_{in}
…	…	…	…	…	…	…	…	…	…
c_m	x_m	a_{m0}	a_{m1}	a_{m2}	a_{m3}	…	a_{mj}	…	a_{mn}
$-$	Δ	a_{00}	Δ_1	Δ_2	Δ_3	…	Δ_j	…	Δ_n

Когда направляющий элемент вычисляется по ниже приведенным правилам, тогда преобразование Гаусса становится симплекс-преобразованием:

a) Направляющий столбец j выбирается при наличии в нем хотя бы одного положительного элемента;

b) Направляющая строка i выбирается по минимальному отношению $\frac{a_{i0}}{a_{ij}}$ при условии $a_{ij} > 0$.

Такое преобразование требует ввести вектор A_j в базис и вывести вектор A_j. Для определения вектора вводимого в базис с целью увеличения целевой функции используются оценки векторов [10].

$$\Delta_j = \sum_{i \in I_\sigma} c_i x_{ij} - c_j = a_{oj},\ j \notin I_\sigma, \tag{21}$$

I_σ- множество индексов базисных векторов. Из условия:

$$\sum_{i \in I_\sigma} A_i x_{ij} = A_j. \tag{22}$$

определяется x_{ij}.

Оценки векторов равны симплекс-разностям для переменных $\{x_j\}$ с противоположным знаком [26]. Итак, для увеличения значения целевой функции, выбирается направляющий столбец A_j с наибольшей отрицательной оценкой, то есть

$$\Delta_j = a_{oj} = \min_k \frac{a_{0k}}{a_{0k}} < 0 \ .$$

Далее симплексным методом на каждой последующей итерацией необходимо заполнять таблицу 1.

Последняя строка таблицы считается индексной, она необходимо для определения направляющего столбца [32]. Ее элементы оценки будут определены формулой (21). Заметим, что для всех базисным векторов A_j , $i = 1,2,\dots,m$ **оценки** $\Delta_i = a_{0i} = 0$.

Целевую функцию a_{00} находят по следующему соотношению

$$a_{00} = \sum_{i=1}^{m} c_i x_i^{(k)}.$$

В столбец B_x записываются базисные переменные x_i , $i = 1,\dots,m$. Значение базисных переменных находят столбцом свободных членов a_{i0}, то есть

$$x_i = a_{i0}, \ i = 1,\dots,m.$$

Указанные стрелки в симплекс-таблице указывают на направляющий столбец A_j и строку A_i.

Переход от данной таблицы к следующей задается соотношениями (16), (17) и (18), если выбран направляющий элемент a_{ij}.

Таким образом, алгоритм решения задачи линейного программирования методом симплекс-таблиц включает следующие пунктов:

1. Начальная таблица заполняется допустимым единичным базисов вместе с индексной строкой [34].

2. Направляющий столбец A_j выбирается из условия:

$$a_{oj} = \min_{1 \le l \le n} \frac{a_{0l}}{a_{0l}} < 0 \ .$$

3. В качестве направляющей строки выбирают строку A_i для которой

$$\frac{a_{i0}}{a_{ij}} = \min_{1 \le r \le m} \{ \frac{a_{r0}}{a_{rj}} | \ a_{rj} > 0 \}.$$

4. При помощи соотношений 16 , 17 ,и 18 делают симплекс-преобразования с направляющим элементом a_{ij}.

Элементы индексной строки для новой таблицы вычисляются следующим образом:

$$a_{00}^{(k+1)} = a_{00}^{(k)} - \frac{a_{i0}^{k} \ a_{0j}^{k}}{a_{ij}^{k}}, \qquad a_{0l}^{k+1} = a_{0l}^{k} - \frac{a_{il}^{k} \ a_{0j}^{k}}{a_{ij}^{k}}, \qquad l = 1,2,\dots,n.$$

Правильность вычисления проверяется при помощи формул непосредственного счета [14].

$$a_{00}^{k+1} = \sum_{i \in I_{\sigma}^{(k+1)}} c_i a_{i0}^{(k+1)}; \tag{23}$$

$$a_{0l}^{k+1} = \sum_{i \in I_{\sigma}^{(k+1)}} c_i a_{il}^{(k+1)} - c_l. \tag{24}$$

В столбце B_x заменяют x_i на x_j, а в столбце c_i на c_j.

5. Если все $a_{0l}^{k+1} \ge 0, \ l = 1,2,\dots,n,$ тогда новое базисное решение $x_i = a_{i0}^{k+1}, i \in I_{\sigma}^{(k+1)}$ является оптимальным. Иначе происходит переход ко второму этапу, и далее производятся очередные итерации [35].

6. Последующие этапы повторяют, пока одна из итераций не закончится одним из двух исходов:

а) все $a_{i0} \ge 0, \ l = 1,2,\dots,n$ является условием оптимальности базиса окончательной таблицы;

b) всегда найдется такой $a_{0j} = \Delta_j < 0$, что все элементы данного столбца $a_{rj} \leq 0 \quad r = 1,2,\dots,m$. Это и есть признак неограниченности целевой функции $z = \sum_j c_j x_j$ множества допустимых решений [5].

Особенности применения симплекс-таблиц.

1. Если начальное базисное решение состоит из базиса свободных переменных для которых $c_i = 0$,тогда оценки всех небазисных переменных равны $\Delta_j = a_{0j} = -c_j$, и значения целевой функции будет записываться таким образом $a_{00} = \sum_{i \in I_\sigma} c_i x_i = 0$.

2. В качестве признака оптимальности базисного решения будет являться отсутствие векторов с отрицательными оценками, это применяется для решения задачи максимизации. Соответственно для задачи минимизации будет использовано аналогичное правило только оценки будут противоположными.

3. Целевая функция в области допустимых решений будет не ограниченной только тогда, когда будет хотя бы одна отрицательная оценка небазисного вектора, а соответственно столбец будет содержать только отрицательные элементы [15].

4. Когда решается задача минимизации, в базис вводятся вектора с наибольшей неотрицательной оценкой.

ГЛАВА 2. МЕТОД ДЕКОМПОЗИЦИИ ЗАДАЧ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

2.1 Теоретические основы метода декомпозиции

Общая задача линейного программирования с блочно-диагональной структурой части матрицы ограничений может быть записана в виде:

максимизировать

$$f = \sum_{j=1}^{J} \sum_{i \in I_j} c_{ij} x_{ij} \tag{2.1}$$

при условиях

$$\sum_{i \in 1_j} b_{ijk} x_{ij} \le P_{jk}; \tag{2.2}$$

$$x_{ij} > 0 \text{ при } i \in I_j; \quad x_{ij} = 0 \text{ при } i \notin I_j, \tag{2.3}$$

$$\sum_{j=1}^{J} \sum_{i \in I_j} b_{ijl} x_{ij} \le P_l, \quad l = 1, L. \tag{2.4}$$

Соотношения (2.2) – (2.3) – блочные, номер j соответствует фиксированному блоку, а (2.4) – связывающие ограничения [15].

Двойственную задачу для (2.1) – (2.4) запишем в таком виде:

минимизировать

$$\psi = \sum_{j=1}^{J} \sum_{k \in K_j} P_{jk} \xi_{jk} + \sum_{l=1}^{L} P_l \nu_l \tag{2.5}$$

при условиях

$$\sum_{k \in K_j} b_{ik} \xi_{jk} + \sum_{l=1}^{L} b_{il} \nu_l \ge c_{ij}, \quad i \in I_j, \quad j = 1, J; \tag{2.6}$$

$$\xi_{jk} \ge 0, \qquad j = 1, J, \qquad k \in K_j, \nu_l \ge 0, \qquad l = 1, L,$$

причем переменные ξ_{jk} соответствует (2.2), а ν_l – ограничениям (2.4).

Как и ранее, введем агрегированные переменные

$$X_i = \sum_{j=1}^{J} x_{ij}, \quad i = 1, I,$$

и веса агрегирования $\alpha_{ij} = \frac{x_{ij}}{X_i}$. Фиксируя веса α_{ij} и подставляя $x_{ij} = \alpha_{ij} X_i$ в (2.1) – (2.6), приходим к задаче в агрегированных переменных:

максимизировать

$$\sum_{i=1}^{I} C_i X_i \tag{2.7}$$

при условиях

$$\sum_{i=1}^{I} B_{ijk} X_i \le P_{jk}, \quad j = 1, J, \qquad k \in K_j; \tag{2.8}$$

$$\sum_{i=1}^{I} B_{il} X_i \le P_l, \quad l = 1, L; \tag{2.9}$$

$$X_i \ge 0, \qquad i = 1, I. \tag{2.10}$$

Введены такие обозначения:

$$B_{ijk} = b_{ijk}\alpha_{ij}, \qquad B_{il} = \sum_{j=1}^{J} b_{ijl}\alpha_{ij}; \quad C_i = \sum_{j=1}^{J} c_{ij}\alpha_{ij}. \tag{2.11}$$

Двойственная задача для (2.7) – (2.10) имеет следующий вид:

$$\min \psi = \sum_{j=1}^{J} \sum_{k \in K_j} P_{jk}\eta_{jk} + \sum_{l=1}^{L} P_l \delta_l \tag{2.12}$$

при условиях

$$\sum_{j=1}^{J} \sum_{k \in K_j} B_{ijk}\eta_{jk} + \sum_{l=1}^{L} B_{il}\delta_l \ge C_i, \quad i = 1, I; \tag{2.13}$$

$$\eta_{jk} \ge 0, \qquad j = 1, J; \quad k \in K_j; \quad \delta_l \ge 0, \quad l = 1, L, \tag{2.14}$$

где переменные η_{jk} соответствуют условию (2.8), а переменные δ_l – (2.9). Итак δ_l^0 это оптимальные двойственные оценки задачи (2.12) – (2.14).

Формируем задачи для отдельных блоков, это будет выглядеть таким образом:

минимизировать

$$h_j = \sum_{i \in I_j} (c_{ij} - \sum_{l=1}^{L} b_{ijl}\delta_l^0)x_{ij} \tag{2.15}$$

при условиях

$$\sum_{i \in I_j} b_{ijk}x_{ij} \le P_{jk}, \quad k \in K_j, \tag{2.16}$$

$$x_{ij} \ge 0 \ \text{при}\ i \in I_j; \quad x_{ij} = 0 \ \text{при}\ i \in I \backslash I_j. \tag{2.17}$$

Двойственные задачи для блочных задач имеют такой вид:

минимизировать

$$x_j = \sum_{k \in K_j} P_{jk}\xi_{jk} \tag{2.18}$$

при условиях

$$\sum_{k \in K_j} b_{ijk}\xi_{jk} \ge c_{ij} - \sum_{l=1}^{L} b_{ijl}\delta_l; \tag{2.19}$$

$$\xi_{jk} \ge 0, \qquad k \in K_j, \qquad j = 1,J, \tag{2.20}$$

где двойственные переменные ξ_{jk} соответствуют условию (2.16). Создание алгоритма решения задачи происходит на основе общей схемы алгоритма декомпозиции. Далее приводим задачу $2.1 - (2.4)$ к задачам меньших размерностей. В действительности, начальная задача имеет такой вид переменных $\sum_{j=1}^{J} J_j = N$. Итак, следующее это то, что задача в агрегированных переменных (2.7) состоит из I переменных, тогда как у блочных задач есть I_j переменные, и наконец, задача максимизации функции $\theta(\rho_j)$ включает в себя J переменных [16].

Каждый шаг процесса итераций требует неоднократного решения задачи в агрегированных переменных (2.7), непосредственно которая, имеет

не очень большое количество переменных $X_i, i = 1, I$, а также большое число ограничений [18].

2.2. Признак оптимальности промежуточного дезагрегированного решения

Итак, дадим формулировку признака оптимальности промежуточного дезагрегированного решения. Пусть будет получено оптимальное решение X_i^0

макрозадачи $2.7 - (2.10)$ при определенных весах α_{ij}, а x_{ij}^0 это дезагрегированное решение. Пускай δ_l^0 будет представлен как единственное оптимальное решение задачи $2.12 - (2.14)$, и x_{ij} это и есть оптимальное решение блочных задач $2.15 - 2.17$.

Теорема. Выполнение данного равенства является достаточным условием оптимальности решения x_{ij}^0 для задачи (1) – (4):

$$\sum_{j=1}^{J} \sum_{i \in I_j} \left(c_{ij} - \sum_{l=1}^{L} b_{ijl}\delta_l\right) x_{ij} + \sum_{l=1}^{L} P_l \delta_l^0 - \sum_{j=1}^{J} \sum_{i \in I_j} c_{ij} \cdot x_{ij}^0 = 0. \quad (2.21)$$

Если же задача (2.1) – (2.4) разрешима и x_{ij}^0 – неоптимально для нее, то в соотношении (2.21) знак равенства заменяется знаком «больше» [20].

Приведем доказательство к теореме 1. x_{ij}^0 это дезагрегированное решение являющиеся допустимым к исходной задаче $2.2 - (2.4)$. Это проверяют непосредственно подстановкой $x_{ij}^0 = \alpha_{ij} X_i^0$ в 2.2 и 2.4 при учете 2.8 , (2.9) и обозначений [19]. Целевая функция будет равна:

$$f^0 = \sum_{j=1}^{J} \sum_{i \in I_j} c_{ij} x_{ij}^0.$$

Набор ξ_{jk} , δ_l , где ξ_{jk} $j = 1, J, k \in K_j$ – оптимальные решения задач, двойственных к блочным задачам (18), это и есть допустимое решение к

задаче, двойственной для задачи (2.1) – (2.4). Все это появляется из соотношенйи (2.19), (2.20), (2.6).

Значение функционала φ

$$\varphi = \sum_{j=1}^{J} \sum_{k \in K_j} P_{jk} \xi_{jk} + \sum_{l=1}^{L} P_l \delta_l^0.$$

Равенство $\varphi = f^0$ дает приемлемость допустимого решения x_{ij}^0 задаче $2.2 - 2.4$. К тому же, когда x_{ij}^0 является неоптимальным для задачи (2.1) – (2.4) при том что она разрешима, $\varphi > f^0$.Резюмируя, получаем критерий оптимальность в виде

$$\sum_{j=1}^{J} \sum_{k \in K_j} P_{jk} \xi_{jk} + \sum_{l=1}^{L} P_l \delta_l^0 = \sum_{j=1}^{J} \sum_{i \in I_j} c_{ij} x_{ij}^0. \tag{2.22}$$

Однако, поскольку ξ_{jk}, $j = 1, J$, - оптимальные решения задач (2.18) – (2.20), тогда в применении к взаимосвязанным блочным задачам (2.15) – (2.17) и (2.18) – (2.20) имеем

$$\sum_{k \in K_j} P_{jk} \xi_{jk} = \sum_{i \in I_j} (c_{ij} - \sum_{l=1}^{L} b_{ijl} \delta_l^0) x_{ij}, \quad j = 1, J. \tag{2.23}$$

Подстановка (2.23) в (2.22) и доказывает равенство (2.21). Аналогично доказывается случай неоптимальных x_{ij}^0.

ГЛАВА 3 РЕАЛИЗАЦИЯ АЛГОРИТМА ДЕКОМПОЗИЦИИ

3.1. Разработка алгоритма декомпозиции

Дадим описание алгоритма декомпозиции на основе агрегирования для общей задачи линейного программирования. Пусть общая задача линейного программирования с блочно-диагональной частью матрицы ограничений задана в виде (2.1) – (2.4).

Предварительный этап.

1. Задаем начальные веса агрегирования $\alpha_{ij}(0)$, задачу решаем в агрегированных переменных (2.7) – (2.10) и находим X_i^0, $x_{ij}^0 = \alpha_{ij}\ 0\ X_i^0, i = 1, I, j = 1, J$.

2. Находим решение двойственной задачи (2.12) – (2.14) и оптимальные двойственные оценки δ_l^0 , $l = 1, L$.

3. Находим решение блочных задач (2.15) – (2.17) и определяем оптимальные решения x_{ij} .

4. Необходимо проверить условие (2.21). После проверки условия смотрим, когда оно строго равно нулю, конец и x_{ij}^0 – это оптимальные решения, если же условие имеет знак больше, то соответственно происходит переход к первой итерации [21].

Первая итерация. Пусть в результате k-й итерации получены $x_{ij}^0\ k$ и $x_{ij}\ k$.

1. Запишем следующее равенство

$$\alpha_{ij}\ k+1\ = \frac{x_{ij}^0\ k\ + \rho_j\ k+1\ [x_{ij}\ k\ - x_{ij}^0\ k\ ]}{\sum_{j=1}^{J} x_{ij}^0\ k\ + \rho_j\ k+1\ [x_{ij}\ k\ - x_{ij}^0\ k\ ]} = \alpha_{ij}\ \ \rho_j\ k+1\ \ .$$

2. Найдем максимум

$$\max_{p_j} \theta(\rho_j\ k+1\ ) = \max \sum_{i=1}^{I} \sum_{j=1}^{J} c_{ij}\alpha_{ij}\ \ \rho_j\ k+1 \quad X_i$$

и определим следующие значения $p_j^*\ k+1$ и $\alpha_j^0\ k+1\ = \alpha_j\ \ p_j^*\ k+1$.

К примеру, если принять $\rho_j = \rho$ для всех j, тогда можно использовать эффективный метод одномерного поиска [16].

3. Далее решим задачу с агрегированными переменными (2.7) – (2.10) с заданным $\alpha_{ij}^0\ k+1$ и находится $X_i^0\ k+1\ \ i = 1, I, \{x_{ij}^0\ k+1\ \}$.

4. Найдем решение двойственной задачи (2.12) – (2.14) к агрегированной, а также выявим оптимальные оценки $\delta_l^0\ \ l = 1, L$.

5. Следующим пунктом нужно решить блочные задачи $2.15 -$ (2.17) при найденных δ_l^0 , и еще находим $\{x_{ij}\}$.

Теперь происходит проверка признака оптимальности.

6. Итак, если при найденных значениях $\{x_{ij}\}$, δ_l^0 и $\{\{x_{ij}^0\}$ условие (2.21) будет выполняться строгим равенством, тогда конец $\{x_{ij}^0\}$ это оптимальные решения, если это условие не выполняется, переходим к $k+1$ – й итерации [17].

3.2. Реализация алгоритма декомпозиции

Максимизировать

$$x_{11} + 2x_{12} + 3x_{13} - 2x_{21} + 3x_{22} + x_{23} \tag{3.1}$$

при условиях

$$\begin{gathered} 5x_{11} + 2x_{21} \le 20; \\ -x_{11} + 3x_{21} \le 6; \\ x_{12} + 3x_{22} \ \le 12; \\ -x_{12} + 2x_{22} \le 5; \\ 3x_{13} + x_{23} \le 10 \\ 2x_{13} - x_{23} \le 4; \\ x_{11} + 2x_{12} + 3x_{13} + 3x_{21} - x_{22} - 4x_{23} \le 12; \\ -x_{11} + 4x_{12} + 2x_{13} + 2x_{21} + 3x_{22} + x_{23} \le 24; \end{gathered} \tag{3.2}$$

$$x_{11} \ge 0, x_{12} \ge 0, x_{13} \ge 0, x_{21} \ge 0, x_{22} \ge 0, x_{23} \ge 0.$$

Первая итерация. 1. Выбрать начальные веса агрегирования

$$\alpha_{11}\ 0\ = 0.4;\ \alpha_{12}\ 0\ = 0.4; \alpha_{13}\ 0\ = 0.2;$$

$$\alpha_{21}\ 0\ = \alpha_{22}\ 0\ = \alpha_{23}\ 0\ = 0.333.$$

2. Составить задачу в агрегированных переменных:

найти

$$\max 1.8X_1 + 0.66X_2 \tag{3.3}$$

при условиях

$$\begin{gathered} 2X_1 + 0.66X_2 \le 20;\ (\eta_1) \\ -0.4X_1 + X_2 \le 6;\ (\eta_2) \\ 0.4X_1 + X_2 \le 12;\ (\eta_3) \\ -0.4X_1 + 0.66X_2 \le 5;\ (\eta_4) \\ X_1 + 0.33X_2 \le 10;\ (\eta_5) \\ 0.66X_1 - 0.33X_2 \le 4;\ (\eta_6) \\ 1.8X_1 - 0.66X_2 \le 12;\ (\delta_1) \\ 1.6X_1 + 2X_2 \le 24;\ (\delta_2) \end{gathered} \tag{3.4}$$

3. Решаем ее графически и находим $X_1\ 1\ = 8.4; X_2\ 1\ = 5, \max f\ = 18.42.$

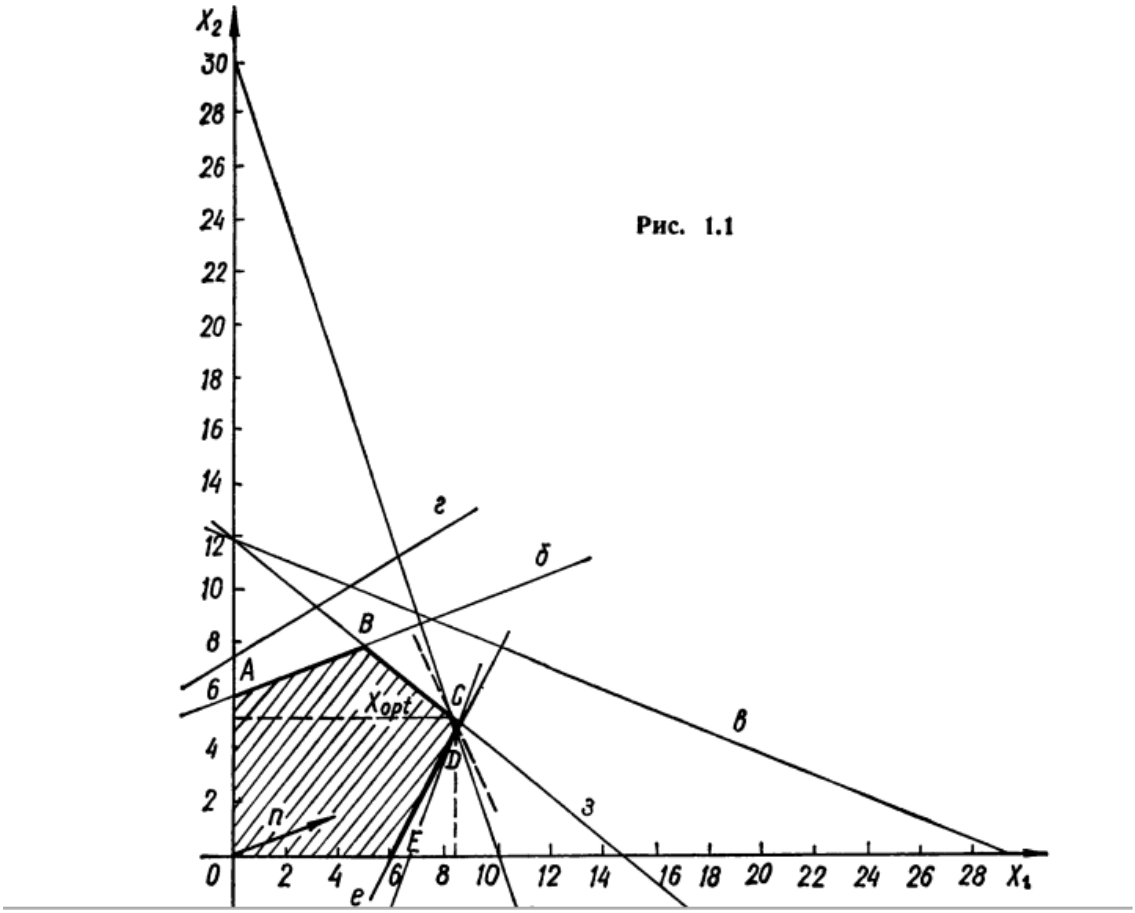

Рисунок 1 – Графический способ решения задачи.

Одновременно находим дезагрегированное решение:

$$x_{11}^{(0)} = \alpha_{11}\ 0\ \cdot X_1\ 1\ = 3.36; x_{12}^{0} = \alpha_{12}\ 0\ \cdot X_2\ 1\ = 3.36;$$

$$x_{13}^{(0)} = \alpha_{13}\ 0\ \cdot X_1\ 1\ = 1.68; x_{21}^{0} = x_{22}^{0} = x_{23}^{0} = 1.66.$$

4. Записать задачу, двойственную к агрегированной (2.3) – (2.4), найти

$$\min\ 20\eta_1 + 6\eta_2 + 12\eta_3 + 5\eta_4 + 10\eta_5 + 4\eta_6 + 12\delta_1 + 24\delta_2 = \min g(\eta, \delta) \qquad (3.5)$$

при условиях

$$2\eta_1 - 0.4\eta_2 + 0.4\eta_3 - 0.4\eta_4 + \eta_5 + 0.66\eta_6 + 1.8\delta_1 + 1.6\delta_2 \geq 1.8; \qquad (3.6)$$

$$0.66\eta_1 + \eta_2 + \eta_3 + 0.66\eta_4 + 0.33\eta_5 + 0.33\eta_6 - \delta_1 + 2\delta_2 \geq 0.66$$

Для нахождения двойственной задачи отметим, что ограничения а), з) строгие неравенства. Значит что, двойственные переменные η_1^0 и δ_2^0 в двойственной задачи не равны нулю.

Теперь, решим систему ограничений (3.6)

$$2\eta_1 + 1.6\delta_2 = 1.8$$
$$0.66\eta_1 + 2\delta_2 = 0.66\ ,$$

находим $\eta_1 = 0.871$ и $\delta_2 = 0.045$ и $\eta_2 = \eta_3 = \eta_4 = \eta_5 = \eta_6 = \delta_1 = 0.$

Где $\min g\ \eta, \delta\ = 18.42.$

5. Для отдельных блоков записываем следующие подзадачи.

Подзадача 1. Найти

$$\max h_1 = \max[c_{11} - (b_{111}\delta_1^0 + b_{112}\delta_2^0)]x_{11} + [c_{21} - (b_{121}\delta_1^0 + b_{122}\delta_2^0)]x_{21} = \ 1 + 1 \cdot 0.045\ x_{11} + \ -2 - 2 \cdot 0.045\ x_{21} = 1.045x_{11} - 2.09x_{21} \quad (3.7)$$

при условиях

$$5x_{11} + 2x_{21} \le 20;$$
$$-x_{11} + 3x_{21} \le 6; \quad (3.8)$$
$$x_{11} \ge 0, x_{21} \ge 0$$

Решив данную систему графически, найдем $x_{11} = 4;\ x_{21} = 0; h_1 = 4.180.$

Подзадача 2.

Найти

$$\max h_2 = (c_{12} - b_{122}\delta_2^0)x_{12} + (c_{22} - b_{222}\delta_2^0)x_{22} = \ 2 - 4 \cdot 0.045\ x_{12} + \ 3 - 3 \cdot 0.045\ x_{22} = 1.82x_{12} + 2.865x_{22}$$

при условиях

$$x_{12} + 3x_{22}\ \le 12;$$
$$-x_{12} + 2x_{22} \le 5;$$
$$x_{12} \ge 0, x_{22} \ge 0$$

Находим ее решение: $x_{12} = 12;\ x_{22} = 0; h_2 = 34.38.$

Подзадача 3. Найти

$$\max h_3 = (c_{13} - b_{132}\delta_2^0)x_{13} + (c_{23} - b_{232}\delta_2^0)x_{23} = 2.91x_{13} + 0.95x_{23}$$

при условиях

$$3x_{13} + x_{23} \le 10$$
$$2x_{13} - x_{23} \le 4;$$
$$x_{13} \ge 0, x_{23} \ge 0$$

Решая ее графически, найдем $x_{13} = 2.8;\ x_{23} = 1.6; h_2 = 9.668.$

6. Проверим условие оптимальности

$$\Delta h = h_1 + h_2 + h_3 + \sum_{l=1}^{2} b_{ijl}\delta_l^0 - \sum_{i=1}^{2}\sum_{j=1}^{3} c_{ij}x_{ij}^0 = 4.180 + 34.38 + 9.668 + 24.045 - 18.42 = 30.888 > 0$$

Так как знак неравенства больше нуля, то настоящее решение является не оптимальным, тем самым переходим ко второй итерации.

Вторая итерация. 1. Формируем новые значения весов $\alpha_{ij}(\rho)$ согласно формуле (2.21):

$$\alpha_{11}\ \rho = \frac{x_{11}^0 + \rho(x_{11} - x_{11}^0)}{\sum_{j=1}^{3} x_{1j}^0 + \rho(x_{1j} - x_{1j}^0)} = \frac{3.36 + \rho(4 - 3.36)}{3.36 + \rho\ 4 - 3.36\ + 3.36 + \rho\ 12 - 3.36\ + 1.68 + \rho(2.8 - 1.68)} = \frac{3.36 + 0.64\rho}{8.4 + 10.4\rho};\ \alpha_{12}\ \rho = \frac{3.36 + 8.64\rho}{8.4 + 10.4\rho},\ \alpha_{13}\ \rho = \frac{1.68 + 1.12\rho}{8.4 + 10.4\rho};\ \alpha_{21}\ \rho = \frac{1.66 - 1.66\rho}{5 - 3.38\rho};$$

$$\alpha_{22}\ \rho = \frac{1.66 - 1.66\rho}{5 - 3.38\rho};\ \alpha_{23}\ \rho = \frac{1.66 - 1.66\rho}{5 - 3.38\rho}.$$

2. Находим выражение для целевой функции $\theta\ \rho$:

$$\theta\ \rho = \sum_{i=1}^{2}\sum_{j=1}^{3} c_{ij}x_{ij}^0\ \rho = X_1 \sum_{j=1}^{3} c_{1j}\alpha_{2j}\ \rho + X_2 \sum_{j=1}^{3} c_{2j}\alpha_{2j}\ \rho = \frac{8.4(15.12 + 21.28\rho}{8.4 + 10.4\rho} + \frac{5(3.34 - 1.74\rho)}{5 - 3.38\rho}.$$

3. Находим p^*, при котором $\theta\ \rho^* = \max_{0 \le \rho \le 1} \theta\ \rho$. Для этого вычисляем и заполняем таблицу значение (табл. 1):

ρ	0	0.5	0.8	0.9	1
$\theta\ \rho$	18.46	19.64	20.382	20.736	21.194

Ясно, что оптимальное значение $p^* = 1.$

4. Вычисляем новые значения весовых коэффициентов:

$$\alpha_{11}^0\ p^* = 1\ = \frac{3{,}36 + 0{,}64}{8{,}4 + 10{,}4} = \frac{4}{18{,}8} = 0{,}213;$$

$$\alpha_{12}^0\ p^* = 1\ = \frac{12}{18{,}8} = 0{,}638; \alpha_{13}^0 = 0.149;\ \alpha_{21}^0 = \alpha_{22}^0 = 0;\ \alpha_{23}^0 = 1.$$

5. Записываем задачу в агрегированных переменных при данных значениях весов:

максимизировать

$$f\ X_1, X_2\ = 1.936X_1 + X_2 \tag{3.9}$$

при условиях

$$\begin{gathered} \text{а})1.065X_1 \leq 20; \\ \text{б}) - 0.213X_1 \leq 6; \\ \text{в})0.638X_1 \leq 12; \\ \text{г}) - 0.638X_1 \leq 5; \\ \text{д})0.447X_1 + X_2 \leq 10; \\ \text{е})0.294X_1 - X_2 \leq 4; \\ \text{ж})1.936X_1 - 4X_2 \leq 12; \\ \text{з})2.637X_1 + X_2 \leq 24; \end{gathered} \tag{3.10}$$

Эта задача легко решается графически, и ее решение (рис.2):

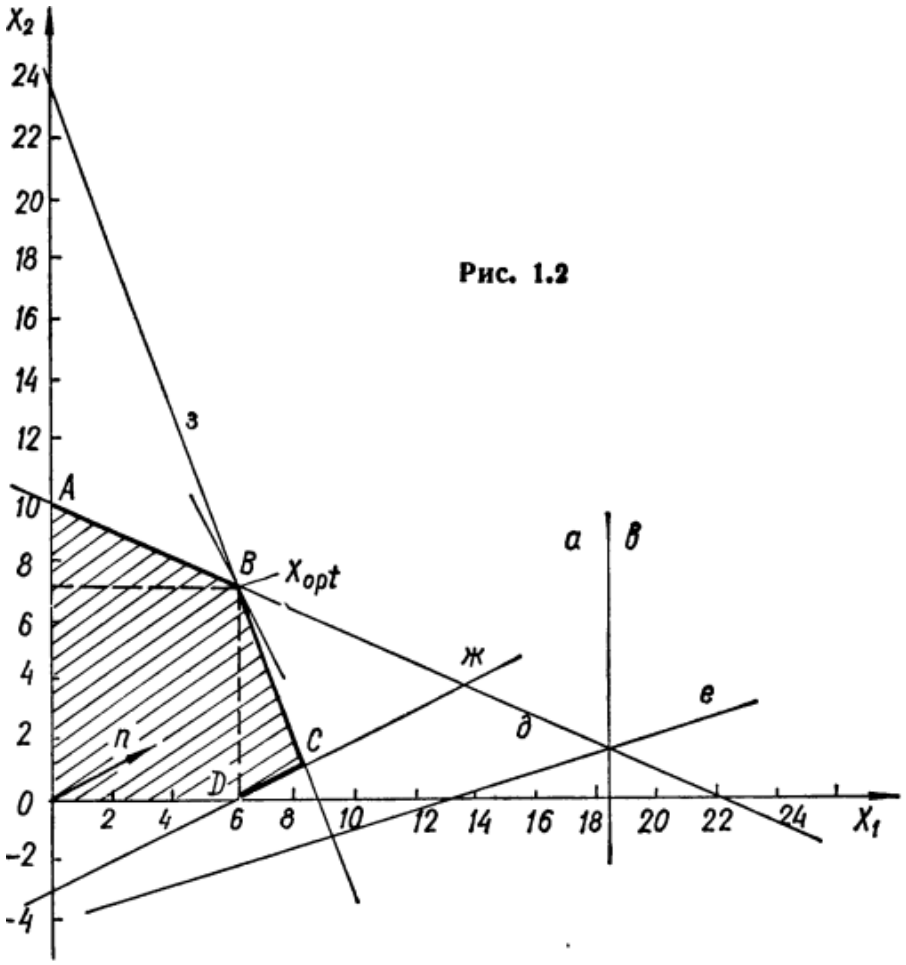

Рис. 1.2

Рисунок 2 – Решение задачи графическим способом (Итерация 2)

$$X_1 = 6.39;\ X_2 = 7.144; f\ X_1, X_2\ = 19.515.$$

Тогда дезагрегированное решение равно:

$x_{11}^0\ 2\ = \alpha_{11}\ 2\ * X_1 = 1.36;\ x_{12}^0\ 2\ = 4.08; x_{13}^0\ 2\ = 0.952; x_{21}^0\ 2\ =$

$x_{22}^0\ 2\ = 0;\ x_{23}^0 = 7.144$.

6. Записываем двойственную задачу к агрегированной (2.9) – (2.10).

Минимизировать

$$g\ \eta,\delta\ = 20\eta_1 + 6\eta_2 + 12\eta_3 + 5\eta_4 + 10\eta_5 + 4\eta_6 + 12\delta_1 + 24\delta_2 \qquad (3.11)$$

при условиях

$$1.065\eta_1 - 0.213\eta_2 + 0.638\eta_3 - 0.638\eta_4 + 0.447\eta_5 + 0.246\eta_6 \qquad (3.12)$$
$$+ 1.936\delta_1 + 2.637\delta_2 \geq 1.936;$$
$$\eta_5 - \eta_6 - 4\delta_1 + \delta_2 \geq 1$$

Для ее решения воспользуемся тем, что оптимальное решение прямой задачи достигаются в точке В, где ограничения 10, д) и 10,з) выполняются как строгие равенства (рис.2). Поэтому, согласно одной из теорем двойственности, соответствующие им двойственные переменные $\eta_5 > 0$ и $\delta_2 > 0$, а все остальные переменные равны 0.

Тогда решая систему ограничений (3.12), при этом условии получим

$$0.447\eta_5 + 2.637\delta_2 = 1{,}936;$$
$$\eta_5 + \delta_2 = 1.$$

Находим $\eta_5 = 0{,}32;\ \delta_2 = 0.68$ и, очевидно, $\delta_1 = 0$. Можно проверить, что при этом решении $g\ \eta^0,\delta^0\ = 1.95 = \max f\ X_1, X_2$.

7. Формируем и решаем блочные задачи.

Подзадача 1.

Максимизировать

$\max h_2 = (c_{11} - b_{112}\delta_2^0)x_{11} + (c_{21} - b_{212}\delta_2^0)x_{21} =\ 1 + 0{,}687\ x_{11} +$

$-2 - 2 * 0{,}68\ x_{21} = 1.68x_{11} - 3{,}36x_{21}$.

при условиях

$$5x_{11} + 2x_{21} \leq 20;$$
$$-x_{11} + 3x_{21} \leq 6;$$
$$x_{11} \geq 0, x_{21} \geq 0$$

Ее решение таково: $x_{11} = 4;\ x_{21} = 0; h_1 = 6.72.$

Подзадача 2.

Максимизировать

$$\max h_2 = (c_{12} - b_{122}\delta_2^0)x_{12} + (c_{22} - b_{222}\delta_2^0)x_{22} = -0.72x_{12} + 0.96x_{22}$$

при условиях

$$x_{12} + 3x_{22} \leq 12;$$
$$-x_{12} + 2x_{22} \leq 5;$$
$$x_{12} \geq 0, x_{22} \geq 0$$

Ее решение таково: $x_{12} = 0;\ x_{22} = 2{,}5; h_2 = 2{,}175.$

Подзадача 3.

Максимизировать

$$\max h_3 = (c_{13} - b_{132}\delta_2^0)x_{13} + (c_{23} - b_{232}\delta_2^0)x_{23} = 1.536x_{13} + 0.268x_{23}$$

при условиях

$$3x_{13} + x_{23} \leq 10$$
$$2x_{13} - x_{23} \leq 4;$$
$$x_{13} \geq 0, x_{23} \geq 0$$

Легко находим графически ее решение: $x_{13} = 2.8;\ x_{23} = 1.6, \text{при этом } h_2 = 4.73.$

8. Проверяем условие оптимальности

$$\Delta h = h_1 + h_2 + h_3 + \sum_{l=1}^{2} P_l\delta_l - \sum_{i=1}^{2}\sum_{j=1}^{3} c_{ij}x_{ij}^0 = 6.928 + 2.175 + 4.73 + 24 * 0.732 - 19.52 = 12.465 > 0$$

Так как $\Delta h\ 2\ > 0$, то текущее решение неоптимально и переходим к третьей итерации.

Третья итерация. 1. Формируем новые значения весов согласно (2.21):

$$\alpha_{11}\ \rho\ = \frac{1.36+2.64\rho}{6.39+0.408\rho};\ \alpha_{12}\ \rho\ = \frac{4.08-4.08\rho}{6.39+4.08\rho},\ \alpha_{13}\ \rho\ = \frac{0.952+1.848\rho}{6.39+0.408\rho};\ \alpha_{21}\ \rho\ = 0;$$

$$\alpha_{22}\ \rho\ = \frac{2.5\rho}{7.144-3.044\rho};\ \alpha_{23}\ \rho\ = \frac{7.144-5.544\rho}{7.144-3.044\rho}.$$

2. Находим выражение для целевой функции $\theta\ \rho$:

$$\theta\ \rho\ = \frac{6.39(12.376+0.024\rho)}{6.39+0.408\rho} + \frac{7.144(7.144+1.956\rho)}{7.144-3.044\rho}.$$

Найдем p^*, при котором $\theta\ \rho^*\ = \max_{0\le\rho\le 1}\theta\ \rho$. Для этого составляем таблицу 2.

ρ	0	0.1	0.4	0.8	1
$\theta\ \rho$	19.52	20.50	21.55	25	27.51

Итак, оптимальное значение ρ равно $\rho^* = 1$.

3. Вычисляем новые значения весов при $\rho^* = 1$.

$$\alpha_{11}^0 = 0{,}588;$$

$$\alpha_{12}^0 = 0; \alpha_{13}^0 = 0.412;\ \alpha_{21}^0 = 0;\ \alpha_{22}^0 = 0.61;\ \alpha_{23}^0 = 0.39.$$

4. Составляем задачу в агрегированных переменных при данных значениях весов:

максимизировать

$$f\ X_1, X_2\ = 1.824X_1 + 2.2X_2 \qquad (3.15)$$

при условиях

$$\begin{gathered} 2.94X_1 \le 20; \\ -0.588X_1 \le 6; \\ 1.83X_2 \le 12; \\ 1.22X_2 \le 5; \\ 1.236X_1 + 0.39X_2 \le 10; \\ 0.824X_1 - 0.39X_2 \le 4; \\ 1.824X_1 - 2.17X_2 \le 12; \\ 0.236X_1 + 2.22X_2 \le 24; \end{gathered} \qquad (3.16)$$

Решая ее графически, находим

$$X_1 = 6.8;\ X_2 = 4.1; f\ X_1, X_2\ = 21.42.$$

5. Записываем двойственную задачу к (3.15) – (3.16).

Минимизировать

$$g\ \eta, \delta\ = 20\eta_1 + 6\eta_2 + 12\eta_3 + 5\eta_4 + 10\eta_5 + 4\eta_6 + 12\delta_1 + 24\delta_2 \qquad (3.17)$$

при условиях

$$2.94\eta_1 - 0.588\eta_2 + 1.236\eta_5 + 0.824\eta_6 + 1.824\delta_1 + 0.236\delta_2 \ge 1.824; \qquad (3.18)$$

$$1.83\eta_3 + 1.22\eta_4 + 0.39\eta_5 - 0.39\eta_6 - 2.17\delta_1 + 2.2\delta_2 \ge 2.2$$

Для ее решения воспользуемся тем же приемом, что и ранее. Поскольку ограничения а) и г) в условиях (3.16) выполняются как строгие равенства. Тогда разрешив условие (3.18) относительно переменных η_1, η_4, получим

$\eta_1 = 0{,}620, \eta_4 = 1.803$ и $\delta_1 = \delta_2 = 0$. Легко проверить, что при этом решении $\min g\ \eta^0, \delta^0\ = 20\eta_1 + 5\eta_4 = \max f\ X_1, X_2\ .$

6. Записываем задачи для отдельных блоков.

а) Подзадача 1.

Максимизировать

$$h_1 = x_{11} - 2x_{21}\ .$$

при условиях

$$5x_{11} + 2x_{21} \leq 20;$$
$$-x_{11} + 3x_{21} \leq 6;$$
$$x_{11} \geq 0, x_{21} \geq 0$$

Откуда $x_{11} = 4;\ x_{21} = 0; h_1 = 4.$

б) Подзадача 2.

Максимизировать

$$h_2 = 2x_{12} + 3x_{22}$$

при условиях:

$$x_{12} + 3x_{22}\ \leq 12;$$
$$-x_{12} + 2x_{22} \leq 5;$$
$$x_{12} \geq 0, x_{22} \geq 0$$

Ее решение $x_{12} = 0;\ x_{22} = 2{,}5; h_2 = 7.5.$

в) Подзадача 3

максимизировать

$$h_3 = 3x_{13} + x_{23}$$

при условиях

$$3x_{13} + x_{23} \leq 10$$
$$2x_{13} - x_{23} \leq 4;$$
$$x_{13} \geq 0, x_{23} \geq 0$$

Ее решение: $x_{13} = 2.8;\ x_{23} = 1.6, \text{откуда}\ h_2 = 10.$

7. Проверяем признак оптимальности:

$$\Delta\ h = 0$$

Так как $\Delta h\ 3\ = 0$, то текущее дезагрегированное решение $\{x_{ij}^{0}\ 3\ \}$ – оптимально. Оно равно

$x_{11}^{0}\ 3\ = \alpha_{11}\ 3\ * X_1 = 4;\ x_{12}^{0}\ 3\ = 0; x_{13}^{0}\ 3\ = 2.8; x_{21}^{0}\ 3\ = 0; x_{22}^{0}\ 3\ = 2.5;\ x_{23}^{0} = 1.6.$

Максимум функции этого решения $\max f\ \ x_{ij}\ \ = 21.5.$

В качестве среды разработки был выбран MATLAB. MATLAB – это язык высокого уровня и интерактивная среда для программирования, численные вычисления и визуализация результатов. Благодаря MATLAB анализируются данные, разрабатываются алгоритмы, создаются модели, приложения [13].

MATLAB одна из широко используемых математических вычислительных сред. В 1970-х годах MATLAB начинался как интерактивный интерфейс для EISPACK и LINPACK. С того момента среда значительно расширилась в ней появились инструменты для решения задач в различных прикладных областях. MATLAB подходит для программирования матричных задач, так как он дает гибкую индексацию матриц. MATLAB имеет удобному интерфейс, хорошие графические возможности, имеющие библиотеки ускоряющие векторные матричные вычисления. Кроме того, исследователи строят очень сложные системы с использованием языка и инструментов MATLAB [6].

MATLAB предоставляет большое количество методов для анализа данных, разработки алгоритмов и создания моделей. MATLAB включает также математические функции для инженерных и научных операций.

Ключевыми особенностями MATLAB является следующее:

- Это высокоуровневый язык программирования, который работает с матричными вычислениями и с разработкой алгоритмов;
- Диалоговая среда разработки кода, управления файлами и данными.

- Функции линейной алгебры, статистики, анализ Фурье, решение дифференциальных уравнений и другие.
- Присутствуют обширные средства визуализации, а также 2-D и 3-D графика.
- Встроенные средства разработки пользовательского интерфейса для создания законченных приложений на MATLAB

MATLAB это инструмент, который обеспечивает взаимодействие оператора со всеми доступными возможностями анализа, сбора данных и презентаций. У него также есть свои минусы и плюсы, как практически и во всех языках программирования.

Недостатки:

- Медленный и перегруженный операторами, командами, функциями язык, основной целью которого является улучшение визуального восприятия.
- Узконаправленный. Нет никакой больше программной платформы, где бы MATLAB был полезен.
- Невысокий спрос. Несмотря на большой интерес к MATLAB практически во всех сферах, фактически и легально его используют лишь немногие.

Достоинства:

- Язык легок для изучения, обладает простым и понятным синтаксисом.
- Огромные возможности. Но это скорее преимущество всего продукта в целом.
- Частые обновления, как правило заметные положительные преобразования происходят не реже пары раз в год.
- Программная среда позволяет преобразовывать его в “быстрый” код на C, C++.

Хоть и используется алгоритм декомпозиции, в основе его реализации будет лежать симплекс-метод. В данном случае он будет применен в нахождении решения блочных задач. В численном примере блоки решались

графическим путем, но так, как это является не очень удобным в реализации, был выбран симплекс-метод [12].

В MATLAB уже существует toolbox для нахождения симплекс-метода, он и будет взят в качестве основы реализации. Toolbox это набор инструментов, либо это специализированные группы программ, во многих источниках это определяется по-разному. В любом случае toolbox это скорее функция, которая упрощает в разы вычисления, конечно же для написания toolbox уходит не малое количество строк кода, но в итоге он может применяться в других файлах в виде одной строки с соответствующими параметрами.

Итак, как уже было выше сказано, в основе реализации метода декомпозиции лежит симплекс-метод. В программном коде будем использовать функцию $linprog$, она состоит из следующих параметров (f, A, b), где f – вектор коэффициентов целевой функции, A – матрица ограничений-неравенств, а b – вектор правых частей ограничений неравенств.

Также при реализации алгоритма, будем использовать параллельное программирование. Для чего его в этом случае используют, чтобы не выполнять решения блоков последовательно, так как они не зависят друг от друга, их можно посчитать параллельно, и по сравнению с последовательным выполнением, параллельное будет выполняться в разы быстрее.

Ниже представлен кусок параллельной программы:

$B = \ 1\,1\,1; -1,1,1; 1, -1,1; -1, -1,1\ ;$

$A = zeros\ \ 4 * N, 3 * N\ ;$

$parfor\ i = 1{:}N$

$A\ \ 4 * i - 3{:}\,4 * i, 3 * i - 2{:}\,3 * i\ \ = B; end$

$b = zeros\ \ 4N, 1\ ;$

В данном случае сначала задаются векторные строки, после переходим к выполнению цикла $parfor$. Цикл $parfor$ используется в параллельном

программировании для того, чтобы выполнять в одно и то же время параллельные вычисления.

После того, как весь цикл будет выполнен, происходит проверка условия оптимальности при помощи следующей формулы:

$$\Delta h = h_1 + \cdots + h_n + \sum_{l=1}^{2} P_l \delta_l - \sum_{i=1}^{2} \sum_{j=1}^{3} c_{ij} x_{ij}^0$$

Далее, если условие оптимальности меньше нуля, мы завершаем процесс итераций, в противном случае следует выполнить еще один цикл. Таким образом цикл будет выполняться до тех пор, пока условие оптимальности не будет выполнено.

Для чего собственно было использовано параллельное программирование, его практическая цель это повышение эффективности компьютерного применения. Многие компьютерные приложения по-прежнему превосходят возможности, обеспечиваемые самыми быстрыми одиночными процессорами, поэтому они вынуждены использовать нескольких процессоров. Даже приложения с высокой однопроцессорной производительностью часто уступают параллельным системам, которые используют намного меньше затрат.

Основным достоинством параллельного программирования является эффективное выполнение кода, так как параллельное программирование экономит время, позволяя выполнять приложения в более короткие сроки. Как следствие эффективного выполнения кода, параллельное программирование часто соотносится с размером проблемы и, таким образом, может решить масштабные проблемы. Как правило, параллельное программирование является средством обеспечения параллелизма, в частности одновременного выполнения нескольких действий.

ЗАКЛЮЧЕНИЕ

Целью данной дипломной работы являлось разработать и реализовать алгоритм декомпозиции. Поставленные задачи были выполнены:

1. Проанализировать теоретические основы симплекс-метода
2. Исследовать сущность декомпозиции для задачи линейного программирования.
3. Реализовать декомпозицию для задачи линейного программирования.

В ходе изучения теоретических основ симплекс-метода было выявлено, что он малоэффективен в задачах большой размерности, так как увеличивается число переменных, и тем самым возникают сложности при минимизации (максимизации) целевой функции.

По этой причине была изучена сущность декомпозиции для задач линейного программирования, и как оказалось в сравнении с симплекс-методом, он неограничен в количестве переменных и разбиваясь на подзадачи упрощает ход решения.

Хоть в качестве алгоритма для решения задач больших размерностей выбран алгоритм декомпозиции, это не исключает факт непосредственного использования симплекс-метода. В численном примере было видно, что разбив задачу на блоки, каждый блок решается графическим способом, но для того чтобы еще более упростить нахождения значений, был использован симплекс-метод.

Таким образом, в завершении данной выпускной квалификационной работы был реализован алгоритм декомпозиции. Который как было выяснено в ходе работы высокоэффективен и очень удобен при решении задач больших размерностей. Одним из главных достоинств этого метода является то, что не требуются громоздкие вычисления, и не возникает трудностей решить задачу, так как благодаря разбиению задач на блоки, каждую блочную задачу очень легко решить.

Благодаря этому методу сокращается количество вычислений, а также существует выигрыш во времени, что еще более убеждает в том, что метод на самом деле является эффективным.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

Научная и методическая литература

1. Вагнер Г. Основы исследования операций: В 3 т. – М. : Мир, 1972. – Т. 2 – 3.

2. Вентцель Е.С. Исследование операций. – М. : Сов.радио,1972.

3. Гладких Б.А. Методы оптимизации и исследование операций для бакалавров информатики. Ч. 1. Введение в исследование операций. Линейное программирование: Учебное пособие. – Томск: Изд-во НТЛ, 2009. – 200 с.

4. Гольштейн Е.Г. Новые направления в линейном программировании. – Учебное издание. – Москва: Изд-во Советское радио, 1966. – 75 с.

5. Дж. Данциг Линейное программирование, его применения и обобщения: Учебное пособие. – Москва: Изд-во Прогресс,1966. – 256 с.

6. Ермольев Ю.М., Ляшко И.И., Михалевич В.С., Тюптя В.И. Математические методы исследования операций. – К.: Вища шк. Головное изд-во, 1979.

7. Есипов Б.А. Методы оптимизации и исследование операций: учеб.пособие/Б.А. Есипов. – Самара: Изд-во Самар.гос.аэрокосм. ун-та, 2007.

8. Зайченко Ю.П. Исследование операций: Нечеткая оптимизация: Учеб. пособие. – К.: Выща шк., 1991.

9. Лэсдон Л. Оптимизация больших систем. – М.: Наука, 1975.

10. Зайченко Ю.П. Исследование операций: Учеб. пособие для студентов вузов. – 2-е изд., перераб. и доп. – Киев: Вища школа. Головное изд-во, 1979. 392 с.

11. Цирков В.И. Декомпозиционные методы решения задач большой размерности. – М. : Наука, 1981.

Электронные ресурсы

12. MATLAB – Википедия [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/MATLAB

13. MATLAB – высокоуровневый язык технических расчетов [Электронный ресурс]. – Режим доступа: https://matlab.ru/products/matlab

14. Горелик В.А. Теория принятия решений [Электронный ресурс]: учебное пособие для магистрантов/ Горелик В.А.— Электрон. текстовые данные.— М.: Московский педагогический государственный университет, 2016.— 152 с.— Режим доступа: http://www.iprbookshop.ru/72518.html.

15. Горлач, Б.А. Исследование операций [Электронный ресурс]: учеб. пособие — Электрон. дан. — Санкт-Петербург: Лань, 2013. — 448 с. — Режим доступа: https://e.lanbook.com/book/4865.

16. Заботин, И.Я. Методы и вычислительные приемы в линейном программировании [Электронный ресурс] : учеб. пособие / И.Я. Заботин, Я.И. Заботин. — Электрон. дан. — Казань : КФУ, 2014. — 116 с. — Режим доступа: https://e.lanbook.com/book/72810.

17. Кутузов, А.Л. Исследование операций: учеб. пособие [Электронный ресурс] : учеб. пособие — Электрон. дан. — Санкт-Петербург : СПбГПУ, 2011. — 98 с. — Режим доступа: https://e.lanbook.com/book/64797.

18. Лабораторная работа №1 [Электронный ресурс]. – Режим доступа: https://studfiles.net/preview/5686766/page:2/

19. Лунгу, К.Н. Линейное программирование. Руководство к решению задач [Электронный ресурс] : учеб. пособие — Электрон. дан. — Москва : Физматлит, 2009. — 132 с. — Режим доступа: https://e.lanbook.com/book/2253.

20. Метод декомпозиции Данцига-Вулфа [Электронный ресурс]. – Режим доступа: http://mylektsii.ru/6-173872.html .

21. Метод декомпозиции Данцига-Вульфа [Электронный ресурс]. – Режим доступа: http://iasa.org.ua/lections/iso/10/10.1.htm/.

22. Методы оптимизации в задачах большой размерности [Электронный ресурс]. – Режим доступа: https://studfiles.net/preview/997091/.

23. Минько, А.Э. Методы прогнозирования и исследования операций. Учебное пособие [Электронный ресурс] : учеб. пособие / А.Э.

Минько, Э.В. Минько. — Электрон. дан. — Москва : Финансы и статистика, 2010. — 480 с. — Режим доступа: https://e.lanbook.com/book/28357.

24. Основы линейного программирования [Электронный ресурс] : учеб. пособие / В.В. Чистов [и др.]. — Электрон. дан. — Москва : МГТУ им. Н.Э. Баумана, 2017. — 64 с. — Режим доступа: https://e.lanbook.com/book/103535.

25. Разложение Данцига-Вульфа [Электронный ресурс]. – Режим доступа: https://www.wikiplanet.click/enciclopedia/RU/Разложение_Данцига-Вулфа.

26. Ржевский, С.В. Исследование операций [Электронный ресурс]: учеб. пособие — Электрон. дан. — Санкт-Петербург: Лань, 2013. — 480 с. — Режим доступа: https://e.lanbook.com/book/32821.

27. Сеславин, А.И. Исследование операций и методы оптимизации [Электронный ресурс] : учеб. пособие / А.И. Сеславин, Е.А. Сеславина. — Электрон. дан. — Москва : УМЦ ЖДТ, 2015. — 200 с. — Режим доступа: https://e.lanbook.com/book/80027.

28. Стронгин, Р.Г. Исследование операций и модели экономического поведения [Электронный ресурс] : учеб. пособие — Электрон. дан. — Москва : , 2016. — 245 с. — Режим доступа: https://e.lanbook.com/book/100769.

29. Чепурницкий, В.С. Исследование операций на основе стандартных программ [Электронный ресурс] : учеб. пособие / В.С. Чепурницкий, А.В. Чесноков. — Электрон. дан. — Москва : Горная книга, 2002. — 121 с. — Режим доступа: https://e.lanbook.com/book/3542.

30. Черников, Ю.Г. Системный анализ и исследование операций [Электронный ресурс] : учеб. пособие — Электрон. дан. — Москва : Горная книга, 2006. — 370 с. — Режим доступа: https://e.lanbook.com/book/3512.

31. Шапкин, А.С. Математические методы и модели исследования операций [Электронный ресурс] : учеб. / А.С. Шапкин, В.А. Шапкин. —

Электрон. дан. — Москва : Дашков и К, 2016. — 400 с. — Режим доступа: https://e.lanbook.com/book/72413.

Литература на иностранном языке

32. A review of the decomposition method in applied mathematics/G.Adomain//Center for applied mathematics, University of Georgia. – 1988. – Режим доступа: https://ac.els-cdn.com/0022247X88901709/1-s2.0-0022247X88901709-main.pdf?_tid=2d1b9517-a5a6-4875-ac3a-4e7d4f42eb99&acdnat=1527571061_e5d623c921e5a2c0a1fce6517c5f6398

33. A toolbox for modeling and optimization in MATLAB/Johan Lofber // Automatic Control Laboratory, ETHZ CH-8092 Zurich, Switzerland. – 2004. – [Электронный ресурс] – Режим доступа: https://s3.amazonaws.com/academia.edu.documents/50931534/YALMIP__a_toolbox_for_modeling_and_optim20161217-7516-1ko3t0w.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1528349050&Signature=cisv95Fb5x5K%2BJ46Y.pdf

34. Automation and combination of linear-programming based stabilization techniques in column generation / A. Pessoa, R. Sadykov, E. Uchoa, and F. Vanderbeck // Universidade Federal Fluminense, LOGIS lab INRIA Bordeaux-Sud-Ouest, team RealOpt University of Bordeaux, Mathematics Institute. – 2017. – Режим доступа:https://hal.inria.fr/hal-01077984/docum

35. The Semantics of a Simple Language for Parallel Programming /Gilles KAHN//IRIA-Laboria, Domaine de Voluceau, 78150 Rocquencourt, France and Commissariat à l'Energie Atomique, France. – 1977 – [Электронный ресурс] – Режим доступа: https://pdfs.semanticscholar.org/d42a/29e6977c28f7bf23d63b00c48f2e9100403e.pdf