**© 2019 г. С.Н. МЕДВЕДЕВ, канд. физ.-мат. наук (s_n_medvedev@mail.ru),
О.А. МЕДВЕДЕВА, канд. физ.-мат. наук (medvedeva@amm.vsu.ru)
(Воронежский государственный университет)**

АДАПТИВНЫЙ АЛГОРИТМ РЕШЕНИЯ АКСИАЛЬНОЙ ТРЕХИНДЕКСНОЙ ЗАДАЧИ О НАЗНАЧЕНИЯХ

Предлагается вероятностная модификация алгоритма поиска минимального элемента для решения аксиальной трехиндексной задачи о назначениях. Общая идея связана с расширением базовых алгоритмических схем "жадного" типа посредством перехода к вероятностной постановке на основе рандомизации переменных. Задача минимизации целевой функции заменяется задачей минимизации ее математического ожидания. Для построения алгоритма реализованы три этапа. На первом задается движение во множестве случайных величин. На втором решается неравенство — условие локального улучшения. На третьем происходит пересчет вероятностей, т.е. происходит процесс "адаптации". Второй этап выявляет одну из особенностей алгоритма: на формирование решения влияют не только "качества" самого элемента, но и возможные потери при его выборе.

Ключевые слова: аксиальная трехиндексная задача о назначениях, дискретная оптимизация, адаптивный алгоритм решения, вероятностная постановка задачи, условие локального улучшения.

DOI: 10.1134/S0005231019040093

1. Введение

Аксиальная задача о назначениях во многих отношениях напоминает классическую задачу о назначениях (ЗОН), однако в отличие от нее является NP-трудной [1].

Для решения аксиальной трехиндексной ЗОН Гансом и Кауфманом [2] был предложен прямо-двойственный алгоритм, который является аналогом венгерского метода решения классической ЗОН. Ставится задача отыскания максимального паросочетания гиперграфа, которая является NP-трудной. Аналогично классическому методу вместо данной задачи можно рассмотреть проблему отыскания минимального вершинного покрытия гиперграфа. Однако для аксиальной трехиндексной ЗОН теорема Кенига не выполняется: минимальное число вершин в минимальном покрытии может быть строго больше, чем мощность любого (максимального) паросочетания. Дальнейшая работа алгоритма [2], основанного на методе ветвей и границ, сводится к поиску соответствующих паросочетания и вершинного покрытия мощности n.

Как было сказано выше, получение оптимального результата аксиальной трехиндексной ЗОН есть задача NP-трудная, поэтому было разработано большое количество приближенных методов решения. Первый эвристический алгоритм был предложен в 1967 г. [3]. В дальнейших исследованиях предла-

гались различные варианты изменений точных методов решения задач дискретной оптимизации, примененных к аксиальной ЗОН. В 2005 г. был разработан "жадный" алгоритм с перекомпановкой [4], который показал хорошие результаты по времени работы на основе компьютерных вычислений. Хуанг и Лим [5] предложили гибридный генетический алгоритм решения аксиальной ЗОН. В [6] представлен метод, названный авторами "редукционным". В разное время было разработано несколько эвристических алгоритмов для задач с декомпозицией целевых коэффициентов. В [7] исследована задача, в которой целевые коэффициенты представимы в виде произведения трех неотрицательных вещественных чисел. В [8] предложена постановка, которая обобщает предложенный подход, в ней разложение коэффициентов происходит на два числа. В ходе вычислительного эксперимента в [9] была найдена экстремальная нижняя оценка для трехиндексной аксиальной задачи о назначениях с 1,2-декомпозиционной матрицей стоимостей в случае, когда количество значений индексов не превосходит трех.

Отдельные исследования посвящены так называемому многограннику ограничений аксиальной трехиндексной ЗОН, т.е. выпуклой оболочке всех допустимых решений. В [10, 11] исследовались различные классы вершин. Причем стоит заметить, что авторы рассматривают как трех-, так и многоиндексные аксиальные ЗОН [12].

В [13–16] авторами был рассмотрен подход, связанный с вероятностной модификацией "жадных" алгоритмов комбинаторной оптимизации посредством перехода к вероятностной постановке задачи. В данной статье предлагается детальное рассмотрение такого подхода для решения трехиндексной аксиальной ЗОН, а также результаты обширного вычислительного эксперимента.

2. Постановка задачи

Аксиальная трехиндексная задача о назначениях, которая является обобщением классической ЗОН, формулируется следующим образом: пусть заданы n^3 коэффициентов c_{ij}^k, $i, j, k = \overline{1, n}$. Требуется найти две перестановки φ и ψ такие, что

$$\sum_{i=1}^{n} c_{i\varphi(i)\psi(i)} \to \min_{\varphi,\psi\in S^n},$$

где S^n – множество всех перестановок чисел $\{1, \ldots, n\}$.

Можно представить данную задачу как задачу целочисленного линейного программирования. Ее математическая модель выглядит следующим образом:

$$L(X) = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} c_{ij}^k x_{ij}^k \to \min, \tag{1}$$

$$\sum_{j=1}^{n}\sum_{k=1}^{n} x_{ij}^k = 1, \quad i = \overline{1, n}, \tag{2}$$

$$\sum_{i=1}^{n}\sum_{k=1}^{n} x_{ij}^{k} = 1, \quad j = \overline{1,n}, \tag{3}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij}^{k} = 1, \quad k = \overline{1,n}, \tag{4}$$

$$x_{ij}^{k} = \{0,1\}, \quad i,j,k = \overline{1,n}. \tag{5}$$

Ограничения (2)–(5) задают допустимое множество Ω, а (1) является целевой функцией.

В [1] показано, что аксиальная трехиндексная задача является NP-трудной.

3. "Жадный" алгоритм решения

Для аксиальной задачи известен "жадный" алгоритм решения [17], основанный на поиске минимального элемента в плоских матрицах при фиксировании одного из индексов. Пример общей схемы одного из таких алгоритмов представлен на рис. 1.

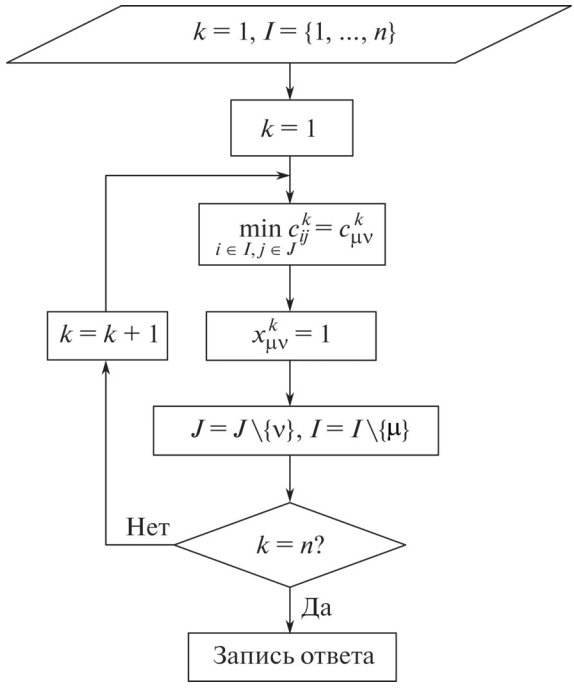

Рис. 1. Блок-схема алгоритма поиска минимального элемента для трехиндексной аксиальной задачи о назначениях.

Заметим, что вычислительная сложность данного алгоритма равна $O(n^3)$.

4. Адаптивный алгоритм решения

Для построения адаптивного аналога (далее адаптивного алгоритма) данного базового алгоритма осуществляется переход к вероятностной постановке задачи на основе рандомизации переменных [13].

Задача

$$L(X) \to \min_{X \in \Omega}$$

заменяется задачей вида

$$M(L(\mathrm{X})) \to \min_{\{\mathrm{X}\}: X \in \Omega},$$

где $\{\mathrm{X}\}$ – множество случайных величин X с реализациями X из множества Ω.

Здесь

$$\mathrm{X} = [\mathrm{X}^1, \ldots, \mathrm{X}^n], \quad \mathrm{X}^k = [\mathrm{X}_1^k, \ldots, \mathrm{X}_n^k]^{\mathrm{T}}, \quad \mathrm{X}_i^k = (\mathrm{X}_{i1}^k, \ldots, \mathrm{X}_{in}^k),$$

где X^k – матрица случайных величин размера $n \times n$, X_i^k – строка матрицы X^k, X_{ij}^k – случайная величина с рядом распределения, представленным в табл. 1.

Таблица 1. Ряд распределения X_{ij}^k

Возможные значения	1	0
Вероятность p	p_{ij}^k	$q_{ij}^k = 1 - p_{ij}^k$

*где $i, j, k = \overline{1, n}$.

Для каждого $k = \overline{1, n}$ вводятся полные группы событий A_{ij}^k, где A_{ij}^k – событие, заключающееся в том, что x_{ij}^k примет значение 1, т.е.

$$\sum_{i=1}^{n} \sum_{j=1}^{n} p_{ij}^k = 1, \quad k = \overline{1, n}.$$

В основе алгоритма лежат этапы I–III:

I. задание движения во множестве случайных величин X_{ij}^k;

II. решение неравенства — условия локального улучшения (УЛУ)

$$M\Big(L\left(\mathrm{X}^{N+1}\right) - L\left(\mathrm{X}^N\right)\Big) = M\Big(L\left(\mathrm{X}^{N+1}\right)\Big) - M\Big(L\left(\mathrm{X}^N\right)\Big) =$$

$$= M_{\mathrm{X}^{N+1}}\Big(M_{\mathrm{X}^N}\left(L\left(\mathrm{X}^{N+1}\right) - L\left(\mathrm{X}^N\right)\right)\Big) \leqslant 0, \tag{6}$$

где M_{X^N} – математическое ожидание случайных величин, зависящих от X^N;

III. пересчет вероятностей p_{ij}^k, $i, j, k = \overline{1, n}$ в соответствии с результатом УЛУ.

Заметим, что УЛУ (6) означает, что случайная величина X^{N+1} должна выбираться так, чтобы значение целевой функции в ней было лучше (меньше), чем значение целевой функции в предыдущей точке, или таким же. Так как случайные величины в данной задаче принимают значения только 1 или 0, то на математическое ожидание влияют только значения вероятностей. Их необходимый пересчет происходит на этапе III.

Опишем подробнее каждый из этапов.

I. Формула движения выбирается следующим образом:

$$\mathrm{X}^{N+1} = \overline{u}\mathrm{X}^N + u\mathrm{Y}^{N+1}, \tag{7}$$

где Y^{N+1} – неизвестная случайная величина, определяющая направление движения на $(N+1)$-м шаге, такая что $\mathrm{Y}^{N+1} \in \{\mathrm{X}\}$ и Y_{ij}^k имеет ряд распределения, представленный в табл. 2.

Таблица 2. Ряд распределения Y_{ij}^k

Возможные значения	1	0
Вероятность p	π_{ij}^k	$1-\pi_{ij}^k$

*где $i, j, k = \overline{1, n}$.

Будем также считать, что

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\pi_{ij}^k = 1, \quad k = \overline{1, n},$$

u – случайная величина с рядом распределения, представленным в табл. 3.

Таблица 3. Ряд распределения u

Возможные значения	1	0
Вероятность p	p_u	$q_u = 1 - p_u$

Лемма 1. Если движение задано формулой (7)*, то УЛУ можно записать в виде неравенства*

$$M_{\mathrm{X}}\Big(M_{\mathrm{Y}}\left(L\left(\mathrm{Y}^{N+1}\right) - L\left(\mathrm{X}^N\right)\right)\Big) \leqslant 0. \tag{8}$$

Перепишем формулу движения в покоординатной форме и возьмем математическое ожидание от обеих частей:

$$M\Big(\left(\mathrm{X}_{ij}^k\right)^{N+1}\Big) = M\Big(\overline{u}\left(\mathrm{X}_{ij}^k\right)^N + u\left(\mathrm{Y}_{ij}^k\right)^{N+1}\Big),$$

где $i, j = \overline{1, n}$ для каждого фиксированного $k = \overline{1, n}$.

С учетом того, что случайные величины u и Y^{N+1}, а также $\overline{u}$ и X^N вводятся как независимые, формула пересчета вероятностей адаптивного алгоритма будет выглядеть следующим образом:

$$(p_{ij}^k)^{N+1} = q_u(p_{ij}^k)^N + p_u(\pi_{ij}^k)^{N+1},$$

где $i, j = \overline{1, n}$ для каждого фиксированного $k = \overline{1, n}$.

Вероятности π_{ij}^k находятся из условия локального улучшения (8), параметр p_u задается некоторым числом от 0 до 1 (рекомендации по его выбору представлены в вычислительном эксперименте).

II. Условие локального улучшения (УЛУ)

Теорема 1. УЛУ (8) *для вероятностной постановки задачи представимо в виде*

$$c_{rs}^l - \left(\sum_{k=l+1}^{n} \sum_{j \in J} c_{rj}^k (p_{rj}^k)^N + \sum_{k=l+1}^{n} \sum_{i \in I} c_{is}^k (p_{is}^k)^N \right) - \\ - \left(c_{qh}^l - \left(\sum_{k=l+1}^{n} \sum_{j \in J} c_{qj}^k (p_{qj}^k)^N + \sum_{k=l+1}^{n} \sum_{i \in I} c_{ih}^k (p_{ih}^k)^N \right) \right) \leqslant 0 \tag{9}$$

для фиксированного $l = \overline{1,n}$ *и произвольных номеров* r, s, q, h,
где I, J *– множества допустимых номеров* i *и* j *соответственно.*

Пусть реализация случайной величины X^l зафиксирована (например, на предыдущей итерации алгоритма), т.е. $X^l = E_{qh}$, где E_{qh} – матрица с одной единицей на пересечении q-й строки и h-го столбца и остальными нулевыми элементами. Тогда для выполнения УЛУ по теореме 1 необходимо, чтобы выполнилось неравенство (9). Для этого номера r и s, $Y^l = E_{rs}$, необходимо выбрать так, чтобы величина $\left(c_{rs}^l - \left(\sum\limits_{k=l+1}^{n} \sum\limits_{j \in J} c_{rj}^k (p_{rj}^k)^N + \sum\limits_{k=l+1}^{n} \sum\limits_{i \in I} c_{is}^k (p_{is}^k)^N \right) \right)$ была минимальной.

В результате решение $\mathrm{Y}^{N+1} = [(\mathrm{Y}^1)^{N+1} \dots (\mathrm{Y}^n)^{N+1}]$ находится следующим образом:

$(\mathrm{Y}_{\mu\nu}^l)^{N+1}$: $\quad \pi_{\mu\nu}^l = 1$, если

$$\min_{r,s} \left(c_{rs}^l - \left(\sum_{k=l+1}^{n} \sum_{j \in J} c_{rj}^k (p_{rj}^k)^N + \sum_{k=l+1}^{n} \sum_{i \in I} c_{is}^k (p_{is}^k)^N \right) \right) = \\ = c_{\mu\nu}^l - \left(\sum_{k=l+1}^{n} \sum_{j \in J} c_{\mu j}^k (p_{\mu j}^k)^N + \sum_{k=l+1}^{n} \sum_{i \in I} c_{i\nu}^k (p_{i\nu}^k)^N \right),$$

$(\mathrm{Y}_{ij}^l)^{N+1}$: $\quad \pi_{ij}^l = 0, \quad i, j = \overline{1,n}, \quad j \neq \nu, \quad i \neq \mu,$

$(\mathrm{Y}^k)^{N+1} = (\mathrm{X}^k)^N, \quad k = \overline{l+1,n}.$

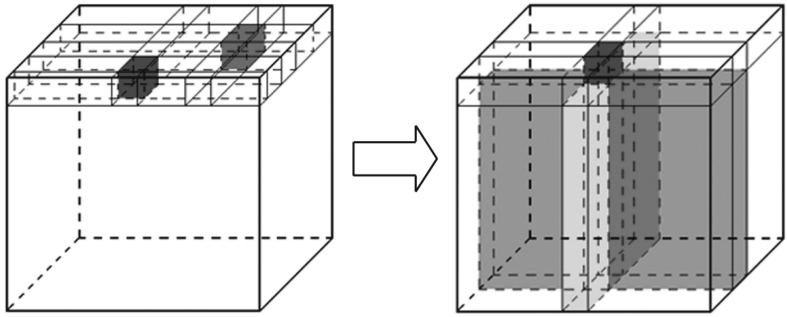

Рис. 2. Изменение “жадного” алгоритма.

Пересчет вероятностей при этом на этапе III с учетом того, что $\pi^l_{\mu\nu} = 1$, а остальные $\pi^l_{ij} = 0$, $i, j = \overline{1,n}$, $j \neq \nu$, $i \neq \mu$, происходит по формулам

$$(p^l_{\mu\nu})^{N+1} = (p^l_{\mu\nu})^N q_u + p_u, \tag{10}$$

$$(p^l_{ij})^{N+1} = (p^l_{ij})^N q_u, \quad i, j = \overline{1,n}, \quad j \neq \nu, \quad i \neq \mu, \tag{11}$$

$$(p^k_{ij})^{N+1} = (p^k_{ij})^N, \quad k = \overline{l+1,n}, \quad i, j = \overline{1,n}. \tag{12}$$

Данный этап отвечает за “адаптацию” вероятностей и дает общее название всему алгоритму, поскольку на каждой итерации производится подстройка вероятностей на основе предыдущих шагов для получения лучшего решения.

Таким образом, УЛУ позволяет модифицировать соответствующий шаг “жадного” алгоритма следующим образом:

при фиксированном k вместо

$$\min_{i\in I, j\in J} c^k_{ij} = c^k_{\mu\nu}$$

вычислить

$$\min_{i\in I, j\in J} \left(c^k_{ij} - \left(\sum_{l=k+1}^{n} \sum_{j\in J} c^l_{ij} (p^l_{ij})^N + \sum_{l=k+1}^{n} \sum_{i\in I} c^l_{ij} (p^l_{ij})^N \right) \right) =$$

$$= c^k_{\mu\nu} - \left(\sum_{l=k+1}^{n} \sum_{j\in J} c^l_{\mu j} (p^l_{\mu j})^N + \sum_{l=k+1}^{n} \sum_{i\in I} c^l_{i\nu} (p^l_{i\nu})^N \right),$$

где J – множество номеров j, I – множество номеров i,
зафиксировать $x^k_{\mu\nu} = 1$ (для получения рекордов).

На рис. 2 наглядно изображена модификация “жадного” алгоритма.

Замечание 1. В указанных формулах участвуют условные вероятности. На основании предположения о том, что все гипотезы распределены равномерно, безусловные вероятности $\overline{p}^{N+1}$ пересчитываются через условные p^{N+1}

по формуле

$$\overline{p}^{N+1} = \frac{N}{N+1}\left(\overline{p}^{N} + \frac{1}{N}p^{N+1}\right). \tag{13}$$

5. Алгоритм 1. Адаптивный алгоритм решения аксиальной трехиндексной ЗОН

1. Задать начальное распределение вероятностей для каждого $k = \overline{1,n}$

$$(p_{ij}^k)^1 = \frac{1}{n^2}, \quad i,j = \overline{1,n}.$$

Задать максимальное количество итераций $N_{\max}$. Положить $N = 1$.

2. Задать множества $J = \{1 \dots n\}$, $I = \{1 \dots n\}$, положить $k = 1$.

3. Вычислить

$$\min_{i\in I, j\in J}\left(c_{ij}^k - \left(\sum_{l=k+1}^{n}\sum_{j\in J} c_{ij}^l (p_{ij}^l)^N + \sum_{l=k+1}^{n}\sum_{i\in I} c_{ij}^l (p_{ij}^l)^N\right)\right) =$$

$$= c_{\mu\nu}^k - \left(\sum_{l=k+1}^{n}\sum_{j\in J} c_{\mu j}^l (p_{\mu j}^l)^N + \sum_{l=k+1}^{n}\sum_{i\in I} c_{i\nu}^l (p_{i\nu}^l)^N\right),$$

зафиксировать соответствующее назначение

$$(x_{\mu\nu}^k)^N = 1.$$

4. Пересчитать вероятности по формулам (10)–(12)

$$(p_{\mu\nu}^k)^{N+1} = (p_{\mu\nu}^k)^N q_u + p_u,$$

$$(p_{ij}^k)^{N+1} = (p_{ij}^k)^N q_u, \quad i,j = \overline{1,n}, \quad j \neq \nu, \quad i \neq \mu,$$

$$(p_{ij}^l)^{N+1} = (p_{ij}^l)^N, \quad l = \overline{k+1,n}, \quad i,j = \overline{1,n}.$$

5. Пересчитать безусловные вероятности по формуле (13).

6. Изменить множества $I,\ J:\ I = I \setminus \{\mu\},\ J = J \setminus \{\nu\}$.

7. Проверить: $k = n$? Если **нет**, то положить $k = k+1$ и перейти к шагу 3, если **да**, то переход к шагу 8.

8. Возможная смена рекорда $(X)^N$, $(L(X))^N$, $(P)^N$.

9. Проверить: $N = N_{\max}$? Если **нет**, то положить $N = N+1$ и перейти к шагу 2, если **да**, то ОСТАНОВ. В качестве ответа записывается последний рекорд.

Замечание 2. На шаге 3 алгоритма 1 для каждого элемента слоя осуществляется проход по двум "плоским" матрицам (сложность $O(n^2)$). Всего элементов на слое n^2, таким образом, сложность шага равна $O(n^4)$. Данный шаг повторяется для каждого $k = \overline{1,n}$, в результате чего сложность одной итерации алгоритма 1 составляет $O(n^5)$, что превышает сложность базового "жадного" алгоритма.

Разработан специальный метод построения дополнительной кубической матрицы $\tilde{C}$, при использовании которого вычислительная сложность данного адаптивного алгоритма на каждой итерации будет равна $O(n^3)$.

6. Алгоритм 2. Построение дополнительной матрицы $\tilde{C}$

1. Задать $i = 1$, $j = 1$.

2. Задать $\tilde{c}_{ij}^{n} = 0$.

3. Задать $k = n - 1$.

4. Вычислить $\tilde{c}_{ij}^{k} = \tilde{c}_{ij}^{k+1} + c_{ij}^{k+1} p_{ij}^{k+1}$.

5. Проверить: $k = 1$?

Если **нет**, то задать $k = k - 1$ и перейти к шагу 4.

Если **да**, то задать $j = j + 1$ и перейти к шагу 6.

6. Проверить: $j = n + 1$?

Если **нет**, то перейти к шагу 2

Если **да**, то задать $j = 1$ и $i = i + 1$ и перейти к шагу 7.

7. Проверить: $i = n + 1$?

Если **нет**, то перейти к шагу 2.

Если **да**, то ОСТАНОВ, матрица $\tilde{C}$ получена.

Данный алгоритм использует один проход по кубической матрице, таким образом, его вычислительная сложность равна $O(n^3)$.

Далее необходимо для каждого слоя матрицы $\tilde{C}$ реализовать стандартную процедуру подсчета суммы по строкам и по столбцам. Ее вычислительная сложность на слое равна $O(n^2)$, на всей кубической матрице – $O(n^3)$.

Обозначим сумму в j-м столбце на k-м слое через $Sum\_stolb_j^k$, а сумму в i-й строке на k-м слое через $Sum\_str_i^k$, $i, j, k = \overline{1, n}$.

Заменим в матрице $\tilde{C}$ каждый элемент на выражение

$$\tilde{c}_{ij}^{k} := c_{ij}^{k} - \left(Sum\_stolb_j^k + Sum\_str_i^k - \tilde{c}_{ij}^{k} \right) \tag{14}$$

в соответствии с (9). При этом слагаемое $\tilde{c}_{ij}^{k}$ вычитается, поскольку оно два раза учтено в подсчете сумм по строкам и столбцам.

В итоге шаг 3 алгоритма 1 заменяется следующей последовательностью шагов:

3.1. Составить дополнительную матрицу $\tilde{C}$ по алгоритму 2.

3.2. Посчитать сумму по строкам и столбцам на каждом слое кубической матрицы.

3.3. Применить формулу (14).

3.4. Вычислить

$$\min_{i \in I, j \in J} \tilde{c}_{ij}^{k} = \tilde{c}_{\mu\nu}^{k},$$

зафиксировать соответствующее назначение

$$x_{\mu\nu}^{k} = 1.$$

Таким образом, вместо прямого применения формулы (9) предлагается последовательное построение дополнительной матрицы в 4 этапа. С учетом того, что их вычислительные сложности равны $O(n^3)$, результирующая сложность каждой итерации алгоритма будет равна $O(n^3)$.

7. Вычислительный эксперимент

Для тестирования и анализа предложенного алгоритма был разработан программный комплекс в среде программирования Borland Delphi 7.0, содержащий в себе адаптивный и базовый алгоритмы.

Цели вычислительного эксперимента:

1) оценка влияния параметра p_u на работу алгоритма;

2) сравнение базового "жадного", итеративного "жадного" и адаптивного алгоритмов;

3) время работы адаптивного алгоритма.

Исходные данные вычислительного эксперимента:

— кубические матрицы трех размерностей: $10 \times 10 \times 10$, $50 \times 50 \times 50$, $100 \times 100 \times 100$;

— три варианта заполнения матриц: равномерное распределение чисел от 1 до 10 (обозначим как $R(10)$), от 1 до 50 ($R(50)$), от 1 до 100 ($R(100)$), от 1 до 300 ($R(300)$);

— различные значения вероятности p_u.

Оценивались следующие параметры: значение целевой функции и время работы алгоритмов.

Для каждого варианта исходных данных задавались 500 кубических матриц и вычислялось среднее арифметическое значений целевых функций.

В табл. 4 для разных p_u приведены результаты работы алгоритма для задач размерности $100 \times 100 \times 100$ при равномерном распределении чисел от 1 до 100.

Таблица 4. Среднее арифметическое значений целевых функций

Значение p_u	10 итераций	50 итераций	100 итераций
$p_u = 0{,}01$	133,6	121,5	118,1
$p_u = 0{,}1$	134,96	126,3	123,68
$p_u = 0{,}2$	135,85	127,94	125,25
$p_u = 0{,}3$	137,78	128,41	125,77
$p_u = 0{,}4$	137,7	128,58	126,43
$p_u = 0{,}5$	138,87	128,67	126,28
$p_u = 0{,}6$	139,07	128,82	126,2
$p_u = 0{,}7$	139,77	129,5	126,78
$p_u = 0{,}8$	138,36	129,15	126,45
$p_u = 0{,}9$	138,91	129,12	125,65
$p_u = 1$	137,82	129,45	125,68

На рис. 3 представлены изменения значений целевой функции в процессе работы алгоритма на первых 30 итерациях при различных значениях параметра p_u.

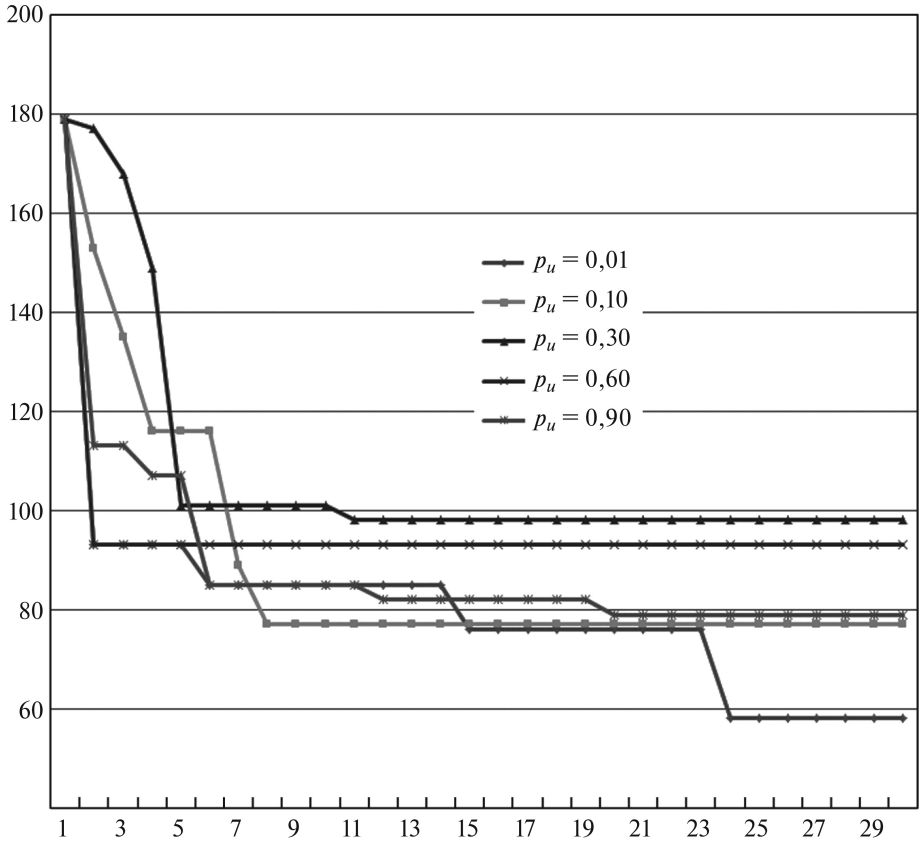

Рис. 3. Изменение значения целевой функции при разных p_u.

Рассмотрим задание параметра p_u в виде ступенчатой функции, зависящей от количества итераций

$$p_u(N) = \begin{cases} 0{,}01, & \text{если } N \leqslant \dfrac{N_{\max}}{5}; \\ 0{,}1, & \text{если } \dfrac{N_{\max}}{5} < N \leqslant \dfrac{2N_{\max}}{5}; \\ 0{,}5, & \text{если } \dfrac{2N_{\max}}{5} < N \leqslant \dfrac{3N_{\max}}{5}; \\ 0{,}1, & \text{если } \dfrac{3N_{\max}}{5} < N \leqslant \dfrac{4N_{\max}}{5}; \\ 0{,}01, & \text{если } N > \dfrac{4N_{\max}}{5}, \end{cases} \tag{15}$$

где N – номер текущей итерации, $N_{\max}$ – максимальное количество итераций.

Такой подход позволит алгоритму избежать “застревания” в локальных минимумах и, соответственно, улучшить решение.

В табл. 5 представлены результаты работы алгоритма для задач размерности $100 \times 100 \times 100$ при равномерном распределении чисел от 1 до 100 с параметром p_u, заданным в (15).

Таблица 5. Параметр p_u, заданный формулой

10 итераций	50 итераций	100 итераций
132,42	121,0	117,51

Теперь рассмотрим сравнение адаптивного и базового "жадного" алгоритмов.

Так как предложенный адаптивный алгоритм является итеративным, то целесообразно проводить его сравнение не только с "жадным" алгоритмом, но и с его итеративной модификацией, заключающейся в изменении порядка следования слоев на каждой итерации.

В табл. 6 представлены результаты работы алгоритмов для задач размерности $100 \times 100 \times 100$. Для итеративных алгоритмов число итераций равно 100.

Таблица 6. Среднее арифметическое значений целевых функций

Разброс	"Жадный" алгоритм	Итеративный "жадный" алгоритм	Адаптивный алгоритм	
			$p_u = 0{,}01$	$p_u(N)$
R(10)	106,9	100,24	100,0	100,0
R(50)	146,1	108,74	106,16	105,62
R(100)	202,9	123,72	118,14	117,51
R(300)	396,66	186,68	178,68	175,92

В табл. 7 приведены средние значения времени работы алгоритма (в секундах) для указанного числа итераций на задачах разных размерностей.

Таблица 7. Время работы алгоритма

Размерность задачи	10 итераций	50 итераций	100 итераций	200 итераций
$10 \times 10 \times 10$	0,0009	0,0024	0,006	0,0104
$50 \times 50 \times 50$	0,061	0,29	0,57	1,17
$100 \times 100 \times 100$	0,76	3,81	7,49	15,28

В табл. 8 приведены средние значения целевых функций при росте числа итераций алгоритма.

Таблица 8. Средние значения целевых функций при $p_u = 0{,}1$

10 итераций	50 итераций	100 итераций	200 итераций	1000 итераций
134,96	126,3	123,68	122,8	121,6

На основе проведенного вычислительного эксперимента можно сделать следующие выводы.

1. Адаптивный алгоритм дает результаты лучше, чем "жадный" (в 1,4–1,6 раз) и итеративный "жадный" алгоритмы (табл. 6).

2. Большое влияние на итоговый результат оказывает параметр p_u. Эксперимент показал, что при малых значениях p_u и в случае его функциональной зависимости от числа итераций алгоритм работает эффективнее (табл. 4 и 5).

3. Увеличение количества итераций приводит к улучшению решения (рис. 3). Однако большое число итераций приводит к незначительным улучшениям ответа (табл. 8), поэтому их количество следует соотносить с временем работы алгоритма (табл. 7).

4. В ряде случаев адаптивный алгоритм решает задачу точно! Например, при размерности задачи $100\times100\times100$ (табл. 6) и равномерном распределении чисел от 1 до 10 по кубической матрице, т.е. при небольшом разбросе значений элементов матрицы относительно размерности задачи, алгоритм дает оптимальное значение целевой функции, равное 100.

При размерности $50 \times 50 \times 50$ и равномерном распределении чисел от 1 до 10 алгоритм дает оптимальное значение целевой функции, равное 50, в среднем на 95 матрицах из 100 при 50 итерациях.

В ходе вычислительного эксперимента оказалось, что задание p_u в виде функциональной зависимости от количества итераций позволяет улучшить результат работы алгоритма. В связи с этим дальнейшие исследования будут направлены на изучение и анализ результатов работы адаптивного алгоритма с изменяющимся в процессе работы параметром p_u, а также на сравнение данного метода с другими эвристическими алгоритмами.

8. Заключение

В статье предложен вероятностный аналог "жадного" алгоритма поиска минимального элемента. Он назван адаптивным ввиду того, что на каждой итерации происходит подстройка вероятностей с учетом результатов на предыдущей итерации. Оригинальность метода заключается в том, что на формирование решения влияют не только "качества" самого элемента, но возможные потери при его выборе, что является несомненным преимуществом и позволяет избежать перебора заведомо "плохих" решений. Также отличительной особенностью алгоритма является учет его предыдущих шагов за счет "адаптации" матрицы вероятностей, которая позволяет организовать направленный перебор допустимых решений.

В статье представлены результаты вычислительного эксперимента, на основании которых сделаны определенные выводы по работе алгоритма.

ПРИЛОЖЕНИЕ

Доказательство леммы 1. При выборе формулы движения (7) значение целевой функции в точке X^{N+1} будет равно

$$L(\mathrm{X}^{N+1}) = \overline{u}L(\mathrm{X}^{N}) + uL(\mathrm{Y}^{N+1}).$$

Подставим данное выражение в УЛУ (6)

$$M\Big(L\left(\mathrm{X}^{N+1}\right)-L\left(\mathrm{X}^{N}\right)\Big)=M\Big(\overline{u}L\left(\mathrm{X}^{N}\right)+uL\left(\mathrm{Y}^{N+1}\right)-L\left(\mathrm{X}^{N}\right)\Big)=$$
$$=q_uM\Big(L\left(\mathrm{X}^{N}\right)\Big)+p_uM\Big(L\left(\mathrm{Y}^{N+1}\right)\Big)-M\Big(L\left(\mathrm{X}^{N}\right)\Big)=$$
$$=(q_u-1)M\Big(L\left(\mathrm{X}^{N}\right)\Big)+p_uM\Big(L\left(\mathrm{Y}^{N+1}\right)\Big)=$$
$$=p_u\Big(M\left(L\left(\mathrm{Y}^{N+1}\right)\right)-M\left(L\left(\mathrm{X}^{N}\right)\right)\Big)=$$
$$=p_uM_{\mathrm{X}}\Big(M_{\mathrm{Y}}\left(L\left(\mathrm{Y}^{N+1}\right)-L\left(\mathrm{X}^{N}\right)\right)\Big)\leqslant 0.$$

Так как $p_u>0$, то в итоге получается, что

$$M_{\mathrm{X}}\Big(M_{\mathrm{Y}}\left(L\left(\mathrm{Y}^{N+1}\right)-L\left(\mathrm{X}^{N}\right)\right)\Big)\leqslant 0.$$

Лемма 1 доказана.

Доказательство теоремы 1. Заметим, что неравенство (8) имеет бесчисленное множество решений.

$$M_{\mathrm{X}}\Big(M_{\mathrm{Y}}\left(L\left(\mathrm{Y}^{N+1}\right)-L\left(\mathrm{X}^{N}\right)\right)\Big)=$$
$$=M_{\mathrm{X}}\left(M_{\mathrm{Y}}\left(\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}c_{ij}^{k}(\mathrm{Y}_{ij}^{k})^{N+1}-\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}c_{ij}^{k}(\mathrm{X}_{ij}^{k})^{N}\right)\right)\leqslant 0.$$

Воспользуемся идеей алгоритма покоординатного спуска. Решим неравенство по неизвестной матрице Y^1. Переменные $\mathrm{Y}^2,\ldots,\mathrm{Y}^n$ зафиксируем, т.е. будем считать, что случайные величины $(\mathrm{Y}_{ij}^{k})^{N+1}$ и $(\mathrm{X}_{ij}^{k})^{N}$ имеют одинаковое распределение, а именно $(\pi_{ij}^{k})^{N+1}=(p_{ij}^{k})^{N}$ при $k=\overline{2,n}$, $i,j=\overline{1,n}$.

Учитывая, что математическое ожидание от произведения независимых случайных величин есть произведение их математических ожиданий, а также используя формулу для условной вероятности, УЛУ можно переписать следующим образом:

$$M_{\mathrm{X}}\Big(M_{\mathrm{Y}}\left(L\left(\mathrm{Y}^{N+1}\right)-L\left(\mathrm{X}^{N}\right)\right)\Big)=$$
$$=M_{\mathrm{X}^1}M_{\mathrm{X}|\mathrm{X}^1}\Big(M_{\mathrm{Y}^1}M_{\mathrm{Y}|\mathrm{Y}^1}\left(L\left(\mathrm{Y}^{N+1}\right)-L\left(\mathrm{X}^{N}\right)\right)\Big)\leqslant 0.$$

С учетом того, что

$$L(\mathrm{X})=\sum_{i=1}^{n}\sum_{j=1}^{n}c_{ij}^{1}\mathrm{X}_{ij}^{1}+\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=2}^{n}c_{ij}^{k}\mathrm{X}_{ij}^{k},$$
$$L(\mathrm{Y})=\sum_{i=1}^{n}\sum_{j=1}^{n}c_{ij}^{1}\mathrm{Y}_{ij}^{1}+\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=2}^{n}c_{ij}^{k}\mathrm{Y}_{ij}^{k},$$

УЛУ перепишется в виде

$$M_{\mathrm{Y}^1}\left(\sum_{i=1}^{n}\sum_{j=1}^{n}c_{ij}^{1}(\mathrm{Y}_{ij}^{1})^{N+1}+M_{\mathrm{Y}|\mathrm{Y}^1}\left(\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=2}^{n}c_{ij}^{k}(\mathrm{Y}_{ij}^{k})^{N+1}\right)\right)-$$

$$-M_{\mathrm{X}^1}\left(\sum_{i=1}^{n}\sum_{j=1}^{n}c_{ij}^{1}(\mathrm{X}_{ij}^{1})^{N}+M_{\mathrm{X}|\mathrm{X}^1}\left(\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=2}^{n}c_{ij}^{k}(\mathrm{X}_{ij}^{k})^{N}\right)\right)\leqslant 0 \tag{П.1}$$

или

$$M_{\mathrm{Y}^1}\left(\sum_{i=1}^{n}\sum_{j=1}^{n}c_{ij}^{1}(\mathrm{Y}_{ij}^{1})^{N+1}+M_{\mathrm{Y}|\mathrm{Y}^1}\left(L\left(\tilde{\mathrm{Y}}^{N+1}\right)\right)\right)-$$

$$-M_{\mathrm{X}^1}\left(\sum_{i=1}^{n}\sum_{j=1}^{n}c_{ij}^{1}(\mathrm{X}_{ij}^{1})^{N}+M_{\mathrm{X}|\mathrm{X}^1}\left(L\left(\tilde{\mathrm{X}}^{N}\right)\right)\right)\leqslant 0, \tag{П.2}$$

где $\tilde{\mathrm{Y}}=[\mathrm{Y}^2\dots\mathrm{Y}^n]$, $\tilde{\mathrm{X}}=[\mathrm{X}^2\dots\mathrm{X}^n]$.

Учитывая, что $\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n}p_{ij}^{1}=1$, зафиксируем реализацию случайной величины X^1. Предположим, что $x_{qh}^{1}=1$ ($x_{ij}^{1}=0$, $i\neq q$, $j\neq h$ соответственно).

Выражение $M_{\mathrm{X}^1}\left(\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n}c_{ij}^{1}(\mathrm{X}_{ij}^{1})^{N}+M_{\mathrm{X}|\mathrm{X}^1}\left(L(\tilde{\mathrm{X}}^{N})\right)\right)$ из формулы (П.2) примет вид $c_{qh}^{1}+M_{\mathrm{X}|\mathrm{X}^1=E_{qh}}\left(L(\tilde{\mathrm{X}}^{N})\right)$. При фиксировании номеров q, h и с учетом того, что $\{\mathrm{X}\}:X\in\Omega$, реализация условного математического ожидания $M_{\mathrm{X}|\mathrm{X}^1=E_{qh}}\left(L(\tilde{\mathrm{X}}^{N})\right)$ примет вид

$$c_{qh}^{1}+M_{\mathrm{X}|\mathrm{X}^1=E_{qh}}\left(L(\tilde{\mathrm{X}}^{N})\right)=$$

$$=c_{qh}^{1}+M_{\mathrm{X}|\mathrm{X}^1=E_{qh}}\left(\sum_{\substack{i=1\\i\neq q}}^{n}\sum_{j=1}^{n}\sum_{k=2}^{n}c_{ij}^{k}(\mathrm{X}_{ij}^{k})^{N}+\sum_{i=1}^{n}\sum_{\substack{j=1\\j\neq h}}^{n}\sum_{k=2}^{n}c_{ij}^{k}(\mathrm{X}_{ij}^{k})^{N}\right)=$$

$$=c_{qh}^{1}+\sum_{\substack{i=1\\i\neq q}}^{n}\sum_{j=1}^{n}\sum_{k=2}^{n}c_{ij}^{k}(p_{ij}^{k})^{N}+\sum_{i=1}^{n}\sum_{\substack{j=1\\j\neq h}}^{n}\sum_{k=2}^{n}c_{ij}^{k}(p_{ij}^{k})^{N}=$$

$$=c_{qh}^{1}+\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=2}^{n}c_{ij}^{k}(p_{ij}^{k})^{N}-\left(\sum_{k=2}^{n}\sum_{j=1}^{n}c_{qj}^{k}(p_{qj}^{k})^{N}+\sum_{k=2}^{n}\sum_{i=1}^{n}c_{ih}^{k}(p_{ih}^{k})^{N}\right). \tag{П.3}$$

Аналогично, при каждой реализации $\mathrm{Y}^1:V^1=E_{rs}$ и с учетом того, что $\mathrm{Y}^{N+1}\in\{\mathrm{X}\}$: $X\in\Omega$, а также, что $(\pi_{ij}^{k})^{N+1}=(p_{ij}^{k})^{N}$ при $k=\overline{2,n}$, $i,j=\overline{1,n}$,

выражение $M_{\mathrm{Y}^1}\left(\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n}c_{ij}^1(\mathrm{Y}_{ij}^1)^{N+1}+M_{\mathrm{Y}|\mathrm{Y}^1}\left(L(\tilde{\mathrm{Y}}^{N+1})\right)\right)$ примет вид

$$c_{rs}^1+\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=2}^{n}c_{ij}^k(p_{ij}^k)^N-\left(\sum_{k=2}^{n}\sum_{j=1}^{n}c_{rj}^k(p_{rj}^k)^N+\sum_{k=2}^{n}\sum_{i=1}^{n}c_{is}^k(p_{is}^k)^N\right). \tag{П.4}$$

УЛУ (П.1) с учетом (П.3) и (П.4) преобразуется следующим образом:

$$M_{\mathrm{Y}^1}\left(\sum_{i=1}^{n}\sum_{j=1}^{n}c_{ij}^1(\mathrm{Y}_{ij}^1)^{N+1}+M_{\mathrm{Y}|\mathrm{Y}^1}\left(L(\tilde{\mathrm{Y}}^{N+1})\right)\right)-$$

$$-M_{\mathrm{X}^1}\left(\sum_{i=1}^{n}\sum_{j=1}^{n}c_{ij}^1(\mathrm{X}_{ij}^1)^N+M_{\mathrm{X}|\mathrm{X}^1}\left(L(\tilde{\mathrm{X}}^N)\right)\right)=$$

$$=c_{rs}^1+\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=2}^{n}c_{ij}^k(p_{ij}^k)^N-\left(\sum_{k=2}^{n}\sum_{j=1}^{n}c_{rj}^k(p_{rj}^k)^N+\sum_{k=2}^{n}\sum_{i=1}^{n}c_{is}^k(p_{is}^k)^N\right)-$$

$$-\left(c_{qh}^1+\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=2}^{n}c_{ij}^k(p_{ij}^k)^N-\left(\sum_{k=2}^{n}\sum_{j=1}^{n}c_{qj}^k(p_{qj}^k)^N+\sum_{k=2}^{n}\sum_{i=1}^{n}c_{ih}^k(p_{ih}^k)^N\right)\right)=$$

$$=c_{rs}^1-\left(\sum_{k=2}^{n}\sum_{j=1}^{n}c_{rj}^k(p_{rj}^k)^N+\sum_{k=2}^{n}\sum_{i=1}^{n}c_{is}^k(p_{is}^k)^N\right)-$$

$$-\left(c_{qh}^1-\left(\sum_{k=2}^{n}\sum_{j=1}^{n}c_{qj}^k(p_{qj}^k)^N+\sum_{k=2}^{n}\sum_{i=1}^{n}c_{ih}^k(p_{ih}^k)^N\right)\right)\leqslant 0. \tag{П.5}$$

Вышеизложенные рассуждения можно обобщить для неизвестной матрицы $\mathrm{Y}^l, l=\overline{1,n}$. В результате получим

$$c_{rs}^l-\left(\sum_{k=l+1}^{n}\sum_{j\in J}c_{rj}^k(p_{rj}^k)^N+\sum_{k=l+1}^{n}\sum_{i\in I}c_{is}^k(p_{is}^k)^N\right)-$$

$$-\left(c_{qh}^l-\left(\sum_{k=l+1}^{n}\sum_{j\in J}c_{qj}^k(p_{qj}^k)^N+\sum_{k=l+1}^{n}\sum_{i\in I}c_{ih}^k(p_{ih}^k)^N\right)\right)\leqslant 0,$$

где I, J – множества допустимых номеров i и j соответственно.

Теорема 1 доказана.

СПИСОК ЛИТЕРАТУРЫ

1. *Karp R.M.* Reducibility Among Combinatorial Problems, in Complexity of Computer Computations // N.Y.: Plenum Press, 1972. P. 85–103.

2. *Hansen P., Kaufman L.* A Primal-Dual Algorithm for the Three-dimensional Assignment Problem // Cahiers du GERO. 1973. V. 15. No. 3. P. 327–336.
3. *Pierskalla W.P.* The Tri-substitution Method for the Three-dimensional Assignment Problem // Canad. Oper. Res. Soc. J. 1967. V. 5. P. 71–81.
4. *Aiex R.M., Resende M.G.C., Pardalos P.M., Toraldo G.* GRASP with Path Relinking for Three-index Assignment // INFORMS J. Comput. 2005. V. 17. P. 224–247.
5. *Huang G., Lim A.* A Hybrid Genetic Algorithm for the Three-index Assignment Problem // Eur. J. Oper. Res. 2006. V. 172. P. 249–257.
6. *Euler R.* Odd Cycles and a Class of Facets of the Axial 3-index Assignment Polytope // Appl. Math. (Zastowania Matematyki). 1987. V. 19. P. 375–386.
7. *Burkard R.E., Rudolf R., Woeginger G.J.* Three-dimensional Axial Assignments Problems with Decomposable Cost Coefficients // Discret Appl. Math. 1996. V. 65. P. 123–139.
8. *Dell'Amico M., Lodi A., Martello S.* Efficient Algorithms and Codes for the k-cardinality Assignment Problem // Discret. Appl. Math. 2001. V. 110. P. 25–40.
9. *Афраймович Л.Г.* Нижняя оценка для трехиндексной аксиальной задачи о назначениях с 1,2-декомпозиционной матрицей стоимостей // Вест. Волж. гос. акад. водного транспорта. 2016. Вып. 49. С. 25–29.
10. *Кравцов В.М., Кравцов М.К.* Характеризация типов максимально нецелочисленных вершин релаксационного многогранника четырехиндексной аксиальной задачи о назначениях // Журн. вычисл. мат. и мат. физики. 2013. Вып. 53. № 5. С. 825.
11. *Кравцов М.К., Кравцов В.М.* О типах максимально нецелочисленных вершин релаксационного многогранника четырехиндексной аксиальной задачи о назначениях // Изв. высш. уч. заведений. Математика. 2012. № 3. С. 9–16.
12. *Цидулко О.Ю.* О разрешимости 8-индексной аксиальной задачи о назначениях на одноциклических подстановках // Дискрет. анализ и исслед. операций. 2013. Т. 20. № 5 (113). С. 66–83.
13. *Львович Я.Е., Каплинский А.И., Чернышова Г.Д., Черных О.И.* Конструирование адаптивных схем перебора для решения дискретных задач оптимизации / Актуальные проблемы фундаментальных наук. 1991. С. 44–46.
14. *Медведев С.Н.* Адаптивные алгоритмы решения трехиндексных задач о назначениях // Современные методы прикладной математики, теории управления и компьютерных технологий: сб. трудов VI Междунар. конф. (Воронеж, 10–16 сентября 2013 г.) 2013. С. 153–156.
15. *Медведев С.Н., Чернышова Г.Д.* Использование адаптивных алгоритмов для решения трехиндексной задачи о назначениях // Вестн. факультета прикл. мат., информ. и механики, 2010. Вып. 8. С. 148–155.
16. *Медведева О.А., Медведев С.Н.* Задача комплектования штатов // Актуальные проблемы прикладной математики, информатики и механики : сб. трудов Междунар. конф. (Воронеж, 26–28 ноября 2012 г.) Ч. 2. 2012. С. 203–208.
17. *Гольштейн Е.Г., Юдин Д.Б.* Задачи линейного программирования транспортного типа. М.: Наука, 1969.

Статья представлена к публикации членом редколлегии А.А. Лазаревым.

Поступила в редакцию 31.07.2017

После доработки 05.06.2018

Принята к публикации 13.11.2018