УДК 519.68

ВЕРОЯТНОСТНЫЙ ПОДХОД К РЕШЕНИЮ ТРЕХИНДЕКСНОЙ АКСИАЛЬНОЙ ЗАДАЧИ О НАЗНАЧЕНИЯХ. ВЫЧИСЛИТЕЛЬНЫЙ ЭКСПЕРИМЕНТ

А. Г. Трегубов, С. Н. Медведев

Воронежский государственный университет

Поступила в редакцию 10.10.2015 г.

Аннотация. В статье представлена формулировка трёхиндексной аксиальной задачи о назначениях, предложен адаптивный алгоритм решения, основанные на переходе к вероятностной постановке задачи и проведено сравнение данного алгоритма с базовым «жадным».
Ключевые слова: трёхиндексная аксиальная задача о назначениях, «жадный» алгоритм, вероятность.
Annotation. The article presents a statement of axial3-indexassignment problem. Offered adaptive algorithm based on transition to probability problem and comparison between adaptive and «greedy» algorithms.
Keywords: axial3-indexassignment problem, «greedy» algorithm, probability.

Одной из самых известных задач дискретной оптимизации является задача о назначениях (ЗОН), которая в своей классической формулировке сводится к задаче о распределении претендентов по рабочим местам с целью минимизации затрат.

Естественным обобщением классической ЗОН является многоиндексная ЗОН, которая имеет две наиболее известные разновидности: аксиальная и планарная трёхиндексные ЗОН. В данной работе остановимся на исследовании аксиальной задачи.

Для получения математической записи аксиальной ЗОН введем n^3 переменных вида:

$$x_{ij}^k = \begin{cases} 1, & \text{если } i\text{-й претендент назначен} \\ & \text{на } j\text{-е место,} \\ 0, & \text{иначе,} \end{cases}$$

где $i, j = \overline{1, n}$.

В результате задача принимает следующий вид:

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

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

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

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

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

Аксиальная ЗОН является N-трудной, что стимулирует разработку приближённых алгоритмов, среди которых выделяют класс «жадных» алгоритмов [1, 3]. Идея алгоритмов этого типа связана с последовательным построением допустимых вариантов. На каждом этапе делается попытка максимально достичь некоторой локальной вспомогательной цели. При использовании данного подхода не делается попыток оценки ни предыдущих, ни последующих шагов алгоритма. На пути использования таких оценок (за счёт углубления памяти алгоритма, оценивания средних значений функций и т. п.) появляются различные модификации «жадных» алгоритмов.

Для трёхиндексной аксиальной задачи известен базовый «жадный» алгоритм, основанный на поиске минимального элемента в плоских матрицах при фиксировании одного из индексов [1]. А именно, при каждом фик-

© Трегубов А. Г., Медведев С. Н., 2015

сированном $k=\overline{1,n}$ ищется локальный минимум

$$\min_{i,j}\{c_{ij}^k\}=c_{\mu\nu}^k$$

и фиксируется соответствующее назначение $x_{\mu\nu}^k=1.$

На основе данного базового метода предлагается построение адаптивного алгоритма решения аксиальной трехиндексной ЗОН (1)–(5). Для этого осуществляется переход к вероятностной постановке задачи на основе рандомизации переменных [2].

Задача минимизации целевой функции

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

заменяется задачей минимизации её математического ожидания

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

где $\{X\}$ – множество случайных величин X с реализациями из множества Ω, $\{X\}:X\in\Omega.$

Здесь

$X=[X^1\ldots X^n]$ – матрица случайных величин,

$X^k=[X_1^k\ldots X_n^k]^T$ – матрица $n\times n$ случайных величин,

$X_i^k=(X_{i1}^k\ldots X_{in}^k)$ – строка матрицы X^k

X_{ij}^k – случайная величина с рядом распределения:

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

где $i,j,k=\overline{1,n}.$

Вводятся полные группы событий A_{ij}^k, для каждого $k=\overline{1,n}$, где A_{ij}^k – событие, заключающееся в том, что x_{ij}^k примет значение 1, для каждого $k=\overline{1,n}$ т. е. $\sum_{i=1}^n\sum_{j=1}^n p_{ij}^k=1,\ k=\overline{1,n}.$

В основе адаптивного алгоритма лежат следующие этапы [4]:

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

II. Решение неравенства – условия локального улучшения (УЛУ)

$$M_X\left(M_Y\left(L(X^{N+1})-L(X^N)\right)\right)\le 0. \quad (6)$$

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

Опишем подробнее каждый из этапов.

I. Формулу движения выберем следующим образом:

$$X^{N+1}=\overline{u}X^N+uY^{N+1}, \quad (7)$$

где Y^{N+1} – неизвестная случайная величина, определяющая направление движения на $(N+1)$-м шаге, такая, что $Y^{N+1}\in\{X\}$, и имеющая ряд распределения:

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

Иными словами, движение осуществляется так: либо остаемся в той же точке, либо переходим в новую, которая «лучше». Для того, чтобы её найти решается неравенство УЛУ.

Будем также считать, что

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

u – дискретная случайная величина с рядом распределения

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

Если взять математическое ожидание от обеих частей равенства (7), то с учетом того, что случайные величины u и Y^{N+1}, а также $\overline{u}$ и X^N вводятся как независимые, получим формулу пересчёта вероятностей адаптивного алгоритма:

$$(p_{ij}^k)^{N+1}=q_u(p_{ij}^k)^N+p_u(\pi_{ij}^k)^{N+1}, \quad (8)$$

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

Вероятности π_{ij}^k находятся из условия локального улучшения (6), параметр p_u задаётся программно.

II. Условие локального улучшения (6) для вероятностной постановки задачи можно представить в виде

$$c_{rs}^l-\left(\sum_{k=l+1}^n\sum_{j=1}^n c_{rj}^k(p_{rj}^k)^N+\sum_{k=l+1}^n\sum_{i=1}^n c_{is}^k(p_{is}^k)^N\right)-$$
$$-\left(c_{qh}^l-\left(\sum_{k=l+1}^n\sum_{j=1}^n c_{qj}^k(p_{qj}^k)^N+\right.\right.$$
$$\left.\left.+\sum_{k=l+1}^n\sum_{i=1}^n c_{ih}^k(p_{ih}^k)^N\right)\right)\le 0, \quad (9)$$

для фиксированного $l=\overline{1,n}$ и произвольных номеров $r,\ s,\ q,\ h.$

Пусть E_{qh} – матрица с одной единицей на пересечении q-го столбца и h-й строки и остальными нулевыми элементами.

На каждой итерации алгоритма реализация случайной величины $X^l = E_{qh}$ уже зафиксирована, тогда номера r и s, $Y^l = E_{rs}$, необходимо выбрать так, чтобы выполнилось неравенство (9). Для гарантированного выполнения номера r и s необходимо выбрать так, чтобы величина

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

была минимальной.

В результате решение $Y^{N+1} = [(Y^l)^{N+1} \dots (Y^n)^{N+1}]$, где Y^k – матрицы размером $n \times n$, $k = \overline{1,n}$, с учётом того, что $Y^1, \dots, Y^{l-1}$ уже найдены, выбирается следующим образом:

$$(Y_{\mu\nu}^l)^{N+1} = 1,$$

если

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

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

$$(Y_{ij}^l)^{N+1} = 0, \ i, j = \overline{1,n}, \ i \neq \mu, \ j \neq \nu,$$

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

III. Пересчёт вероятностей, с учётом того, что $\pi_{\mu\nu}^l = 1$, а остальные $\pi_{ij}^l = 0$, $i, j = \overline{1,n}$, $j \neq \nu$, $i \neq \mu$, происходит по формулам

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

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

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

Таким образом, условие локального улучшения позволяет модифицировать соответствующий шаг «жадного» алгоритма следующим образом:

при фиксированном k вместо

$$\min_{i,j} c_{ij}^k = c_{\mu\nu}^k$$

вычислить

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

Зафиксировать $x_{\mu\nu}^k = 1$ (для получения рекордов).

Замечание. В указанных формулах участвуют условные вероятности, которые необходимо пересчитать в безусловные.

В результате адаптивный алгоритм решения трехиндексной аксиальной задачи о назначениях (1)–(5) можно представить схемой (рис. 1).

Для анализа эффективности построенного адаптивного алгоритма была осуществлена программная реализация данного алгоритма и базового «жадного».

В ходе вычислительного эксперимента проводилось по 100 испытаний для каждого алгоритма: базового «жадного» и адаптивного с разным числом итераций, для матриц разной размерности. Матрицы заполнялись случайными целыми числами из диапазона от 1 до 100. Затем считалось среднее значение целевых функций и среднее значение времени выполнения. В приведённых ниже таблицах значения целевых функций округлены до целой части.

В табл. 1, 2 приведены результаты работы программной реализации двух алгоритмов для матриц разной размерности.

Выводы:

- Адаптивный алгоритм даёт значения целевых функций существенно лучше, чем соответствующий базовый $\left(\dfrac{L_{баз}}{L_{адапт}} \geq 1,5 \right)$;
- Время работы адаптивного алгоритма линейно зависит от количества итераций;
- Для матриц меньшей размерности достаточно меньшего числа итераций, т.к. начиная с некоторого значения числа итераций улучшение значения целевой функции становится незначительным.

На рис. 2 представлен график работы алгоритма с условием фиксирования рекорда и без. График отображает результат одного опыта для матрицы $10 \times 10 \times 10$ с параметром $p_u = 0,5$.

По оси Y расположены полученные значения целевой функции на данной итерации, по оси X – номер итерации.

Можно заметить, что далеко не на каждом шаге получается хорошее решение, однако в перспективе, за счёт фиксирования рекорда

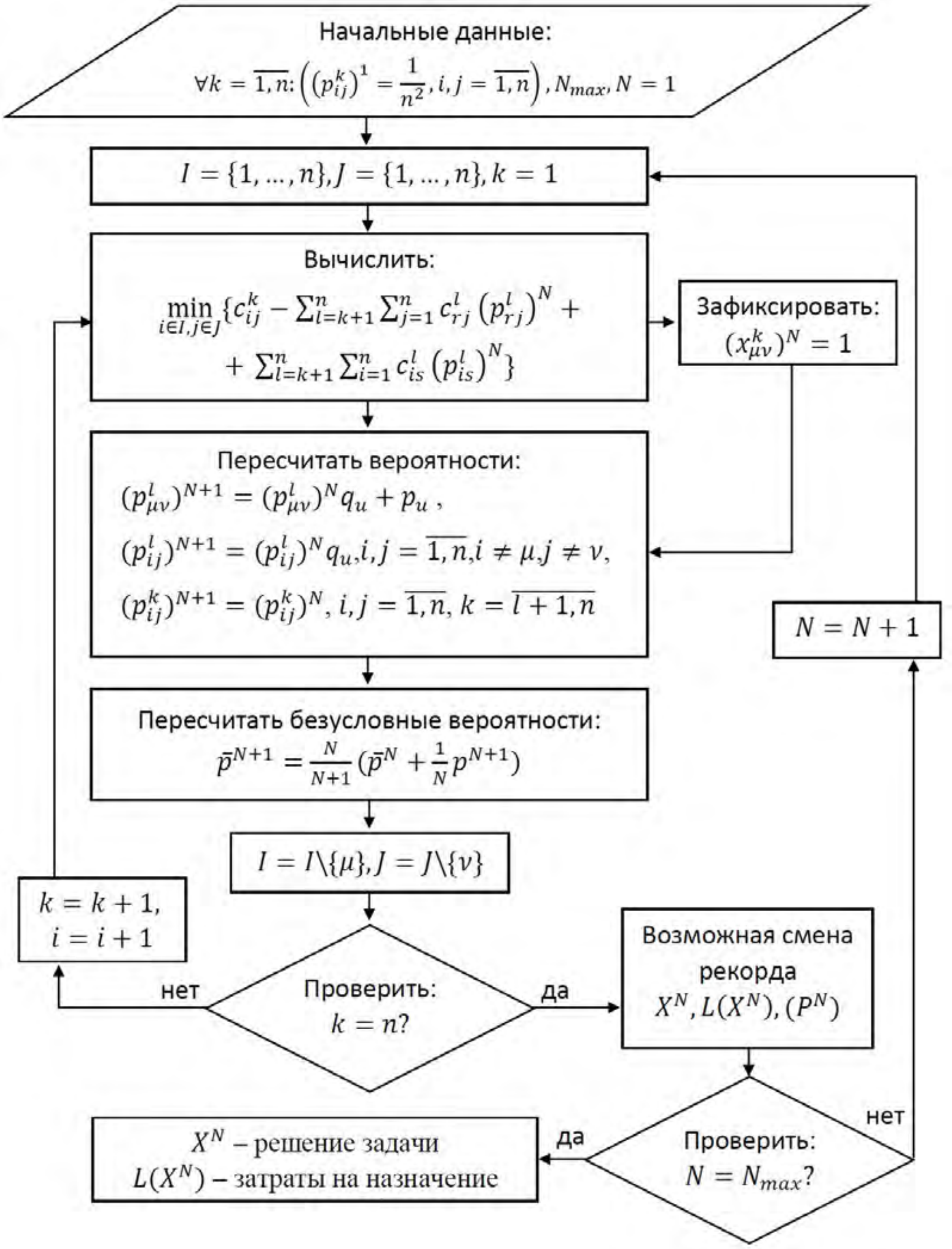

Рис. 1. Блок-схема адаптивного алгоритма решения для трёх индексной аксиальной ЗОН

Таблица 1

Сравнение значений целевой функции

Размерность матрицы	Алгоритм			
	«Жадный»	Адаптивный		
		10 итераций	50 итераций	100 итераций
10×10×10	103	56	43	40
50×50×50	145	90	75	71
100×100×100	195	142	104	94

Таблица 2

Сравнение времени выполнения (сек)

	10 итераций	50 итераций	100 итераций	200 итераций
$10\times10\times10$	0,0009 с	0,0024 с	0,006 с	0,0104 с
$50\times50\times50$	0,061 с	0,29 с	0,57 с	1,14 с
$100\times100\times100$	0,76 с	3,81 с	7,68 с	15,23 с

Рис. 2. Сравнение работы адаптивного алгоритма с фиксированием рекорда и без

результат сильно превосходит результат базового «жадного» алгоритма уже после первых десяти итераций.

Исследуем зависимость значений целевых функций от параметра p_u при разном числе итераций для матриц разной размерности.

В данном эксперименте, как и в предыдущем пункте, проводились серии испытаний (по 100 на каждый случай). Матрицы заполнялись случайными целыми числами из диапазона от 1 до 100. Затем считалось среднее значение целевых функций (округлено до целой части).

В табл. 3 приведены результаты работы программной реализации адаптивного алгоритма с разными значениями параметра p_u и разным числом итераций для матриц $10\times10\times10$. Жирным шрифтом выделены лучшие результаты значений целевой функции.

Выводы:

- При значении параметра p_u меньше 0,1 результат резко ухудшается.
- На малом числе итераций (до 20) лучшее значение целевой функции получается при значениях p_u от 0,2 до 0,3. При этом от 0,01 до 0,2 идёт резкое улучшение, а после 0,3 – плавный спад.
- При числе итераций, большем 30, чем больше значение параметра p_u, тем лучше значение целевой функции, но начиная с определённого значения p_u улучшение значения целевой функции очень незначительное (практически отсутствует), и чем больше число итераций, тем это значение параметра p_u больше.
- На первых десяти итерациях значение целевой функции сильно улучшается, от 10 до 50 итераций улучшение идёт линейно, больше 50 – улучшение незначительно.

Таблица 3

Сравнение значений целевой функции в зависимости от параметра p_u и числа итераций для матриц размерности $10\times10\times10$

$10\times10\times10$		Число итераций					
		5	10	20	30	50	100
Параметр p_u	$p_u = 0,01$	88	84	81	79	76	72
	$p_u = 0,1$	67	58	52	50	48	47
	$p_u = 0,2$	**65**	**54**	**48**	47	45	46
	$p_u = 0,3$	**65**	**54**	**48**	**46**	45	45
	$p_u = 0,4$	66	55	49	46	44	45
	$p_u = 0,5$	67	56	52	46	**43**	43
	$p_u = 0,6$	67	57	52	46	43	41
	$p_u = 0,7$	69	58	53	46	43	**40**
	$p_u = 0,8$	70	59	54	46	43	40
	$p_u = 0,9$	71	59	55	46	43	40
	$p_u = 0,99$	72	59	55	46	43	40

Таким образом, для матриц $10\times10\times10$ наиболее удачным будет применение алгоритма с параметром $p_u = 0,5$ и числом итераций $N = 50$.

На рис. 3 представлены графики функций «жадного» алгоритма и адаптивного алгоритма с разными параметрами p_u для матриц $10\times10\times10$, построенные по результатам предыдущей серии опытов.

По оси Y расположены полученные средние значения целевой функции на данной итерации, по оси X – номер итерации.

На графике видно, что при всех значениях p_u происходит значительное улучшение базового «жадного» алгоритма, хоть и по-разному.

СПИСОК ЛИТЕРАТУРЫ

1. *Корбут А. А.* Дискретное программирование / А. А. Корбут, Ю. Ю. Финкельштейн [под ред. Д.Б. Юдина]. – М. : Наука, 1969. – 368 с.

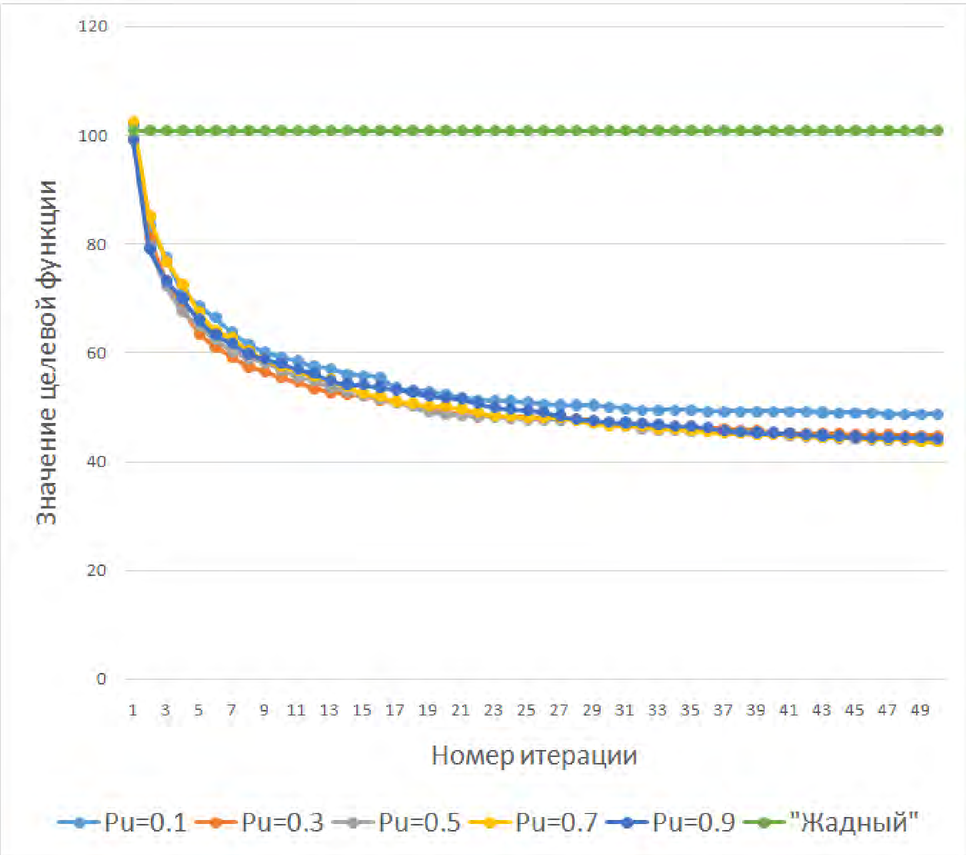

Рис 3. График работы алгоритма с разными параметрами p_u

2. *Львович Я. Е.* Конструирование адаптивных схем перебора для решения дискретных задач оптимизации / Я. Е. Львович, А. И Каплинский, Г. Д. Чернышова, О. И. Черных // В сб.: Актуальные проблемы фундаментальных наук. – М. : Изд-во МГТУ, 1991.

3. *Малюгина О. А.* Задача комплектования штатов / О. А. Малюгина, С. Н. Медведев, Г. Д. Чернышова // Системное моделирование социально-экономических процессов : труды 31-й Международной научной школы-семинара, Воронеж, 1–5 октября 2008 г. : в 3 ч. – Воронеж : ИПЦ Воронеж. гос. ун-т., 2008. – Ч. III. – С. 265–272.

4. *Медведев С. Н.* Адаптивные алгоритмы решения трёхиндексных задач о назначениях / С. Н. Медведев // Современные методы прикладной математики, теории управления и компьютерных технологий: сб. тр. VI Междунар. конф. «ПМТУКТ-2013», Воронеж, 10–16 сентября 2013 г. – Воронеж: ИПЦ Воронеж. гос. ун-т, 2013. – С. 153–156.

Трегубов А. Г. – магистр кафедры вычислительной математики и прикладных информационных технологий факультета прикладной математики, информатики и механики Воронежского государственного университета.
E-mail: n.a.p.o.l.e.o.n@gmail.ru.

Медведев С. Н. – кандидат физико-математических наук, преподаватель кафедры вычислительной математики и прикладных информационных технологий факультета прикладной математики, информатики и механики Воронежского государственного университета.
E-mail: s_n_medvedev@mail.ru.

Tregubov A. G. – master of the Department of Computing Mathematicsand Applied Information Technology, Applied Mathematics, Informatics and Mechanics Faculty, Voronezh State University.
E-mail: n.a.p.o.l.e.o.n@gmail.ru

Medvedev S. N. – Candidate of Physico-Mathematical Sciences, Lecturer of the Department of Computing Mathematics and Applied Information Technology, Applied Mathematics, Informatics and Mechanics Faculty, Voronezh State University.
E-mail: s_n_medvedev@mail.ru.