

ЭВОЛЮЦИОННО-ФРАГМЕНТАРНАЯ МОДЕЛЬ ЗАДАЧИ ТРАССИРОВКИ

Аннотация. Рассматривается один из вариантов задачи трассировки на плоской целочисленной решетке. Показано, что эта задача может быть представлена как задача поиска слов с определенными свойствами над конечным алфавитом. В свою очередь задача поиска оптимальных слов может рассматриваться как задача с фрагментарной структурой. Получена комбинаторная оценка множества допустимых слов, установлена нижняя оценка плотности в задаче поиска оптимальной трассировки с критерием плотности. Построена эволюционно-фрагментарная модель задачи трассировки, для малых размеров получены оптимальные и близкие к оптимальным решения этой задачи

Ключевые слова: задача трассировки, плотность трассировки, комбинаторная оптимизация, фрагментарная структура, эволюционная модель.

ВВЕДЕНИЕ

Одна из основных задач плоской однослойной трассировки заключается в построении соединений между заданными контактами таким образом, чтобы проводники не пересекались между собой и при этом выполнялись определенные конструктивные ограничения. Ограничения могут быть на равномерность распределения проводников, суммарную длину соединений, длину самой длинной связи, число слоев, число межслойных переходов и т.д.

Большинство задач трассировки являются сложными в вычислительном смысле. Для таких задач оправдано применение разного рода метаэвристик, в частности эволюционных моделей. Эволюционные модели и генетические алгоритмы часто используются для поиска приближенных оптимальных решений различных постановок задач трассировки [1–7].

В настоящей статье рассматривается один из вариантов задачи трассировки. Показано, что в этом случае задача имеет фрагментарную структуру. Установлены нижние оценки плотности трассировки. Предложена универсальная эволюционно-фрагментарная модель задачи, с помощью которой получен ряд приближенных оптимальных решений задачи трассировки с критерием плотности.

ПОСТАНОВКА ЗАДАЧИ

Рассматривается задача трассировки с конструктивным ограничением на плотность распределения проводников на плате с матрично расположенными контактами. Как показано в [8], эта задача может быть сведена к задаче трассировки на квадрате, который представляет собой верхнюю левую четверть платы.

Пусть задана квадратная целочисленная решетка на плоскости размера $n \times n$, где n — количество узлов на стороне квадрата. На рис. 1 приведен пример такой решетки с $n = 5$. Будем называть единичные отрезки, соединяющие узлы решетки, звеньями, единичные квадраты решетки — клетками. Граница ре-

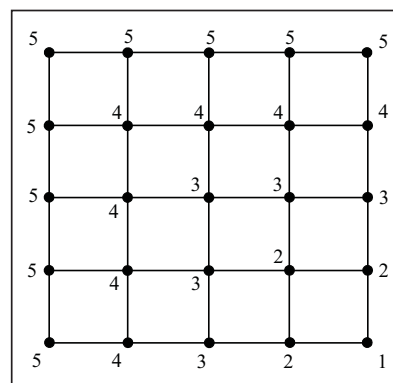


Рис. 1. Решетка размера 5×5

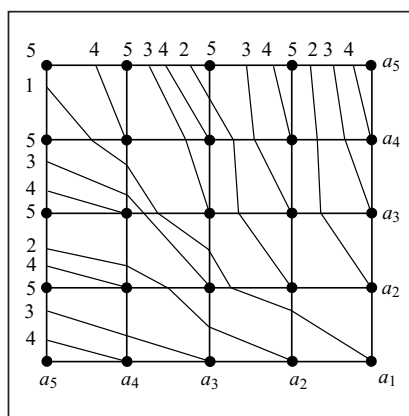


Рис. 2. Допустимая трассировка

сетки — это узлы и звенья левой и верхней сторон квадрата. Все узлы, находящиеся на границе решетки размера $n \times n$, назовем узлами n -го уровня. Уровни остальных узлов решетки определим рекурсивно по следующему правилу. Удаляя границу решетки размера $(k+1) \times (k+1)$, звенья, инцидентные узлам границы, и внутренности прилегающих к границе клеток, получаем решетку размера $k \times k$, у которой узлы границы принадлежат k -му уровню (см. рис. 1). Клеткам решетки, находящимся между узлами уровня k и $k+1$, припишем уровень k . Таким образом, существует ровно $2k-1$ клеток k -го уровня, $k=1, n-1$.

Трассировка — набор линий, соединяющих узлы с точками на границе решетки. Каждая такая линия определяется началом и последовательностью звеньев решетки, которые она пересекает. Линии, у которых начала и последовательности прохождения звеньев решетки одинаковы, будем считать эквивалентными. Соответственно будем считать эквивалентными трассировки, состоящие из попарно эквивалентных линий.

Определение 1. Допустимой трассировкой на решетке размера $n \times n$ будем называть трассировку, при которой:

- 1) линии трассировки не пересекаются;
- 2) при движении вдоль линии от первого уровня к n -му номер уровня проходимых клеток не понижается;
- 3) линия трассировки пересекает каждое звено решетки не больше одного раза.

На рис. 2 приведен пример допустимой трассировки на решетке размера 5×5 .

Пусть задана допустимая трассировка на целочисленной решетке размером $n \times n$. Тогда очевидно, что эта трассировка порождает допустимые трассировки и на любой решетке размера $k \times k$, состоящей из клеток уровня от 1 до k , $k=1, n-1$.

С точностью до эквивалентности каждая допустимая трассировка на решетке $n \times n$ может быть описана некоторым словом A в алфавите a_1, a_2, \dots, a_n . Буквой a_i будем кодировать выход на границу решетки той линии трассировки, которая начинается в узле i -го уровня [8]. Слово, кодирующее трассировку, формируется из букв, кодирующих концы линий трассировки, по часовой стрелке, начиная от нижнего левого узла решетки до правого верхнего узла. Например, слово A_5 , определяющее трассировку, приведенную на рис. 2, в алфавите a_1, a_2, a_3, a_4, a_5 имеет вид $A_5 = a_5 a_4 a_3 a_5 a_4 a_2 a_5 a_4 a_3 a_5 a_1 a_5 a_4 a_5 a_3 a_4 a_2 a_5 a_3 a_4 a_5 a_2 a_3 a_4 a_5$ (концы линий трассировки на рисунке обозначены только цифрами-индексами букв алфавита).

Пусть A — произвольное слово в алфавите a_1, a_2, \dots, a_n . Спектром этого слова назовем последовательность слов

$$A_n = A, A_{n-1} = A \setminus \{a_n\}, A_{n-2} = A_{n-1} \setminus \{a_{n-1}\}, \dots, A_0 = A_1 \setminus \{a_1\} = \emptyset.$$

Определение 2. Слово A будем называть правильным, если:

- 1) любое слово A_k из его спектра начинается и заканчивается буквой a_k , $k=1, n$;
- 2) буква a_k входит в слово A ровно $2k-1$ раз.

Очевидно, что если слово A правильное, то каждое слово A_k из его спектра также правильное в алфавите a_1, a_2, \dots, a_k .

Теорема 1. С точностью до эквивалентности каждое правильное слово определяет единственную допустимую трассировку в решетке. Наоборот, каждая допустимая трассировка описывается единственным правильным словом.

Доказательство. Докажем прямое утверждение по индукции. Рассмотрим правильное слово A_n , кодирующее некоторую трассировку на решетке $n \times n$. Поскольку из каждого узла выходит ровно одна линия, то количество линий, выходящих из узлов k -го уровня, равно $2k-1$, $k = \overline{1, n}$, т.е. количеству букв a_k в слове A_n . Правильное слово $A_1 = a_1$ описывает тривиальную допустимую трассировку на решетке 1×1 . Допустим, правильное слово A_{j-1} ($j = \overline{2, n}$) кодирует допустимую трассировку на решетке размера $(j-1) \times (j-1)$. Построим слово A_j следующим образом: возьмем $(2j-1)$ букв a_j ; по краям слова A_{j-1} припишем по букве a_j ; оставшиеся $(2j-3)$ букв расположим внутри слова A_{j-1} произвольным образом. Слово A_j правильное по построению. При этом крайние буквы a_j кодируют крайние узлы j -го уровня новой решетки размера $j \times j$, а внутренние буквы a_j определяют звенья j -го уровня, через которые проходят линии с $(j-1)$ -го уровня и добавляют в этих позициях новые линии j -го уровня. Поскольку исходная последовательность букв слова A_{j-1} не менялась, то линии $(j-1)$ -го уровня на выходе из j -го уровня по-прежнему не пересекаются, при этом к ним добавляются линии j -го уровня. Так как звено однозначно задается узлами, которые оно соединяет, то слово A_j определяет последовательность звеньев для каждой линии. Следовательно, правильное слово A_j однозначно кодирует допустимую трассировку на решетке размера $j \times j$.

Докажем обратное утверждение. Пусть имеем решетку $n \times n$ с допустимой трассировкой и слово A_n , кодирующее эту трассировку. Очевидно, что слово A_n содержит ровно $(2i-1)$ букв a_i , $i = \overline{1, n}$, и начинается и заканчивается буквой a_n , так как при допустимой трассировке линии не выходят через нижнюю и правую стороны решетки. Удаляя все вхождения буквы a_n в слово A_n , убираем из рассмотрения все узлы n -го уровня, что эквивалентно построению решетки $(n-1) \times (n-1)$, на которой, как отмечалось выше, трассировка также допустима. Следовательно, полученное слово A_{n-1} начинается и заканчивается буквой a_{n-1} . Применяя последовательно эту процедуру, убеждаемся, что все слова A_i спектра начинаются и заканчиваются буквой a_i . Таким образом, слово A_n правильное. Поскольку узлы, т.е. буквы слова, однозначно определяют звенья, то слово A_n единственное для данной трассировки.

ПЕРЕЧИСЛЕНИЕ ДОПУСТИМЫХ ТРАССИРОВОК

Пронумеруем выходы линий допустимой трассировки из клеток $(k-1)$ -го уровня в порядке их физического размещения на решетке от 1 до m . Эту нумерацию будем называть кортежем. Разделением кортежа $C = \langle 1, 2, \dots, m \rangle$ на s субкортежей назовем последовательность $\langle \langle 1, 2, \dots, j_1 \rangle, \langle j_1 + 1, j_1 + 2, \dots, j_2 \rangle, \dots, \langle j_{s-1} + 1, j_{s-1} + 2, \dots, m \rangle \rangle$, где $s = \overline{1, m}$,

$j_0 = 0, j_i = j_{i-1} + l_i, i = \overline{1, s}, l_i$ — длина i -го субкортежа, при этом $\sum_{i=1}^s l_i = m$.

Все возможные разделения k_m^s кортежа C длины m на s субкортежей можно получить, расставляя $s-1$ ($s = \overline{1, m}$) разделителей C_{m-1}^{s-1} способами, т.е.

$k_m^s = C_{m-1}^{s-1} = \frac{(m-1)!}{(s-1)!(m-s)!}$. При этом количество всех возможных разделений кортежа C длины m равно

$$K_m = \sum_{s=1}^m k_m^s = \sum_{s=1}^m C_{m-1}^{s-1} = \left[\begin{matrix} s-1=l \\ m-1=p \end{matrix} \right] = \sum_{l=0}^p C_p^l = 2^p = 2^{m-1}.$$

С другой стороны, число k_m^s можно рассматривать как перестановку s ненулевых длин l_i субкортежей, при этом не различаем одинаковые по величине длины. Тогда k_m^s — сумма слагаемых вида

$$k_{l_1, l_2, \dots, l_s} = \frac{s!}{t_1! t_2! \dots t_m!},$$

где t_j — количество субкортежей длины j , $0 \leq t_j \leq m$, $j = \overline{1, m}$, среди которых s ненулевой длины, $R_s = \left\{ (t_1, t_2, \dots, t_m) : t_1 + 2t_2 + \dots + mt_m = \sum_{j=1}^m jt_j = m, 0 \leq t_j \leq m, j = \overline{1, m} \right\}$ — множество всех возможных разбиений числа m на s слагаемых, при этом $s = \sum_{j=1}^m t_j$. Пусть r — элемент множества R_s , тогда $k_m^s = s! \sum_{r \in R_s} \frac{1}{\prod_{j=1}^m t_j!}$. Таким образом, получаем тождество

$$s! \sum_{r \in R_s} \frac{1}{\prod_{j=1}^m t_j!} = C_{m-1}^{s-1}. \quad (1)$$

Пусть на выходе из клеток k -го уровня платы имеется $m_k = k^2$ линий трассировки, а на следующем $(k+1)$ -м уровне добавляется $p_k = 2k+1$ новых линий. Тогда для распределения по (p_k-1) звеньям s непересекающихся групп линий трассировки, соответствующим s субкортежам, имеется $k_{m_k}^s * C_{p_k-1}^s$ способов. Количество способов N_k провести соединения между k - и $(k+1)$ -м уровнями равно

$$N_k = \sum_{s=1}^{p_k-1} k_{m_k}^s C_{p_k-1}^s.$$

Следовательно, количество правильных слов N , которое определяет с точностью до эквивалентности количество всех допустимых трассировок, —

$$N = \prod_{k=1}^{n-1} N_k = \prod_{k=1}^{n-1} \sum_{s=1}^{p_k-1} k_{m_k}^s C_{p_k-1}^s = \prod_{k=1}^{n-1} \sum_{s=1}^{p_k-1} C_{m_k-1}^{s-1} C_{p_k-1}^s, \quad n=2, 3, \dots \quad (2)$$

С учетом соотношений $p_k = 2k+1$, $m_k = k^2$, $k = \overline{1, n-1}$, формула (2) принимает следующий вид:

$$N = \prod_{k=1}^{n-1} \sum_{s=1}^{2k} C_{k^2-1}^{s-1} C_{2k}^s, \quad n=2, 3, \dots \quad (3)$$

ПЛОТНОСТЬ ТРАССИРОВКИ. НИЖНЯЯ ОЦЕНКА ПЛОТНОСТИ

Для каждого $i=1, 2, \dots, n-1$ перенумеруем клетки i -го уровня: $1, 2, \dots, 2i-1$. Рассмотрим допустимую трассировку A и обозначим P_{ij} количество линий трассировки, проходящих через j -ю клетку i -го уровня. Это число будем называть плотностью клетки в трассировке A .

Определение 3. Плотностью трассировки A будем называть число P_A^n , равное максимальной из плотностей клеток решетки в трассировке A .

Задача об оптимальной трассировке состоит в отыскании такой допустимой трассировки A на заданной решетке размера $n \times n$, для которой плотность P_A^n минимальна, т.е. задача состоит в поиске минимума целевой функции

$$P_A^n = \max P_{ij} \rightarrow \min, \quad i = \overline{1, n-1}, \quad j = \overline{1, 2i-1},$$

на множестве допустимых трассировок, или, что, то же самое, на множестве правильных слов в алфавите a_1, a_2, \dots, a_n .

В решетке размером $n \times n$ ($n > 1$) имеется ровно $2n-3$ клеток уровня $n-1$. Количество линий трассировки, проходящих через клетки уровня $n-1$, равно $(n-1)^2$. Таким образом, плотность любой допустимой трассировки на решетке размера $n \times n$ ограничена снизу числом

$$M_n = \left\lceil \frac{(n-1)^2}{2n-3} \right\rceil = \left\lceil \frac{n}{2} - \frac{1}{4} + \frac{1}{8n-12} \right\rceil. \quad (4)$$

Здесь $\lceil x \rceil$ — округление числа x до целого в большую сторону. Следовательно, для любой трассировки A на решетке размера $n \times n$ имеет место неравенство $P_A^n \geq M_n$. Учитывая, что $\frac{1}{4} - \frac{1}{8n-12} \leq \frac{1}{2}$, из формулы (4) получаем, что

$$M_n = \left\lceil \frac{n+1}{2} \right\rceil, \quad (5)$$

где $[x]$ — целая часть числа x . При нечетных значениях $n=2k+1$ нижняя оценка принимает вид $P_A^{2k+1} \geq k+1$, $k=1, 2, \dots$. Для четных значений $n=2k$ соответственно имеем $P_A^{2k} \geq k$, $k=2, 3, \dots$. Вопрос о достижимости этой оценки остается открытым.

ЭВОЛЮЦИОННО-ФРАГМЕНТАРНАЯ МОДЕЛЬ ЗАДАЧИ

Покажем, что задача трассировки в рассматриваемой постановке допускает фрагментарную структуру [9]. В качестве допустимых фрагментов будем брать те слова в алфавите a_1, a_2, \dots, a_n , которые являются префиксными для правильных слов. Пусть слово просматривается слева направо. Тогда критерием допустимости слова A являются следующие условия:

- 1) для любого $k=1, 2, \dots, n$ количество букв a_k в слове A не превосходит числа $2k-1$;
- 2) перед первым по порядку появлением буквы a_k в слове A не встречаются буквы с номерами, меньшими k ;
- 3) после $(2k-1)$ -го вхождения буквы a_k в слове A не встречаются буквы с номерами, меньшими k ;
- 4) между 1- и $(2k-1)$ -м вхождениями буквы a_k в слове содержится ровно $2i-1$ экземпляров каждой буквы a_i , $i = \overline{1, k-1}$.

Максимальный допустимый фрагмент, т.е. правильное слово A , содержит ровно n^2 букв.

Множество элементарных фрагментов состоит из букв алфавита a_1, a_2, \dots, a_n . При этом каждая буква a_k , $k=1, n$, представлена в этом множестве ровно $2k-1$ раз. Следовательно, мощность множества элементарных фрагментов

равна $1+3+5+\dots+2n-1 = n^2$. По каждой перестановке элементарных фрагментов можно построить правильное слово следующим «жадным» алгоритмом: составляется список элементарных фрагментов (букв) в порядке, заданном перестановкой. На первом шаге допустимое слово пустое. На очередном шаге к слову добавляется первая по порядку буква списка так, чтобы выполнялись условия допустимости 1)–4). Добавленная буква удаляется из списка. Трудоемкость этого алгоритма не превосходит величины $O(n^4)$. Наличие фрагментарной структуры позволяет перейти к эволюционно-фрагментарной модели [10]. В качестве базового множества эволюционной модели предлагается множество перестановок S_{n^2} из n^2 элементов. Каждая перестановка соответствует определенному порядку следования фрагментов рассматриваемой модели и однозначно определяет правильное слово, т.е. определенную трассировку. Оператором кроссовера выбирается бинарный оператор $K : S_{n^2} \times S_{n^2} \rightarrow S_{n^2}$, сохраняющий порядок следования элементов в перестановках [10]. В качестве оператора мутации предлагается оператор, случайным образом меняющий два наугад выбранных элемента в перестановке. Критерием задачи является плотность трассировки.

РЕЗУЛЬТАТЫ РАБОТЫ ЭВОЛЮЦИОННО-ФРАГМЕНТАРНОГО АЛГОРИТМА ДЛЯ РЕШЕТОК МАЛЫХ РАЗМЕРНОСТЕЙ

Применяя эволюционно-фрагментарный алгоритм, удается получить приближенные оптимальные решения задачи для решетки любого размера. Приведем некоторые решения для малых размеров, полученные в результате работы алгоритма:

$$\begin{aligned}
 P_A^n &= 3, M_n = 3 \text{ при } n = 5; \\
 P_A^n &= 4, M_n = 3 \text{ при } n = 6; \\
 P_A^n &= 4, M_n = 4 \text{ при } n = 7; \\
 P_A^n &= 5, M_n = 4 \text{ при } n = 8; \\
 P_A^n &= 5, M_n = 5 \text{ при } n = 9; \\
 P_A^n &= 8, M_n = 5 \text{ при } n = 10; \\
 P_A^n &= 8, M_n = 6 \text{ при } n = 11.
 \end{aligned}$$

Отметим, что для решеток 5×5 , 7×7 и 9×9 полученные трассировки оптимальные. Решения для решеток 9×9 и 11×11 приведены на рис. 3 и рис. 4.

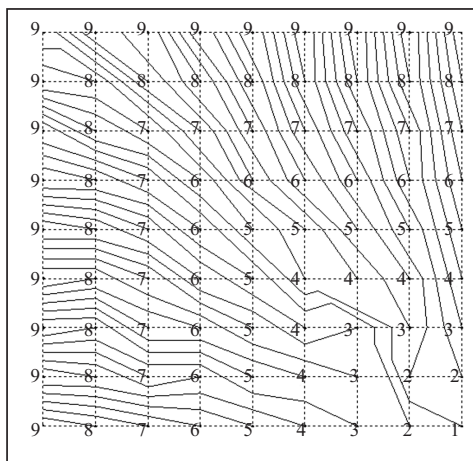


Рис. 3. Трассировка с плотностью 5 на решетке 9×9

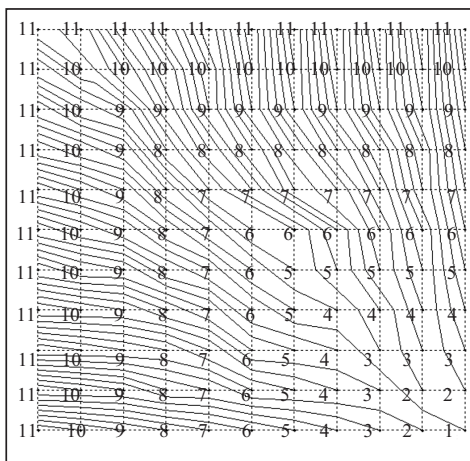


Рис. 4. Трассировка с плотностью 7 на решетке 11×11

ЗАКЛЮЧЕНИЕ

В настоящей работе показано, что задача трассировки определенного типа на целочисленной решетке может быть сведена к комбинаторной задаче поиска правильных слов в языке с конечным алфавитом. Установлено точное число (3) трассировок определенного типа, откуда следует, что переборный алгоритм поиска оптимальных решений задачи трассировки имеет экспоненциальную трудоемкость, при этом было получено комбинаторное тождество (1). Доказано, что задача трассировки может рассматриваться как задача на фрагментарной структуре, в частности, возможна сюръекция множества перестановок конечного набора элементов на множество допустимых решений задачи. Таким образом, задача поиска оптимальной трассировки может быть сведена к комбинаторной задаче поиска оптимальной перестановки.

Наличие фрагментарной структуры позволяет для задачи поиска приближенной оптимальной трассировки использовать эволюционно-фрагментарную модель. С помощью этой модели получен ряд приближенных решений. Установлена нижняя оценка (5) плотности допустимых трассировок. Это позволяет доказать, что некоторые из построенных эволюционным алгоритмом решений оптимальные.

Предлагаемый метод поиска приближенных решений можно легко перенести и на другие типы задач трассировки с различными конструктивными ограничениями.

СПИСОК ЛИТЕРАТУРЫ

1. Kernytskyu A., Kryvyu R., Tkatchenko S. VLSI Topology synthesis using the method of parallel genetic algorithms // III International Conference of Young Scientists MEMSTECH'2007, May, 23–26, 2007. — Lviv: Polyana, 2007. — P. 151–152.
2. Лебедев Б.К., Воронин Е.И. Генетический алгоритм распределения соединений по слоям при многослойной глобальной трассировке сбис // Изв. ЮФУ. Техн. науки. Тем. вып.: Интеллектуальные САПР. — 2012. — № 7 (132). — С. 14–21.
3. Гладков Л.А., Щеглов С.Н. Синтез архитектуры гибридных интеллектуальных систем на основе принципов самоорганизации и саморегулирования // Труды международных конференций IEEE AIS'02 и ИСАПР 2002. — М.: Физматлит, 2002. — С. 173–80.
4. Корпулјов D., Sviridova T., Tkachenko S. Using of genetic algorithms in design of Hybrid Integrated Circuits // Proc. of IXth International Conference on the Experience of Designing and Application of CAD Systems in Microelectronics, February, 20–24, 2007. — Lviv: Polyana, 2007. — P. 302.
5. Li enig J. Physical design of VLSI circuits // Evolutionary algorithms in engineering applications / D. Dasgupta, Z. Michalewicz (Eds). — Berlin: Springer, 1997. — P. 277–293.
6. Chi Keong Goh, Wei Ling, Yong Han Chew, Kay Chen Tan. A multi-objective evolutionary algorithm for channel routing problems / K.P. Dahal, P.I. Cowling (Eds) // Evolutionary scheduling — Berlin: Springer, 2007. — P. 404–436.
7. Schnecke V. Genetic design of VLSI layouts / A.M.S. Zalzal, P.J. Flemming (Eds) // Genetic algorithms in engineering systems. — London: The Institution of Electrical Engineers, 1997. — P. 229–253.
8. Трассировка подключающей пластины многозондового устройства контроля BGA-компонентов / И.Ш. Невлюдов, С.В. Курапов и др. // VI Междунар. науч. конф. «Функциональная база нанозлектроники». — Харьков: ХНУРЭ, 2013. — С. 129–32.
9. Козин И.В., Полюга С.И. О свойствах фрагментарных структур // Вісник Запорізького нац. ун-ту. Математичне моделювання і прикладна механіка. — 2012. — № 1. — С. 99–106.
10. Козин И.В., Полюга С.И. Использование ЭВФ-алгоритмов для решения задачи прямоугольного раскрыя // Питання прикладної математики і математичного моделювання. — Дніпропетровськ: Вид-во Дніпропетр. нац. ун-ту імені Олеся Гончара, 2009. — С. 199–208.

Поступила 10.01.2014