

Д.А. Архипова, А.Р. Юсупова
(г. Казань, Казанский национальный исследовательский технический
университет им. А.Н.Туполева-КАИ)

**СРАВНЕНИЕ ПОСЛЕДОВАТЕЛЬНОГО АЛГОРИТМА
С АЛГОРИТМОМ СЛЕПОГО ПОИСКА РЕШЕНИЯ ЗАДАЧИ
РАЗМЕЩЕНИЯ ЭЛЕМЕНТОВ НА ПЕЧАТНОЙ ПЛАТЕ**

COMPARISON OF A SEQUENTIAL ALGORITHM WITH A BLIND SEARCH
ALGORITHM FOR SOLVING THE PROBLEM OF PLACING ELEMENTS
ON A PRINTED CIRCUIT BOARD

Представлены исследования эффективности работы последовательного и слепого алгоритма в задаче размещения конструктивных элементов. Основной идеей последовательного алгоритма является получение оптимального размещения при расположении максимально связанных конструктивных элементов в соседних позициях. Главной идеей метода слепого поиска является случайный выбор позиции из числа не занятых позиций, на которой закрепляется подлежащий размещению элемент, начиная с первого.

The paper presents studies of the effectiveness of sequential and blind algorithms in the problem of placing structural elements. The main idea of the sequential algorithm is to obtain the optimal placement when placing the most connected structural elements in neighboring positions. The main idea of the blind search method is to randomly select a position from among the unoccupied positions, where the element to be placed is fixed, starting from the first one.

Ключевые слова: задача размещения, метод слепого поиска, последовательный алгоритм, эффективность.

Keywords: placement problem, blind search method, sequential algorithm, efficiency.

Основная задача размещения формулируется следующим образом: минимизировать суммарную взвешенную длину связей. Поместить каждый элемент графа в единственную позицию таким образом, чтобы оптимизировать целевую функцию.

Дискретные алгоритмы размещения используют дискретные методы оптимизации. Если имеется n установочных мест, в которых нужно разместить k элементов, то полный перебор всех возможных вариантов размещения k элементов в n установочных местах составит $C_n^k n!$. При большом числе n и k эта задача практически может оказаться неосуществимой, поэтому в этом методе применяют подходы, которые позволяют сделать направленных перебор возможных вариантов размещения.^[1]

Целью представленной работы является исследование эффективности применения последовательного алгоритма и метода слепого поиска к задаче размещения элементов на печатной плате. Проведем эксперимент и сравним по скорости решения и значению целевой функции последовательного алгоритм и метод слепого поиска.

Первым рассмотрим последовательный алгоритм размещения.

Основной идеей этого алгоритма является идея упорядочения элементов по определенным признакам. Последовательные алгоритмы основаны на допущении, что для получения оптимального размещения необходимо в соседних позициях располагать элементы, максимально связанные друг с другом. Сущность этих алгоритмов состоит в последовательном закреплении заданного набора конструктивных элементов на коммутационной плате относительно ранее установленных. Выбор очередного элемента для размещения осуществляется путем вычисления следующей характеристики для каждого неразмещенного элемента $e_i \in \overline{E}_k$.

$$a_j = \max_{e_i \in \overline{E}_k} \sum c_{ij}$$

Решение задачи размещения с помощью последовательного алгоритма в отличие от метода слепого поиска, где алгоритм из множества случайного перебора находит наиболее оптимальное значение целевой функции, проходит в один этап перебора всех элементов.

Пошаговая инструкция последовательного алгоритма размещения конструктивных элементов:^[2]

1. Вычислить матрицу смежности $C = \parallel c_{ij} \parallel$.
2. Для каждой строки матрицы смежности определить локальную степень каждой вершины $\rho(x_i) = \sum_{j=1}^n c_{ij}$
3. Выбрать строку в матрице смежности, соответствующую вершине x_s с наибольшей локальной степенью. Номер выбранной строки соответствует номеру конструктивного элемента.
4. Поместить выбранный конструктивный элемент в первую позицию на монтажной плате. Исключить первую позицию из подмножества \overline{S}_k , а вершину x_s из подмножества \overline{E}_k .
5. Для каждого неразмещенного элемента $e_j \in \overline{E}_k$ вычислить $a_j = \max_{e_i \in \overline{E}_k} \sum c_{ij}$
6. Поместить конструктивный элемент с наибольшим значением a_j в очередную позицию из подмножества \overline{S}_k . Исключить номер очередной занятой позиции из подмножества \overline{S}_k , а вершины x_s из подмножества \overline{E}_k .
7. Если подмножество \overline{S}_k не пусто, то переходим к пункту 5.
8. Конец работы алгоритма.

В методе слепого поиска случайным образом выбирается позиция из числа незанятых позиций и на ней закрепляется подлежащий размещению элемент, начиная с первого. Эта же процедура выполняется для следующих элементов до тех пор, пока все элементы не будут размещены. По результатам размещения вычисляется значение целевой функции F . Указанную процедуру повторяем вновь и получаем новую целевую функцию. Сравнивая полученные значения целевой функции, запоминаем размещение, для которого это значение будет наименьшим. Наша программа проходит 20 итераций.

Разберем метод слепого поиска по шагам.^[2]

Замечание. Обозначим n – число размещаемых конструктивных элементов;

1. Пронумеровать все возможные позиции для размещения конструктивных элементов в порядке от 1 до m

2. С использованием датчика случайных чисел с равномерным распределением генерировать n случайных чисел (по числу размещаемых элементов) в интервале (l, m_1) .

3. Разместить конструктивные элементы в позициях, номера которых выбраны случайно в п. 2. Вычислить значение целевой функции F .

4. Если это первая итерация, то полученное значение F запоминается как $F_{\text{опт}}$. При выполнении всех последующих итераций полученное значение F запоминается как $F_{\text{опт}}$ только в том случае, если оно оказывается меньше полученного на предыдущей итерации.

5. Запомнить полученное размещение конструктивных элементов.

6. Проверяется условие, все ли элементы размещены. Если оно не выполняется, то осуществляется переход к п. 2, в противном случае – к п. 7

7. Конец алгоритма.

Для исследования эффективности двух алгоритмов в работе были взяты две программные реализации этих алгоритмов на примере решения задач о размещении конструктивных элементов на печатной плате. Для исследования эффективности работы было сгенерировано 10 различных графов размерности от 4 до 70 элементов. Некоторые данные результатов представлены в таблице 1.

Таблица 1. Результаты

Количество элементов	Последовательный алгоритм		Алгоритм слепого поиска	
	Время выполнения алгоритма	Значение целевой функции	Время выполнения алгоритма	Значение целевой функции
4	0.04	10	0.13	10
4	0.53	26	0.60	26
9	0.13	28	0.18	32

Количество элементов	Последовательный алгоритм		Алгоритм слепого поиска	
	Время выполнения алгоритма	Значение целевой функции	Время выполнения алгоритма	Значение целевой функции
9	0.18	138	0.34	142
16	0.13	94	0.22	116
16	0.26	376	0.36	388
25	0.14	210	0.31	216
25	0.75	692	0.83	700
36	0.39	480	0.52	484
36	0.61	1214	0.65	1220
49	0.132	746	0.66	762
64	0.190	1032	0.91	1080

Выводы

В данной работе было проведено исследование эффективности работы двух алгоритмов. В качестве параметров эффективности выбрали значения целевой функции и время работы алгоритма. Мы провели исследование для разных графов с одинаковыми вершинами. Из полученных результатов видно, что целевая функция последовательного алгоритма размещения составляет 10% от величины целевой функции слепого алгоритма размещения, что свидетельствует, что последовательный алгоритм позволяет получить более точные значения. По времени выполнения последовательный алгоритм работает более быстро, но если количество соединения элементов друг с другом увеличивается, то время выполнения алгоритма значительно увеличивается. Метод слепого поиска дает точное значение целевой функции только при маленьком количестве элементов.

Список литературы

1. *Деньдобренко, Б.Н.* Автоматизация конструирования РЭС/ Б.Н. Деньдобренко, А.С. Малика. – Москва: Высш. школа, 1980. – 366 с.
2. *Криптофидес, Н.* Теория графов. Алгоритмический подход/ Н. Криптофидес. – М.: Мир, 1978. - 432 с.

Материал поступил в редколлегию 02.10.20.