

УДК 518.5

А.Д. Закревский

ЭНЕРГОСБЕРЕГАЮЩЕЕ КОДИРОВАНИЕ СОСТОЯНИЙ КОНЕЧНОГО АВТОМАТА. МЕТОД КВАДРАТОВ

Рассматривается задача отображения произвольного неориентированного графа в булево пространство, имеющая важные приложения в энергосберегающей реализации конечных автоматов, при которой минимизируется переключательная активность схемы. В результате ее решения максимальное число ребер графа должно отобразиться на некоторые ребра гиперкуба, представляющего структуру булева пространства. Для решения данной задачи предлагается визуальный метод, использующий матрицу смежности и карту Карно и заключающийся в построении последовательности конфигураций из ребер и квадратов, образующих фрагменты гиперкуба.

Введение

Рассмотрим следующую комбинаторную задачу. Задан произвольный неориентированный граф $G = (V, E)$ с n вершинами, составляющими множество V , и m ребрами, образующими множество E . Требуется разместить вершины графа на вершинах k -мерного булева куба ($k = \lceil \log_2 n \rceil$) так, чтобы максимальное число ребер графа совместилось при этом с ребрами куба (будем говорить, что эти ребра размещаются на кубе). Назовем эту операцию оптимальным размещением графа в булевом k -мерном пространстве $M = \{1, 0\}^k$ (рис. 1). В данном случае не удалось разместить лишь три ребра: 4-6, 2-6 и 3-5.

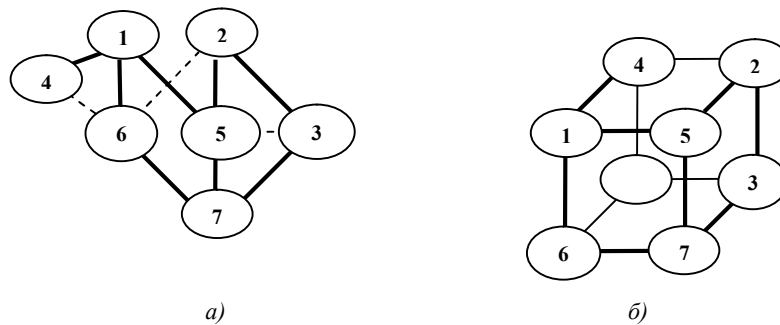


Рис. 1. Размещение графа в булевом пространстве: а) исходный граф; б) результат его размещения в трехмерном булевом пространстве

Рассматриваемую задачу можно интерпретировать как кодирование состояний конечного автомата булевыми k -векторами, имеющее целью минимизацию числа переключений двоичных элементов памяти при схемной реализации автомата. Переходы между состояниями автомата реализуются путем переключения некоторых элементов памяти (смены их состояний). На это тратится некоторая энергия, что сопровождается нежелательным выделением тепла в схеме, ограничивающим возможности микроминиатюризации [1]. Поэтому чем меньше таких переключений будет происходить в схеме, тем лучше. Возникает задача снижения *переключательной активности*, определяемой как частота переключений элементов памяти [2].

Другая полезная интерпретация данной задачи связана с асинхронной реализацией конечного автомата, при которой устраняются состязания между элементами памяти. В автомат вводится минимальное число дополнительных состояний, после чего при любом переходе между состояниями переключается лишь один элемент памяти [3].

В данных интерпретациях граф G рассматривается как граф смежности состояний автомата (два состояния *смежны*, если они связаны некоторым переходом), а получаемые при кодировании векторы (координаты соответствующих вершин куба) задают наборы состояний элементов памяти. Требуется, чтобы в максимальном числе пар смежных состояний их коды

различались ровно в одном разряде – в этом случае соответствующее ребро графа проектируется на некоторое ребро куба.

Граф, для которого существует идеальное решение, где коды любых смежных состояний различаются ровно в одном разряде, назовем *булевым* [3]. Очевидны следующие простые условия, которым должен удовлетворять булев граф:

- граф должен быть *бихроматичным* (не содержащим циклов нечетной длины);
- степени вершин не должны превышать k .

Эта пара условий необходима, но не достаточна, в чем можно убедиться на следующем примере графа, не являющегося булевым (рис. 2).

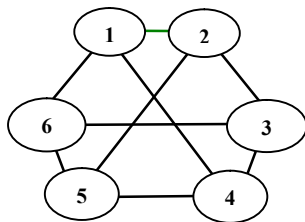


Рис. 2. Пример небулева графа

В рассмотренном выше примере (см. рис. 1) задача решается довольно легко, но в общем случае она весьма сложна, так как принадлежит классу комбинаторных NP-трудных задач, и ее строгое решение (нахождение оптимума) оказывается практически невозможным уже для графа с несколькими десятками вершин. В связи с этим был предложен ряд приближенных методов, основанных на разнообразных эвристиках [4, 5].

В настоящей статье предлагается также приближенный метод. В его основу положено представление булева пространства картой Карно, последовательное выделение в графе квадратов, их стыковка и размещение на булевом кубе. Используемые здесь термины поясняются ниже.

1. Визуальный метод размещения графа

Если число вершин n в графе G ограничено парой десятков, задачу его размещения в булевом пространстве можно решать визуально, отображая рисунками как граф G , так и k -мерный куб, на который требуется его спроектировать. При $k > 3$ традиционное (по аналогии со случаем $k = 3$) представление k -мерного куба на плоскости оказывается слишком громоздким, поэтому удобнее воспользоваться матрицей (картой) Карно, хорошо зарекомендовавшей себя при минимизации булевых функций [6, 7].

Например, при числе вершин n , не превышающем 16, граф G можно разместить на четырехмерном кубе, представив последний картой Карно размером 4×4 (рис. 3).

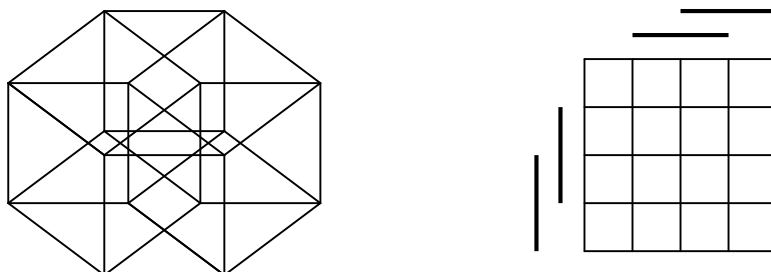


Рис. 3. Четырехмерный куб и соответствующая карта Карно

Множество клеток карты Карно изоморфно множеству вершин куба. Любой паре смежных вершин куба (связанных ребром) соответствует при этом пара клеток карты, соседних по

горизонтали или по вертикали, либо крайних в одном ряду (такие клетки также считаются соседними).

Введем в рассмотрение аналогичную по структуре многозначную матрицу Q , элементы которой могут принимать значения из множества $\{1, 2, \dots, n\}$. Процесс размещения графа G будем трактовать как приписывание некоторым элементам этой матрицы соответствующих значений, в результате чего матрица Q представит искомое взаимно однозначное отображение множества вершин V графа G в булево пространство M .

Введем следующие определения. Назовем квадратом цикл в графе, образованный четырьмя ребрами. Будем представлять квадрат упорядоченной четверкой вершин, через которые проходит данный цикл, начиная с вершины с минимальным номером, затем показывая смежную с ней вершину с меньшим номером, затем вершину, смежную с предыдущей, затем последнюю. Такое обозначение квадрата назовем каноническим. Например, в графе на рис. 4 существуют квадраты 1.3.7.6, 2.4.5.11, 2.6.10.11, 2.6.3.9 и т. д.

Непересекающиеся по вершинам ребра графа, входящие в некоторый квадрат, будем считать соседними. В каждом квадрате существуют две пары соседних ребер. Например, в квадрате 1.3.7.6 соседними являются ребра 1-3 и 7-6, а также 3-7 и 6-1. Будем говорить, что два квадрата стыкуются, если они обладают ровно одним общим ребром.

Рассматривая заданный граф, будем находить в нем соседние ребра и образуемые ими квадраты. Далее будем строить последовательность квадратов, находящихся в отношении стыковки, и размещать их на матрице Q .

При нормальном представлении некоторого квадрата в матрице Q его смежные вершины должны разместиться на соседних элементах матрицы. Следовательно, его можно представить конфигурацией матричных элементов, либо целиком заполняющих некоторый один ряд, либо образующих подматрицу размером 2×2 , которая составляется из соседних (по вертикали и горизонтали) элементов.

Соседние ребра и образуемые ими квадраты легко распознаются визуально на изображении графа и затем переносятся в матрицу Q . Продемонстрируем этот процесс на конкретном примере графа с 11 вершинами (рис. 4).

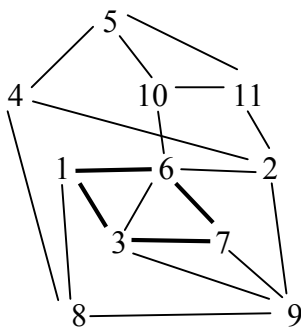


Рис. 4. Граф G

Выделим в графе некоторый квадрат. Пусть это будет цикл из четырех отмеченных на рисунке ребер, который можно представить соответствующим перечнем входящих в него вершин 1.3.7.6. Отобразим его конфигурацией

$$\begin{matrix} 1 & 6 \\ 3 & 7 \end{matrix}$$

Найдем стыкующийся с ним по ребру 1-3 квадрат 1.3.9.8 и отобразим множество перечисленных в двух квадратах вершин конфигурацией

$$\begin{matrix} 8 & 1 & 6 \\ 9 & 3 & 7 \end{matrix}$$

Далее находится стыкуемый с этой конфигурацией (по ребру 8-9) квадрат 2.4.8.9 и строится включающая его конфигурация

$$\begin{matrix} 4 & 8 & 1 & 6 \\ 2 & 9 & 3 & 7 \end{matrix}$$

Затем находится квадрат $\begin{matrix} & & & 5 \\ & & 4 & 8 & 1 & 6 \\ & 2 & 9 & 3 & 7 & \\ & & & & & 11 \end{matrix}$ и строится конфигурация

правом может быть признана за фрагмент (4×4) -матрицы Q , поскольку она не выходит за ее пределы. В результате на ребра четырехмерного куба проектируются все выделенные на рис. 5, а) ребра, включающие все вершины графа G , кроме вершины 10. Осталось разместить последнюю.

Учитывая, что вершина 10 смежна вершинам 5, 6 и 11, находим в матрице Q свободный элемент, соседний с теми, которые приняли значения 5 и 6, и помещаем туда номер этой вершины, расширяя конфигурацию до окончательного вида.

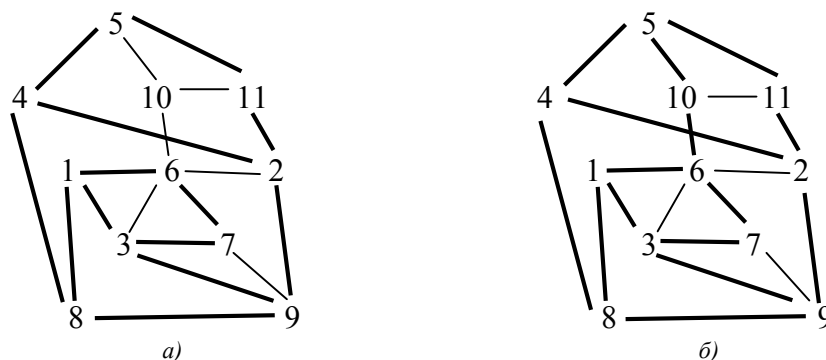


Рис. 5. Ребра графа, проектируемые на ребра четырехмерного куба: а) на предпоследнем шаге алгоритма; б) на последнем

Последовательный процесс роста конфигурации можно представить следующей цепочкой:

$$\begin{matrix} & & & & 5 & & & & 5 & 10 \\ 1 & 6 & \rightarrow & 8 & 1 & 6 & \rightarrow & 4 & 8 & 1 & 6 & \rightarrow & 4 & 8 & 1 & 6 \\ 3 & 7 & \rightarrow & 9 & 3 & 7 & \rightarrow & 2 & 9 & 3 & 7 & \rightarrow & 2 & 9 & 3 & 7 \\ & & & & & & & & & & & & & & & 11 & & & & 11 \end{matrix}$$

В результате в булево пространство M оказались спроектированными все ребра рассматриваемого графа G , кроме четырех: 2-6, 3-6, 7-9 и 10-11 – эти пары оказались не на соседних элементах пространства. Анализируя полученную в результате конфигурацию, которая определяет коды состояний соответствующего конечного автомата, нетрудно увидеть, что при его схемной реализации при каждом из соответствующих переходов будет производиться два переключения элементов памяти. Действительно, состояния 2 и 6 оказались расположенными в конфигурации на расстоянии 2 (по Хэммингу), то же можно сказать об остальных трех парах.

Легко доказать, что полученное решение оптимально. Для этого достаточно показать, что не существует решения, в котором число «неспроектированных» ребер меньше четырех. Очевидно, что эти ребра должны покрывать все нечетные циклы в графе. Поскольку рассматриваемый пример довольно прост, относительно небольшим перебором можно найти единственную тройку ребер, удовлетворяющую данному условию, а именно ребра 5-10, 1-6 и 3-7. Однако эта тройка не затрагивает квадраты 6-2-9-7 и 3-6-7-9, которые образуют фрагмент графа, оказывающийся небулевым (поскольку квадраты стыкуются по двум соседним ребрам). Следовательно, полученное выше решение оптимально.

2. Формализация метода на базе матрицы смежности

С ростом размерности графа его представление на рисунке и последующий визуальный анализ становятся затруднительными, поэтому целесообразнее перейти к матричному пред-

ставлению, позволяющему более строго сформулировать правила размещения графа в булевом пространстве и разработать соответствующую программу.

Для начала введем ряд определений, рассматривая для примера граф G с 15 вершинами, заданный матрицей смежности G , в которой для удобства обозрения элементы со значением 1, представляющие ребра графа, показаны квадратиками (рис. 6).

В этой матрице отмечены четыре элемента, которые представляют ребра графа, образующие квадрат 1.3.6.8. Назовем *противоположными* те из них, которые не находятся в одном ряду. Данный квадрат однозначно определяется парой противоположных элементов, задающих соседние ребра, – левым верхним из данной четверки (с координатами 1.3) и правым нижним (с координатами 6.8). Конкатенация этих пар координат составляет каноническое обозначение квадрата 1.3.6.8.

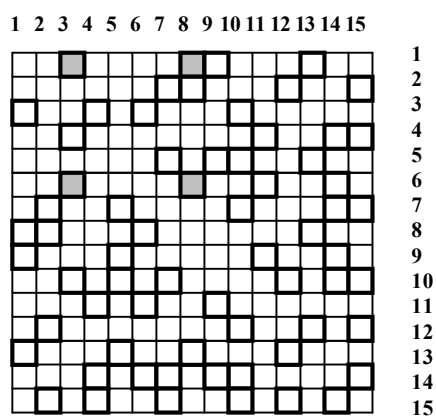


Рис. 6. Квадрат в графе

По данной матрице нетрудно найти все квадраты графа. Учитывая симметричность графа, из которой следует, что каждый квадрат имеет двойника, достаточно перебрать единичные элементы, расположенные выше главной диагонали, и найти противоположные им элементы, находящиеся ниже и правее (при программировании такого поиска полезно использовать операцию покомпонентной конъюнкции строк матрицы). Сформированные при этом пары образуют искомые квадраты.

Например, для элемента 1.3 находится единственный противоположный ему элемент 6.7, в то время как для элемента 3.4 – четыре элемента 11.6, 14.6, 14.1 и 15.10. Рассмотрение остальных элементов приводит к следующим результатам.

Элементы, для которых находятся противоположные элементы (правее и ниже), т. е. входящие в квадраты, отмечены на рис. 7.

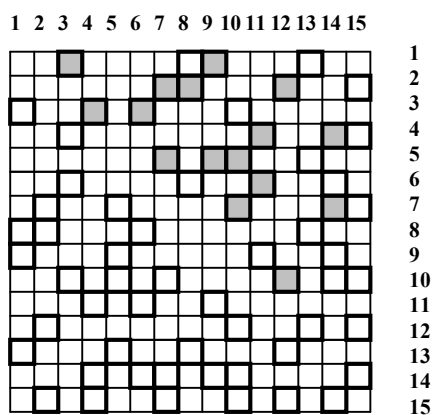


Рис. 7. Ребра, входящие в квадраты

Множество всех квадратов, заданных в канонической форме, представляется следующим списком: 1.3.6.7, 1.9.5.13, 2.7.10.12, 2.7.10.15, 2.7.14.15, 2.7.15.12, 2.8.13.12, 2.12.10.15, 3.4.11.6, 3.4.14.6, 3.4.14.1, 3.4.15.10, 3.6.14.10, 4.11.6.14, 4.11.9.14, 4.14.7.15, 4.14.10.15, 5.7.14.9, 5.7.14.10, 5.7.15.10, 5.9.14.10, 5.10.11.13, 6.11.9.14, 7.10.12.15, 7.10.14.15, 7.10.15.14, 7.14.10.16, 10.12.15.14.

Рассматривая граф $G = (V, E)$, упорядочим лексикографически множество его ребер E , представляя каждое ребро парой вершин в порядке возрастания их номеров и полагая, что ребро a - b старше ребра c - d , если $a > c$ или $b > d$ при равных a и c .

Выше было предложено решать задачу оптимального размещения графа в булевом пространстве путем последовательного усложнения конфигурации, представляющей собой фрагмент карты Карно. На очередном шаге этого процесса анализируется текущая конфигурация и определяется ее приращение.

Уточним понятие конфигурации. Будем считать, что это частично определенная матрица Q , некоторые элементы которой получили значения из множества V , а остальные остаются неопределенными. Положим, что представленные в текущей конфигурации вершины графа G составляют подмножество V^* множества V , а ребра графа G , представленные некоторыми парами соседних элементов конфигурации, составляют подмножество E^* множества E . Вершины из множества $V \setminus V^*$ и опирающиеся на них ребра графа G назовем *свободными*, равно как и представляющие эти ребра элементы матрицы Q . Очевидно, что свободные элементы матрицы Q расположены на пересечении строк и столбцов, соответствующих свободным вершинам.

При анализе текущей конфигурации будем рассматривать представленные в ней ребра в порядке их старшинства, начиная с младшей пары, и проверять каждое из них на наличие соседнего ребра из числа свободных. При положительном результате проверки будем стыковать найденную пару соседних ребер, соответственно расширяя конфигурацию, если это оказывается возможным в пределах матрицы Q .

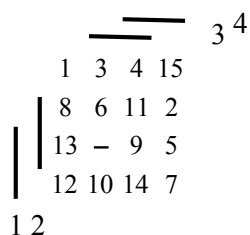
Для рассматриваемого примера этот процесс последовательного расширения конфигурации порождает показанную ниже последовательность значений матрицы Q . Она начинается с выбора младшего ребра 1-3, затем находится единственное соседнее с ним ребро 6-8 и строится квадрат с четырьмя ребрами, которые рассматриваются по очереди, для ребра 3-6 находится соседнее ребро 4-11 и т. д.:

1 3 --	--	→	1 3 --	--	→	1 3 4 -	-	→	1 3 4 -	-	→	1 3 4 15	-	→	1 3 4 15
----	----	→	8 6 --	--	→	8 6 11 -	-	→	8 6 11 -	-	→	8 6 11 -	-	→	8 6 11 -
----	----	→	----	----	→	----	----	→	----	----	→	-- 9 5	-	→	-- 9 5
----	----	→	----	----	→	----	----	→	- 10 14 -	-	→	- 10 14 7	-	→	- 10 14 7

На шестой конфигурации процесс нахождения соседних ребер из числа свободных затухает. Остаются три свободные вершины 2, 12 и 13, не вошедшие в последнюю конфигурацию. Рассматривая эти вершины по очереди, будем размещать их в матрице Q по возможности рядом с соседними с ними вершинами графа:

1 3 4 15	-	→	1 3 4 15	-	→	1 3 4 15	-	→	1 3 4 15
8 6 11 -	-	→	8 6 11 2	-	→	8 6 11 2	-	→	8 6 11 2
-- 9 5	-	→	-- 9 5	-	→	-- 9 5	-	→	13 - 9 5
- 10 14 7	-	→	- 10 14 7	-	→	12 10 14 7	-	→	12 10 14 7

Учитывая коды элементов в используемой карте Карно



получаем следующую матрицу кодирования вершин графа G :

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x_1	0	0	0	0	1	0	1	0	1	1	0	1	1	1	0
x_2	0	1	0	0	1	1	0	1	1	0	1	0	1	0	0
x_3	0	0	1	1	0	1	0	0	1	1	1	0	0	1	0
x_4	0	1	0	1	1	0	1	0	1	0	1	0	0	1	1

Так получается окончательный результат. Отметим в матрице Q те ребра графа G , которые удалось разместить на четырехмерном кубе (рис. 8):

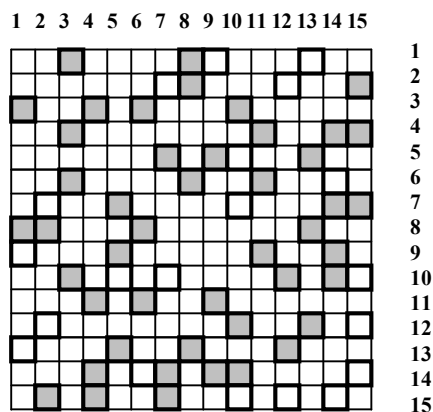


Рис. 8. Размещенные ребра графа (отмечены темным)

Остались неразмещенными десять ребер e_i графа (из общего числа 33), перечисленные ниже в свободной нумерации. Вершины в соответствующих им парах оказались на расстоянии d_i друг от друга, большем чем 1:

i	1	2	3	4	5	6	7	8	9	10
e_i	1-9	1-13	2-7	2-12	5-10	6-14	7-10	10-15	12-15	14-15
d_i	4	2	2	3	3	3	2	3	2	2

Представляет интерес оценка качества полученного решения. Точную оценку получить затруднительно, поскольку неизвестно оптимальное решение – неизвестен практически эффективный алгоритм его нахождения. Очевидно, что она сильно зависит от выбора примера.

В рассматриваемом примере суммарный дефект полученного размещения графа в булевом пространстве можно оценить величиной Def , определяемой как

$$Def = \sum_{i=1}^{10} (d_i - 1).$$

В данном случае она равна 16.

Возможен и другой способ оценки качества решения – через относительное число «задействованных» ребер гиперкуба. Теоретически максимальное число таких ребер четырехмерного куба при размещении графа с 15 вершинами равно $32 - 4 = 28$. В рассмотренном примере были задействованы 23 ребра, что составляет 80 %.

3. О программной реализации метода

Предложенный метод был с успехом опробован на серии примеров из Benchmark Set [8] вручную, в визуальном режиме. Эти примеры довольно просты, позволяют ограничивать число кодирующих переменных четырьмя–пятью, и в соответствующих графах переходов находится, как правило, достаточное количество пар соседних ребер, что обеспечивает применимость метода и его конкурентоспособность.

Результаты испытаний показывают, что данный метод может быть положен в основу программы оптимального размещения графа в булевом пространстве. Разумеется, при этом его следует дополнить рядом других процедур, например укладывающих на гиперкубе длинные цепочки ребер или учитывающих вероятности состояний и переходов.

Дело в том, что при решении задачи минимизации переключательной активности в схемной реализации автомата на ребра гиперкуба должны отображаться в первую очередь переходы с наибольшей вероятностью. В этом случае описанный выше алгоритм размещения графа в булевом пространстве следует модифицировать, предварительно дифференцируя ребра графа по весам – вероятностям соответствующих переходов.

Предположив, что все наборы значений входных переменных автомата равновероятны, сделать это можно следующим образом [2]. Сначала вычисляются условные вероятности $p(i, j)$ переходов из состояния i в состояние j (при условии, что автомат находится в состоянии i). Такая вероятность равна весу соответствующей булевой функции перехода, т. е. отношению числа входных наборов, обращающих ее в единицу, к общему числу наборов. Затем, рассматривая поведение автомата как марковский процесс, находят установившиеся вероятности $P(i)$ состояний автомата. Для этого методом Гаусса решается система линейных уравнений

$$P(i) = \sum_{k=1}^n P(k) p(k, i).$$

После этого по формуле $p^*(i, j) = P(i) p(i, j)$ вычисляются абсолютные вероятности переходов $p^*(i, j)$. В заключение находится вес $w(i, j)$ ребра, связывающего вершины i и j . Он равен сумме $p^*(i, j) + p^*(j, i)$.

В модифицированном алгоритме при рассмотрении свободных ребер, соседних с уже включенными в текущую конфигурацию, предпочтение отдается ребрам с более высоким весом.

Заключение

При синтезе логических схем, реализующих конечные автоматы, возникает задача такого размещения произвольного графа с n ребрами на гиперкубе, при котором по возможности максимизируется число ребер графа, проектируемых на ребра куба. Предложен эвристический метод ее решения, названный методом квадратов. Его временная сложность оценивается ориентировочно как не превышающая $O(n^3)$.

Список литературы

1. Najm F., Goel S., Hajj I.N. Power estimation in sequential circuits // Proc. of the 32th Design Automation Conference. – USA, 1995. – P. 635–640.
2. Luca Benini, Giovanni De Micheli. State assignment for low power dissipation // IEEE Journal of Solid-State Circuits. – Vol. 30. – № 3. – 1995. – P. 258–268.
3. Закревский А.Д. Алгоритмы преобразований таблицы переходов, устраняющие условия состязаний // Тр. СФТИ. Вып. 47. – 1965. – С. 48–55.

4. Noth W., Kolla R. Spanning tree based state encoding for low power dissipation // Proc. of DATE 99: Design, Automation and Test in Europe Conference, 1999.– P. 168–174.
5. Low-power state assignment techniques for finite state machines / P. Bacchetta, L. Daldoss, D. Scuito, C. Silvano // ISCAS 2000 – IEEE International symposium on circuits and systems. – Geneva, Switzerland, 2000. – P. 641–644.
6. Закревский А.Д. Визуально-матричный метод минимизации булевых функций // Автоматика и телемеханика. – 1960. – Т. 21. – № 3. – С. 368–373.
7. Закревский А.Д. Логический синтез каскадных схем. – М.: Наука, 1981.
8. McElvain K. LGSynth'93 Benchmark Set: Version 4.0, 1993 // <http://www.cbl.ncsu.edu/benchmarks>

Поступила 16.06.05

*Объединенный институт проблем
информатики НАН Беларуси,
Минск, Сурганова, 6
e-mail:zagr@newman.bas-net.by*

A.D. Zakrevskij

**ENERGY-SAVING CODING OF FINITE AUTOMATION STATES.
THE METHOD OF QUADRATES**

The problem of optimal mapping an arbitrary non-oriented graph into the Boolean space is considered, which has important applications in low-power circuit implementation of finite state machines by minimizing the switching activity. As a result the maximal number of the graph edges should be mapped onto some edges of a hypercube. A visual method for solving this problem is suggested, that uses matrices of adjacency and Karnaugh maps and consists in constructing a succession of rising configurations of quadrates and edges which could serve as some fragments of the hypercube.