# Генетический алгоритм синтеза дискретных управляющих систем на базе ПЛМ

А. С. Казимиров, С. Ю. Реймеров
(Иркутский государственный университет)

Рассматривается алгоритм синтеза дискретных управляющих систем в виде программируемых логических матриц. Программируемые логические матрицы строятся на основе полиномиальных нормальных форм булевых функций. Для дискретных управляющих систем важен результат на определенном подмножестве всех входных значений. Такие системы можно моделировать с помощью частично заданных булевых функций. Предлагается использовать генетические алгоритмы для нахождения близких к минимальным полиномиальных представлений таких функций.

**Ключевые слова:** логический синтез, полиномиальная нормальная форма, генетические алгоритмы, булевы функции.

Дискретные управляющие системы представляют собой логические схемы, призванные определенным образом реагировать на определенные входные значения. На вход системы поступает набор значений $x_1, x_2, \ldots, x_n$ из множества $\{0, 1\}$. На выходе получаются значения $y_1, y_2, \ldots, y_k$ из множества $\{0, 1\}$. При этом обычно выходные значения определены для некоторого подмножества всех входных значений, на остальных входных наборах ответ системы не имеет значения. Таким образом, каждое из выходных значений можно описать в виде частично заданной булевой функции. Булевы функции удобно задавать вектором значений на множестве всех входных наборов в лексикографическом порядке.

Одной из реализаций дискретных управляющих систем являются программируемые логические матрицы, представляющие собой логические схемы из элементов $\bar{x}$ (отрицание), $\cdot$ (конъюнкция) и $\oplus$ (сложение по модулю 2), соединенных таким образом, что из входных переменных и их отрицаний получаются все необходимые конъюнкции, которые затем суммируются для получения результата. Такие схемы являются реализацией полиномиальных нормальных форм булевых функций. Полиномиальным представлением булевой функции называется ее представление в виде суммы по модулю 2:

$$f(x_1, \ldots, x_n) = K_1 \oplus \ldots \oplus K_n,$$

где $K_i$ — произведение переменных или их отрицаний.

Каждая полностью определенная булева функция $n$ аргументов реализуется $2^{3^n - 2^n}$ различными полиномиальными представлениями. Одним из критериев оценки этих представлений является их сложность. Под сложностью полиномиального представления понимается количество слагаемых в этом представлении. Чем меньше сложность полинома, тем он предпочтительнее, так как меньшая сложность дает меньшие размеры и большую скорость работы электронных схем, построенных с использованием данного полинома. Поэтому сложность $L(g)$ полностью определенной функции $g$ определяется как количество слагаемых в наименьшем представлении, реализующем данную функцию. Полностью определенная функция $g$ является доопределением частично заданной функции $f$, если на множестве определения $f$ значения функций $f$ и $g$ совпадают. Поскольку на наборах, на которых функция $f$ не определена, функция $g$ может принимать любые значения, то для частично заданной функции в общем случае существует несколько различных доопределений. В такой ситуации естественно под сложностью частично заданной функции понимать наименьшую из сложностей ее доопределений.

Задача минимизации как для тотальных, так и для частично заданных булевых функций заключается в нахождении минимального полинома, реализующего данную функцию. Будем говорить, что полином реализует частично заданную функцию, если он реализует одно из ее доопределений.

В настоящее время существует алгоритм нахождения минимальных полиномов для булевых функций шести переменных [1], использующий библиотеку представителей классов N-эквивалентности функций пяти переменных. В [2] предложен алгоритм минимизации булевых функций шести и семи переменных сложности не более 16, основанный на ограниченном переборе функций меньшей размерности. Также существуют алгоритмы частичной минимизации булевых функций, основанные на частичном переборе [3, 4].

Любую булеву функцию $f(x_1, \ldots, x_n)$ можно представить в виде

$$f(x_1, \ldots, x_n) = f_1(x_1, \ldots, x_{n-1}) \oplus \\ \oplus\ x_n \cdot f_2(x_1, \ldots, x_{n-1}) \oplus \bar{x}_n \cdot f_3(x_1, \ldots, x_{n-1}).$$

Подставляя вместо $f_1$ произвольные функции $n-1$ переменной и вычисляя функции $f_2$ и $f_3$ из соотношений

$$f_2(x_1, \ldots, x_n) = f_1(x_1, \ldots, x_{n-1}) \oplus f(x_1, \ldots, x_{n-1}, 1)$$
$$f_3(x_1, \ldots, x_n) = f_1(x_1, \ldots, x_{n-1}) \oplus f(x_1, \ldots, x_{n-1}, 0),$$

можно получить все возможные полиномы, реализующие данную функцию $f$. Таким образом, можно задачу минимизации функции размерности $n$ свести к минимизации функций размерности $n-1$. При этом имеет место неравенство $L(f) \leqslant L(f_1) + L(f_2) + L(f_3)$. Но перебор всех функций меньшей размерности не представляется возможным осуществить уже при $n = 7$. Возможны различные варианты уменьшения множества перебора, применение которых в общем случае не дает точный минимум. Одним из таких способов является применение генетической модели.

В данной работе предлагается следующий алгоритм: создается первоначальная популяция, в которой в качестве особи используются двоичные вектора, представляющие функцию $f_1$. Эта функция является произвольной булевой функцией $n-1$ переменной. Приспособленность особи считается как сложность полинома исходной функции при подстановке в разложение функции, закодированной в данной особи. Сложность полинома получается из суммы сложностей функций $f_1$, $f_2$, $f_3$, которые рекурсивно минимизируются тем же алгоритмом. Генетический оператор мутации реализуется в виде изменения нескольких бит в векторе. Оператор кроссовера представляет собой обмен произвольными участками векторов.

Основной цикл генетического алгоритма заключается в последовательном построении новых особей с помощью генетических операторов с последовательным их оцениванием. Если полученные особи имеют лучшую приспособленность, чем уже имеющиеся, то они выталкивают из популяции худшие особи.

Предложенный алгоритм позволяет находить полиномиальные представления для частично заданных булевых функций от 8 и менее переменных.

Работа выполнена при поддержке РФФИ, проект № 16–31–00280.

## Список литературы

[1] Gaidukov A. Algorithm to derive minimum ESOP for 6-variable function // 5th International Workshop on Boolean Problems. — Freiberg, Germany, 2002. — P. 141–148.

[2] Sasao T. EXMIN2: A simplification algorithm for exclusive-OR sum-of-products expressions for multiple-valued-input two-valued-output functions // IEEE Trans. Comput.-Aided Des. Integrated Circuits & Syst. — 1993. — Vol. 12, no. 5. — P. 621–632.

[3] Винокуров С. Ф., Казимиров А. С. Генетический алгоритм поиска минимальных полиномов булевых функций // Дискретная математика и ее приложения: Материалы X Международного семинара. — М.: Мех.-мат. факультет МГУ, 2010. — С. 175–177.

[4] Казимиров А. С., Реймеров С. Ю. Вычислительная оценка сложности полиномиальных представлений булевых функций // Известия Иркутского государственного университета. Серия: Математика. — Иркутск: ГОУ ВПО «Иркутский государственный университет», 2010. — Том 3. № 4. — С. 33–43.