Рекурсивный генетический алгоритм минимизации частично заданных булевых функций

С. Ф. Винокуров[1], А. С. Казимиров[2], А. С. Францева[3]
Иркутский государственный университет
[1]servin38@gmail.com, [2]a.kazimirov@gmail.com, [3]a.s.frantseva@gmail.com

***Аннотация.* Рассматривается задача нахождения минимальных полиномов частично заданных булевых функций. Частично заданные функции часто возникают на практике в задачах логического синтеза, когда имеет значение поведение функции только на определенных наборах, например, при реализации конечных автоматов. Для решения данной задачи предлагается рекурсивный генетический алгоритм приближенной минимизации, основанный на спуске к функциям пяти переменных с последующим их доопределением.**

Ключевые слова: булевы функции; минимизация; полиномиальная нормальная форма; генетические алгоритмы

I. Введение

Полиномиальной нормальной формой или полиномом называется представление булевой функции в виде суммы по модулю 2

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

где каждое из K_i является произведением переменных или их отрицаний, в частности K_i может быть равно 1.

Среди всех классов полиномиальных представлений полиномиальные нормальные формы являются наиболее широким классом и требуют наименьшего числа слагаемых для представления булевых функций. Логические схемы, построенные с помощью элементов «сложение по модулю 2» имеют некоторые преимущества перед схемами, использующими только элементы И и ИЛИ. Такие схемы имеют меньший размер для многих функций, использующихся на практике [2], и обладают лучшей тестируемостью [1].

Для каждой булевой функции n аргументов существует $2^{3^n-2^n}$ различных полиномиальных представлений. Одним из основных параметров таких представлений является их длина как количество слагаемых. Сложностью полинома называется количество слагаемых в этом полиноме. Сложность функции f определяется как сложность наименьшего полинома, представляющего эту функцию, и обозначается $L(f)$. Задача минимизации булевой функции заключается в получении минимального полиномиального представления заданной функции.

Размер формулы, представляющей булеву функцию, влияет на размер и эффективность логических схем, реализующих эту формулу. В частности, полиномиальные представления используются в программируемых логических матрицах.

Существует множество алгоритмов минимизации, основанных на частичном переборе и преобразовании формул. Но такие алгоритмы не гарантируют минимальность полученных полиномов. Алгоритмы, основанные на разложении функции [3, 4, 6], позволяют находить минимальные полиномы функции n аргументов перебором всех $(n-1)$-местных функций, но при этом не позволяют за реальное время найти полиномы для функций 6 и более аргументов.

Известно несколько алгоритмов приближенной минимизации для полиномиальных форм и их подклассов [8, 14], в которых полный перебор подфункций заменяется перебором с помощью генетического алгоритма [9, 13]. В данной статье рассматривается модификация алгоритма из [10].

II. Разложение булевых функций

Один из подходов к минимизации булевых функций заключается в разложении функции в сумму трех функций меньшего числа аргументов.

Любое полиномиальное представление булевой функции $f(x_1,\ldots,x_n)$ можно представить в следующем виде:

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

Такое разложение можно использовать в алгоритмах минимизации для спуска к функциям меньшей размерности. Данные три функции также могут быть представлены полиномами, но без вхождения переменной x_n и ее отрицания. Другими словами, $x_n f_1$ является частью полинома, содержащей x_n; $\bar{x}_n f_2$ – часть полинома, содержащая отрицание x_n; f_3 – часть полинома, не зависящая от x_n.

Если выбрать функции f_1, f_2 и f_3 таким образом, чтобы сумма их сложностей была минимальной, то тогда можно будет получить минимальный полином для f:

$$L(f) = \min(L(f_1) + L(f_2) + L(f_3))$$

Таким образом, для минимизации функции f нужно перебрать все возможные функции f_1, f_2 и f_3. Среди этих функций только одна может быть выбрана произвольным образом, остальные единственным образом получаются из следующих равенств (которые получаются подстановкой значений 0 и 1 вместо x_n в разложение минимизации):

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

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

Любую из этих функций можно выбрать в качестве параметра, остальные будут определены первой функцией. Таким образом, минимизацию n-местной функции f можно свести к задаче поиска такой $(n-1)$-местной функции f_3, что значение $L(f_1)+L(f_2)+L(f_3)$ будет минимальным.

Для уменьшения объема перебора можно использовать классы эквивалентности по преобразованиям, сохраняющим сложность функции. Наиболее общими из таких преобразований являются LP-преобразования [5, 7, 11]. LP-преобразование формулы заключается в перестановке переменных и/или перестановке слагаемых f_1, f_2 и f_3. Такие преобразования не изменяют количество слагаемых в полиноме, таким образом, LP-эквивалентные функции имеют одинаковую сложность.

Точная минимизация булевых функций в настоящее время возможна только для $n \le 6$ [3] и некоторых функций большего числа аргументов [4]. Алгоритм точной минимизации основывается на исчерпывающем переборе всех $(n-1)$-местных функций, сложности которых можно рассчитать заранее. Для функций 4 и менее аргументов можно осуществить спуск до унарных функций. Для минимизации функций большего числа аргументов удобнее хранить предрасчитанные сложности.

III. Генетический алгоритм минимизации

Различные ограничения на пространство поиска для функции f_3 приводят к разным алгоритмам приближенной минимизации. Такие алгоритмы получают верхнюю оценку сложности функции и полиномиальное представление, которое для некоторых функций будет хуже, чем минимальное.

Одним из способов уменьшения перебора является генетическая модель.

Для булевых функций существует несколько различных представлений. Одним из основных способов задания функции является формульное представление. Но поскольку область определения булевой функции конечна, то можно представить функцию в виде таблицы всех ее значений, лексикографически упорядоченного по векторам входных значений. В таком случае значения аргументов избыточны и их можно опустить. Таким образом, булеву функцию можно представить вектором из 2_n значений.

Генетический алгоритм минимизации булевых функций основан на использовании вектора функции f_3 в качестве хромосомы одной особи в популяции. Алгоритм можно записать следующим образом.

Сначала генерируется стартовая популяция различных вариантов функции f_3. Каждая функция в популяции оценивается с помощью функции приспособленности, равной $L(f_1)+L(f_2)+L(f_3)$. Популяция сортируется по убыванию приспособленности. После этого выбираются особи для мутации и скрещивания с вероятностью, линейно убывающий от первой к последней особи.

Мутация реализуется как случайное изменение одного бита в хромосоме. Скрещивание представляет собой объединение частей двух хромосом, соответствующих подфункциям, полученных подстановкой значений 0 и 1 вместо x_i.

При этом изменение одного бита в векторе функции f_3 меняет сложность соответствующего полинома для f не более, чем на 3. Таким образом, пространство поиска является гладким, что подразумевает хорошую применимость генетической модели для данной задачи.

С помощью данного алгоритма была получена верхняя оценка сложности всех 7-местных функций [12].

IV. Минимизация частично заданных функций

Назовем булеву функцию f доопределением частично заданной функции g, если их значения совпадают для всех входных наборов, на которых g определена.

Алгоритм минимизации частично заданной функции f начинает работу с разложения на три функции f_1, f_2, f_3. Функция f_3 является полностью определенной. Популяция строится из набора случайно выбранных f_3. Остальные функции вычисляются из соотношений

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

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

Значения f_1, f_2 для входов, на которых f не определена, также задаются как неопределенности. Все три функции в разложении минимизируются рекурсивно данным алгоритмом. Полностью определенные функции 5 аргументов минимизируются алгоритмом точной минимизации. Частично заданные функции 5 аргументов минимизируются другим генетическим алгоритмом, который подбирает наилучшее доопределение.

Использование данного алгоритма для функций 8 и более аргументов предполагает не менее трех этапов спуска к меньшей размерности, для каждого из которых генерируется своя популяция для промежуточных функций. Для оптимизации данного рекурсивного алгоритма предлагается для каждой функции сохранять по одной лучшей особи для промежуточных функций f_1, f_2, f_3. Поскольку в результате мутации функции f получается некоторая функция f^*, вектор которой отличается всего в одном бите от исходного, то вектора функций f_1^*, f_2^*, f_3^* тоже будут иметь не более одного отличия от векторов f_1,

f_2, f_3. Таким образом, если добавить в популяцию лучшие особи из прошлой итерации, то можно получить близкие к минимальным полиномы для f_1^*, f_2^*, f_3^* за меньшее число шагов. Для минимизации функции 8 переменных и при размере популяции, равном p, необходимо хранить в памяти p функций 7 переменных, $3p$ функций 6 переменных и $9p$ функций 5 переменных, а также $9p$ лучших уточнений для функций 5 переменных.

Суммарный объем расходуемой дополнительно памяти исчисляется килобайтами и получается ничтожным по сравнению с объемами массивов предрасчитанных сложностей. При этом рекурсивный алгоритм с сохранением позволяет находить приближенные к минимальным полиномы в десятки раз быстрее по сравнению с алгоритмом из [10].

СПИСОК ЛИТЕРАТУРЫ

[1] Chatterjee M., Pradhan D.K., Kunz W. LOT: Logic Optimization with Testability. New Transformations for Logic Synthesis, IEEE Trans. CAD, Vol.17, No.5, 1998, pp. 386–399.

[2] Debnath D., Sasao T. A heuristic algorithm to design AND-OR-EXOR three-level networks. Proc. Asia and South Pacific Design Automation Conference, 1998, pp. 69–74.

[3] Gaidukov A. Algorithm to derive minimum ESOPs for 6-variable functions. Proceedings of the 5th International Workshop on Boolean Problems 2002, Freiberg, Germany, Sept. 19-20, 2002, pp. 141-148.

[4] Hirayama T., Nishitani, Y. Exact minimization of AND-EXOR expressions of practical benchmark functions, Journal of Circuits, Systems and Computers, Vol. 18, No. 3, 2009, pp. 465–486. DOI: 10.1142/S0218126609005356.

[5] Koda N. LP equivalence class of logic functions, IFIP 10.5 Workshop on Application of the Reed-Muller expansion in Circuit Design, Sept. 1993. pp. 99-106.

[6] Nishitani Y., Shimizu K. Lower bounds on size of periodic functions in exclusiveOR sum-of-products expressions, IEICE Trans. Fundamentals, E77-A, 1994, pp. 475–482.

[7] Винокуров С.Ф., Казимиров А.С. Перечисление операторных классов булевых функций // Известия Иркутского государственного университета. Серия: Математика. Иркутск: ГОУ ВПО «Иркутский государственный университет», 2009. Том 2. № 2. С. 40–55.

[8] Винокуров С.Ф., Францева А.С., Рябец Л.В., Тодиков С.И. Алгоритм построения минимального представления многовыходных функций алгебры логики в классе обратимых схем // XX Международная конференция по мягким вычислениям и измерениям (SCM-2017). Сборник докладов в 3-х томах. Санкт-Петербург. 24–26 мая, 2017. Том 2. С. 175-178.

[9] Ильин Б.П. Эволюционные алгоритмы в задаче минимизации булевых функций / Б.П. Ильин, А.С. Казимиров, В.И. Пантелеев, С.Ю. Реймеров, Н.Л. Семичева // XX Международная конференция по мягким вычислениям и измерениям (SCM-2017). Сборник докладов в 3-х томах. Санкт-Петербург24–26 мая, 2017. Том 1. С. 482–485.

[10] Казимиров А.С. Генетический алгоритм минимизации частично заданных булевых функций // Вестник Бурятского университета: Математика и информатика. Улан-Удэ: Бурятский госуниверситет, 2006. Серия 13. Выпуск 3. С. 28–32.

[11] Казимиров А.С. Оценка числа классов LP-эквивалентности булевых функций // Вестник Бурятского университета: Математика и информатика. Улан-Удэ: Бурятский государственный ун-т, 2005. Серия 13. Выпуск 2. С. 17–22.

[12] 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.

[13] Казимиров А.С., Реймеров С.Ю. Генетические алгоритмы и нейронные сети в минимизации булевых функций // XIX Международная конференция по мягким вычислениям и измерениям SCM'2016 : Материалы конф. Санкт-Петербург, 2016. С.474–476.

[14] Францева А.С., Винокуров С.Ф., Рябец Л.В. Генетический алгоритм в задаче поиска минимальных представлений функций алгебры логики в классе кронекеровых форм // XX Международная конференция по мягким вычислениям и измерениям (SCM-2017). Сборник докладов в 3-х томах. Санкт-Петербург. 24–26 мая, 2017. Том 1. С. 501-504.