С помощью введения специальных операций над сжатыми графами, свободными от собственных ребер, сохраняющих данные свойства, и с использованием данной теоремы доказывается следующее утверждение.

Утверждение 3. Существуют непустые сжатые графы, свободные от зафиксированных клик, имеющие $\rho(G) = \rho$ и $\Delta(G) = \Delta$, где Δ и ρ — произвольные натуральные числа, удовлетворяющие ограничениям: $\rho \geqslant 3$, $\Delta \geqslant \rho + 1$.

ЛИТЕРАТУРА

1. *Cavers M. S.* Clique partitions and coverings of graphs. University of Waterloo, 2005.
2. *Kou L. T., Stockmeyer L. J., Wong C. K.* Covering edges by cliques with regard to keyword conflicts and intersection graphs // Communicat. ACM. 1978. V. 21. No. 2. P. 135–139.
3. *Orlin J.* Contentment in graph theory: Covering graphs with cliques // Indagationes Math. 1977. V. 39. P. 406–424.

УДК 519.7

БЕНТ-ФУНКЦИИ И ЛИНЕЙНЫЕ КОДЫ В CDMA[1]

А. В. Павлов

Булева функция от четного числа переменных называется *бент-функцией*, если она максимально удалена от класса аффинных булевых функций. Задача построения бент-функций возникает во многих областях, в том числе в теории кодирования, где находит свое применение в системах коллективного доступа, таких, как стандарты цифровой сотовой связи CDMA. Данные стандарты используют бент-функции для построения кодов постоянной амплитуды, что позволяет предельно снизить коэффициент отношения пиковой и средней мощностей сигнала. Такие коды состоят из векторов значений бент-функций. И как известно, предпочтение отдается линейным кодам, так как они довольно просты в реализации.

Так возникла задача построения максимального линейного кода на основе заданной бент-функции, такого, что при сдвиге данной бент-функции на любое кодовое слово не нарушалось бы свойство «бент». В [1] предлагается использовать для построения кода конструкцию Мак-Фарланда [2] $f(x, y) = \langle x, \pi(y)\rangle + g(y)$, где $x, y \in E^{n/2}$; $g(y)$ — булева функция от $n/2$ переменных; π — подстановка на $E^{n/2}$; $E^{n/2}$ — булев куб размерности $n/2$. Рассмотрим линейный код длины 2^n, состоящий из векторов значений функций $h(x, y) = g(y)$ и всех аффинных функций от n переменных. Размерность данного кода равна $k = 2^{n/2} + n/2$, кодовое расстояние равно $d = 2^{n/2}$. Например, для любой бент-функции из класса Мак-Фарланда от 6 переменных имеем линейный код с параметрами $[2^6, 11, 8]$, а для 8 переменных — с параметрами $[2^8, 20, 16]$.

В [3] было доказано, что две бент-функции находятся на минимальном расстоянии $2^{n/2}$ друг от друга тогда и только тогда, когда они отличаются на аффинном подпространстве размерности $n/2$ и обе на нём аффинны. Исходя из этого критерия, предложен следующий алгоритм построения максимального линейного кода.

Алгоритм

1) Вход: бент-функция f.
2) Добавляем f в список функций $functionList$.

[1]Работа выполнена при финансовой поддержке гранта Президента РФ для молодых российских ученых (грант МК № 1250.2009.1).

3) Строим все аффинные подпространства размерности $n/2$, на которых данная бент-функция аффинна (см. [3]), и добавляем их в список $list$.
4) Далее вызываем рекурсивную функцию **findCode**(f, $list$, $functionList$).

findCode(f, $list$, $functionList$)

1) Вход: бент-функция f; список аффинных подпространств, на которых f аффинна; список бент-функций.
2) Для каждого аффинного подпространства из $list$ строим бент-функцию g на минимальном расстоянии от f.
3) Если g линейно независима со всеми функциям из $functionList$, то добавляем g в $functionList$.
4) Далее формируем список аффинных подпространств $newList$. Для каждого аффинного подпространства из $list$ проверяем условие: если функция g аффинна на данном аффинном подпространстве, добавляем это аффинное подпространство в $newList$.
5) Вызываем рекурсивную функцию **findCode**(g, $newList$, $functionList$).

С помощью данного алгоритма удалось построить линейные коды для бент-функций от 6 переменных и для некоторых бент-функций от 8 переменных. Стоит заметить, что построенные по предлагаемому алгоритму коды не обязательно являются оптимальными.

Нетрудно показать, что для аффинно эквивалентных функций размерности таких кодов будут одинаковыми. Далее приведём таблицу с аффинно неэквивалентными бент-функциями и размерностями кодов, построенных для них.

n	Бент-функция	Размерность кода
6	$x_1x_2x_3 \oplus x_2x_4x_5 \oplus x_3x_4x_6 \oplus x_1x_4 \oplus x_2x_6 \oplus x_3x_4 \oplus$ $\oplus x_3x_5 \oplus x_3x_6 \oplus x_4x_5 \oplus x_4x_6$	15
6	$x_1x_2x_3 \oplus x_2x_4x_5 \oplus x_1x_2 \oplus x_1x_4 \oplus x_2x_6 \oplus x_3x_5 \oplus x_4x_5$	15
6	$x_1x_2x_3 \oplus x_1x_4 \oplus x_2x_5 \oplus x_3x_6$	15
6	$x_1x_2 \oplus x_3x_4 \oplus x_5x_6$	15
8	$x_1x_2x_3 \oplus x_2x_4x_5 \oplus x_3x_4x_6 \oplus x_1x_4x_7 \oplus x_3x_5 \oplus$ $\oplus x_2x_7 \oplus x_1x_5 \oplus x_1x_6 \oplus x_4x_8$	29
8	$x_1x_2x_3 \oplus x_2x_4x_5 \oplus x_3x_4x_6 \oplus x_3x_5 \oplus x_2x_6 \oplus$ $\oplus x_2x_5 \oplus x_1x_7 \oplus x_4x_8$	28
8	$x_1x_2x_3 \oplus x_2x_4x_5 \oplus x_3x_4x_6 \oplus x_3x_5 \oplus x_1x_3 \oplus$ $\oplus x_1x_4 \oplus x_2x_7 \oplus x_6x_8$	30
8	$x_1x_2x_3 \oplus x_2x_4x_5 \oplus x_3x_4x_6 \oplus x_3x_5 \oplus x_2x_6 \oplus$ $\oplus x_2x_5 \oplus x_1x_2 \oplus x_1x_3 \oplus x_1x_4 \oplus x_7x_8$	30
8	$x_1x_2x_3 \oplus x_2x_4x_5 \oplus x_3x_4x_6 \oplus x_3x_5 \oplus x_1x_6 \oplus x_2x_7 \oplus x_4x_8$	28
8	$x_1x_2x_7 \oplus x_3x_4x_7 \oplus x_5x_6x_7 \oplus x_1x_4 \oplus x_3x_6 \oplus$ $\oplus x_2x_5 \oplus x_4x_5 \oplus x_7x_8$	29
8	$x_1x_2x_3 \oplus x_2x_4x_5 \oplus x_3x_4 \oplus x_2x_6 \oplus x_1x_7 \oplus x_5x_8$	28
8	$x_1x_2x_3 \oplus x_2x_4x_5 \oplus x_1x_3 \oplus x_1x_5 \oplus x_2x_6 \oplus x_3x_4 \oplus x_7x_8$	30
8	$x_1x_2x_3 \oplus x_1x_4 \oplus x_2x_5 \oplus x_3x_6 \oplus x_7x_8$	29
8	$x_1x_2 \oplus x_3x_4 \oplus x_5x_6 \oplus x_7x_8$	28

Так же как и в кодах, основанных на конструкции Мак-Фарланда, код, полученный с помощью предложенного алгоритма, имеет то же кодовое расстояние, равное

минимальному расстоянию между бент-функциями $2^{n/2}$. Особенностью метода является то, что он зависит от конкретного вида бент-функции, в отличие от конструкции Мак-Фарланда.

ЛИТЕРАТУРА

1. *Paterson K. G.* Sequences For OFDM and Multi-code CDMA: two problems in algebraic Coding Theory // Sequences and their applications. Seta 2001. Second Int. Conference (Bergen, Norway, May 13–17, 2001). Proc. Berlin: Springer, 2002. P. 46–71.
2. *McFarland R. L.* A family of difference sets in non-cyclic groups // J. Combin. Theory. Ser. A. 1973. V. 15. No. 1. P. 1–10.
3. *Коломеец Н. А., Павлов А. В.* Свойства бент-функций, находящихся на минимальном расстоянии друг от друга // Прикладная дискретная математика. 2009. № 4. С. 5–20.

УДК 519.175.1

О НОВОМ ПОЛНОМ ИНВАРИАНТЕ АЦИКЛИЧЕСКИХ ГРАФОВ

А. В. Пролубников

В задаче проверки изоморфизма графов (задача ИГ) даны два обыкновенных графа с одинаковым числом вершин и ребер. Необходимо ответить на вопрос, существует ли такое биективное отображение (изоморфизм) множества вершин одного графа на множество вершин второго, которое сохраняло бы смежность соответствующих вершин?

Поскольку любой алгоритм решения задачи ИГ представляет собой проверку равенства некоторых инвариантных относительно изоморфизма количественных характеристик графов, неясный статус задачи ИГ в иерархии теории сложности непосредственно связан с вопросом сложности вычисления полного инварианта графа. На данный момент не доказано, что задача ИГ является NP-полной, равно как и не найдено полиномиальных алгоритмов решения общего случая задачи.

Единственным известным полным инвариантом графа является его канонический код — максимальное число, двоичная запись которого может быть получена путем некоторой конкатенации строк верхне-(нижне-)треугольной подматрицы матрицы смежности графа [1].

Полные инварианты известны лишь для немногих относительно простых классов графов, поскольку наличие полиномиально вычислимого полного инварианта для графов из некоторого класса эквивалентно полиномиальной разрешимости задачи ИГ для графов из этого класса. Так, в работе [2] представлен полный инвариант для деревьев и в целом класса ациклических графов, в работе [3] — для планарных графов. Однако в этих работах, как и в большинстве работ, нацеленных на нахождение полного инварианта для ограниченного класса графов, полный инвариант ищется как результат канонизации графа — процесс, который может быть описан следующим образом. Пусть $\mathbf{G}$ — некоторый класс графов. Пусть $f : \mathbf{G} \to \{0,1\}^*$ — функция, отображающая граф в пространство битовых строк (канонических кодов), такая, что для всех $G, H \in \mathbf{G}$ имеем $G \simeq H \Leftrightarrow f(G) = f(H)$, то есть f — полный инвариант для графов из $\mathbf{G}$. Если f дает для G граф $f(G)$ такой, что $G \simeq f(G)$, то $f(G)$ — канонический код графа, по которому восстанавливается сам граф.

В этой работе предлагается алгебраический полный инвариант ациклических графов, который не получается в результате канонизации графа, а представляет собой множество из $1 + n(n+1)/2$ числовых значений, каждое из которых есть произведе-