

УДК 519.714.71

## **Конъюнктивная декомпозиция булевых функций: эксперименты с различными представлениями**

*Емельянов П.Г. (Институт систем информатики СО РАН, Новосибирский  
государственный университет)*

В статье представлены результаты экспериментов по конъюнктивной декомпозиции различных представлений булевых функций (ZDD, BDD, OKFDD) методами, которые получаются путем специализации общего алгоритма декомпозиции. Тестовыми наборами являются случайные булевы функции с различными параметрами, а также набор широко известных бенчмарков, используемых для тестирования алгоритмов оптимизации логических схем. В сравнении участвуют последовательная и многопоточная реализация алгоритма.

**Ключевые слова:** булевы функции, задача оптимизации представления, конъюнктивная декомпозиция с непересекающимися носителями, представления булевых функций.

### **1. Введение**

Композиционное построение новых объектов - мощное средство современной математики. Примером этого может служить фундаментальная операция декартового произведения. Обращение композиции – декомпозиция – сложных объектов является важнейшим методологическим приемом математики, позволяющий снизить сложность их анализа и преобразований. Декомпозиция некоторого обобщения декартового произведения, известного в компьютерных науках как произведение семейств множеств (Family Product), является одним из важных вариантов декомпозиции, имеющий многочисленные приложения как в фундаментальных областях математики (декомпозиция некоторых алгебраических структур, раскраска гиперграфов и т.д.), так и технологических областях (оптимизация комбинационной части логических схем, декомпозиция таблиц баз данных и таблиц принятия решений, декомпозиция онтологий и т.д.).

Цель данной статьи – представить результаты исследований, относящиеся к изучению свойств конъюнктивной/дизъюнктивной декомпозиции различных представлений булевых функций. Особенностью подхода является то, что алгоритм декомпозиции для конкретного представления получается путем специализации обобщенного алгоритма.

## 2. Теоретические основания

Произведение семейств множеств определяется следующим, образом. Если  $A$  и  $B$  – множества подмножеств некоторых непересекающихся конечных базовых множеств, то  $A \otimes B = \{ a \cup b \mid a \in A \wedge b \in B \}$ . Довольно просто установить связь между произведением семейств множеств и произведением мультилинейных полиномов над полем  $GF(2)$  с непересекающимися множествами переменных, если предположить, что мономы представляют элементы  $A$  и  $B$ .

Таким образом, задача декомпозиции произведение семейств множеств сводится к задаче факторизации указанных полиномов. Впервые полиномиальная сложность задачи факторизации мультилинейный полиномов над конечными полями была указана Шпилькой и Волковичем в 2010 году [10] для полиномов, представленных в виде арифметических схем. Алгоритм (по существу, несколько алгоритмов) полиномиальной временной сложности для полиномов, представленных в виде списка мономов (в точки зрения булевых функций – это представление в АНФ), был получен Емельяновым и Пономаревым в 2014 году [4, 5]. Это дает эффективное средство декомпозиции сразу для нескольких представлений булевых функций, так как эти представления напрямую выражаются с помощью полиномов. С учетом эффективного взятия отрицания и, соответственно, возможности проведения дизъюнктивной декомпозиции, такими представлениями являются полные/частичные таблицы истинности, позитивные ДНФ/КНФ, СДНФ/СКНФ.

Однако теоретическая эффективность не означает практическую. А так как эта задача возникает в важных прикладных областях, среди которых укажем лишь оптимизацию логических схем и анализ данных и извлечение знаний, то ее дальнейшее изучение в этом направлении имеет особую актуальность. Понимание причин, по которым та или иная задача имеет такую сложность, важно для принятия решений при разработке программного обеспечения. Кроме того, на практике представление объектов декомпозиции, оказывает существенное влияние на эффективность декомпозиции/факторизации.

Предыдущие исследования показали, что алгоритм [6], основанный на неявном вычислении формальной производной некоторого полинома, построенного из исходного, может быть адаптирован для произвольных представлений булевых функций и, при выполнении некоторых дополнительных условий, их временная сложность может быть даже полиномиальной. Таким образом, разработан обобщенный алгоритм конъюнктивной

декомпозиции, при соответствующей специализации небольших частей применим различным представлениям (при некоторых предположениях это также гарантирует полиномиальную сложность полученных алгоритмов). Это чрезвычайно актуальное свойство в рамках технологических цепочек, когда представление функций предписано и конвертация их в более "удобное" для декомпозиции представление снижает общую эффективность процесса.

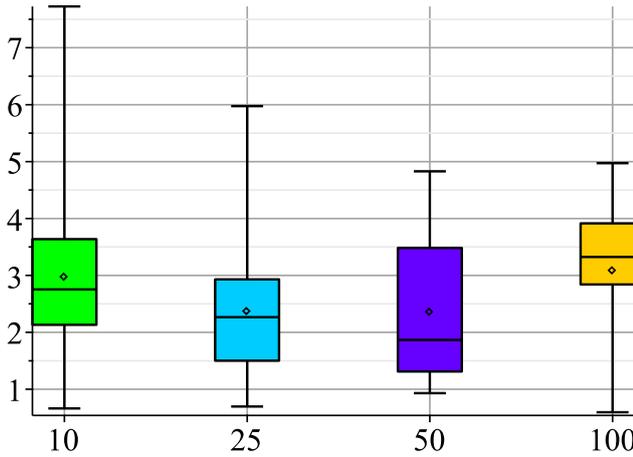
### 3. Эксперименты с различными представлениями

В рамках проекта по исследованию задачи декомпозиции велись исследования версий алгоритма декомпозиции для важных в рамках синтеза логических схем представлений таких, как частичные таблицы истинности, BDD [2], OKFDD [3], AIG [8]. На данный момент, для применяемых в микроэлектронной индустрии представлений не известны эффективные алгоритмы конъюнктивной/дизъюнктивной декомпозиции с непересекающимися множествами входных переменных (известны довольно быстрые эвристики, которые не гарантируют отсутствие у компонент общих переменных [1]).

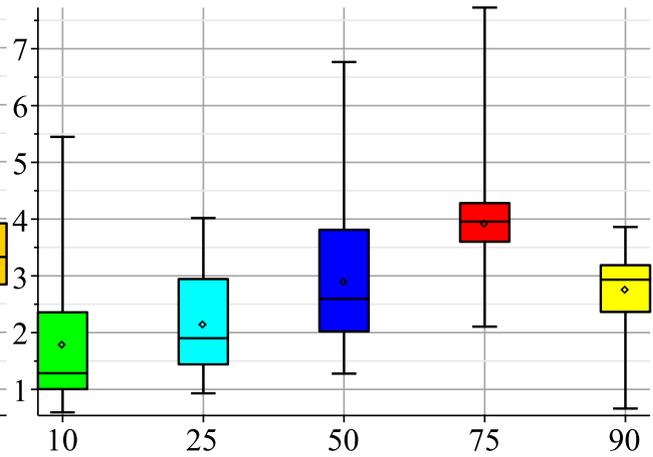
Вначале была выполнена параллельная реализация алгоритма конъюнктивной декомпозиции булевых функций в АНФ форме, где полиномы представлялись в виде ZDD [7]. Для реализации была использована библиотека CUDD 3.0 [11]. Были сгенерированы два набора случайных полиномов, являющихся произведением двух компонент с непересекающимися носителями, со следующими характеристиками. Все содержат 100 переменных, которые разбиваются по компонентам 10 на 90, 25 на 75, 50 на 50, 75 на 25, 90 на 10. Один набор – это полиномы, имеющие 104 мономов (по компонентам 10 на 1000, 25 на 400, 50 на 200, 100 на 100), и второй – 105 мономов (100 на 1000, 200 на 400, 316 на 316).

Результаты тестирования первого набора (в вариантах оптимального и случайного упорядочивания переменных и однопоточного и 8-поточного исполнения) представлены на бокс-диаграммах (см. рис. 4). В представленных диаграммах ускорение означает отношение времени исполнения в 8-поточном режиме ко времени исполнения в однопоточном режиме. В общем и целом, результаты тестирования подтверждают гипотезы о зависимости сложностных характеристиках алгоритма от характеристик компонент декомпозиции. Отметим, что для реализации с использованием библиотеки CUDD потребовалась новая по сравнению с реализацией на Maple схема мемоизации результатов вычислений, которая ускорила вычисления на 30-40%, «эффективность кэша» (отношение числа успехов к общему числу обращений) варьировалась от 0.01 до 0.58.

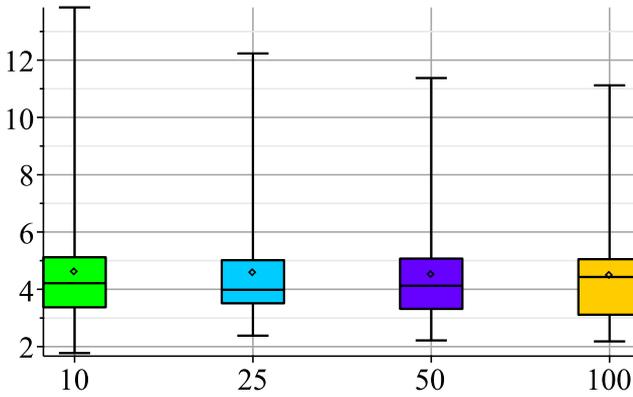
Speedup for the optimal ordering grouped by monom number



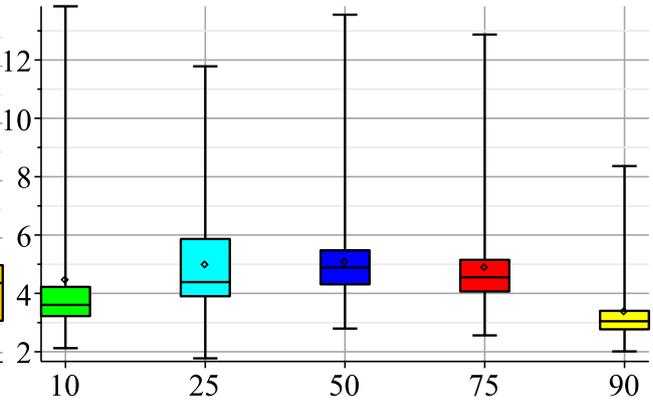
Speedup for the optimal ordering grouped by variable number



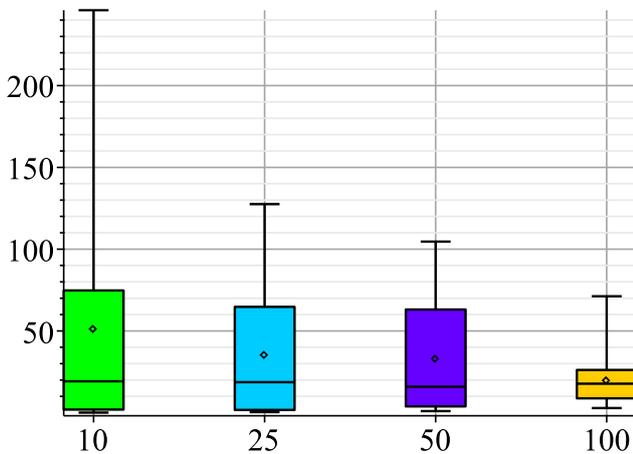
Speedup for a random permutation grouped by monom number



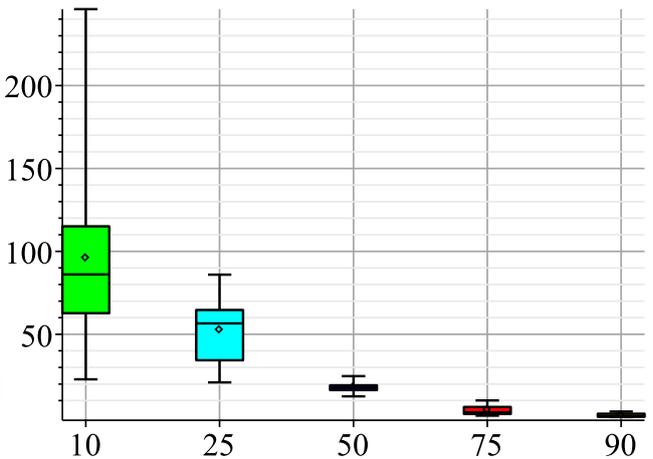
Speedup for a random permutation grouped by variable number

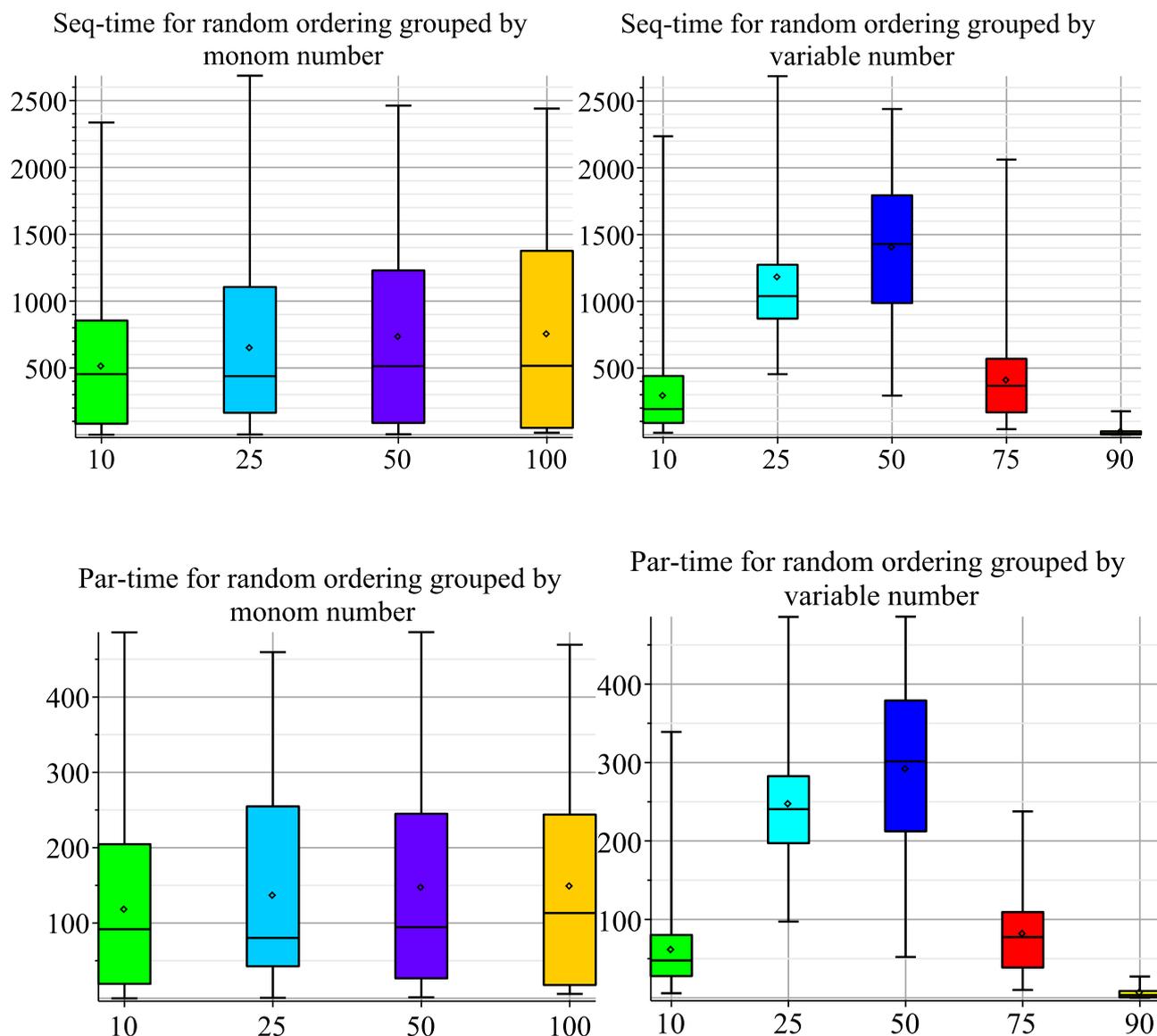


Seq-time for the optimal ordering grouped by monom number



Seq-time for optimal ordering grouped by variable number





**Рисунок 4.** Результаты тестирования первого набора полиномов при оптимальном и случайном упорядочивании переменных однопоточного и 8-поточного исполнения

**Таблица 1.** Характеристики декомпозиции для выходов схем

Бенчмарк	Схема	Тип	№ функции/имя	Кол-во аргумент.	Кол-во трив. комп.	Кол-во перемен в 1 комп.	Кол-во перемен во 2 комп.	Время декомпозиции, сек.
EPFL	router	or	1/po1	60	0	57	3	0.006
EPFL	mem_ctrl_part0	and	63/po067	113	74	33	6	0.048
ISCAS	s15850_part0	and	34/n560	144	4	115	25	0.034
ISCAS	s15850_part0	and	498/n450	146	1	141	4	0.383
ISCAS	s15850_part0	or	97/n2045	58	8	41	9	0.001
IWLS2005	wb_dma	or	314/n1369	39	0	29	10	0.001
IWLS93	i10_part0	or	135/po191	53	0	49	4	22.453
IWLS93	rot_part0	and	10/po11	45	1	42	2	52.187
LEKO	g625	and	0/po000	500	0	300	200	12632.000
LGSynth91	i2_part0	or	0/po0	201	3	132	69	8935.200
LGSynth91	i10_part0	and	79/po119	48	3	40	5	24.341
LGSynth91	pair_part0	or	4/po004	51	0	28	23	39.557
Other	sudoku_check_part2	and	1/n3454	729,	0	9	720	207620.000
QUIP	oc_minirisc	and	165/n1104	63	4	53	6	1576.636

Далее было решено настроить данный алгоритм для формата BDD, который является де-факто с середины 80-х годов является индустриальным стандартом в области синтеза логических схем. Задача декомпозиции является существенной компонентой этапа анализа булевых функций, составляющих схему, для целей ее оптимизации по времени срабатывания, размеров и энергопотребления. Для сравнения работы алгоритма на случайных функциях был реализован конвертор из текстового представления АНФ в BDD. Для тестирования реальных логических схем был выбран широко известный набор описаний схем в формате BLIF, представленный в работе [9]. Этот набор составлен на основании нескольких широко признанных индустриальных бенчмарков. Из 1314 файлов обработано 1118, для остальных анализ не удалось завершить в виду нехватки памяти. Для 1112 функций обнаружены конъюнктивные разложения. В таблице 1 приведены характеристики декомпозиции для выходов схем, имеющих «много» аргументов, и нетривиальные компоненты разложения (две компоненты имеют носитель, состоящий из не менее 2 переменных). Отметим, что количество входов больше 40 делает невозможным декомпозицию на основе полного перебора.

### 3. Заключение

В статье представлены результаты экспериментов по конъюнктивной и дизъюнктивной декомпозиции различных представлений булевых функций (ZDD, BDD, OKFDD) методами, которые получаются путем специализации общего алгоритма декомпозиции. Результаты тестирования подтверждают гипотезы о сложностных характеристиках алгоритма. Проведена оценка эффективности мемоизации.

### Список литературы

1. Bengtsson T., Martinelli A., Dubrova E., A fast heuristic algorithm for disjoint decomposition of Boolean functions // 11th IEEE/ACM International Workshop on Logic & Synthesis (IWLS'02). New Orleans, USA. 2002, PP. 51–55.
2. Bryant R.E. Graph-Based Algorithms for Boolean Function Manipulation // IEEE Transactions on Computers. 1986. Vol. C-35, № 8. P. 677-691.
3. Drechsler R. and Becker B. Ordered Kronecker functional decision diagrams – a data structure for representation and manipulation of Boolean functions // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 1998. Vol. 17, № 10. P. 965–973.
4. Emelyanov P. and Ponomaryov D. On the tractability of disjoint AND-decomposition of boolean formulas // PSI 2014: Ershov Informatics Conference / Lecture Notes in Computer Science. Berlin, New York: Springer. 2015. Vol. 8974. P. 92-101. DOI: 10.1007/978-3-662-46823-4\_8

5. Emelyanov P. and Ponomaryov D. Algorithmic issues of AND-decomposition of boolean formulas // Programming and Computer Software. 2015. Vol. 41, № 3. P. 162–169. DOI: 10.1134/S0361768815030032. Translated: Programmirovaniye, № 3, P. 62-72, 2015.
6. Emelyanov P. and Ponomaryov D. The Complexity of AND-decomposition of Boolean Functions // Discrete Applied Mathematics. 2020. Vol. 280. P. 113–132. DOI: 10.1016/j.dam.2019.07.005.
7. Minato S.-i. Zero-suppressed BDDs for set manipulation in combinatorial problems // 3d International Design Automation Conference (DAC'93). New York: ACM. 1993. P. 272–277.
8. Mishchenko A., Chatterjee S., and Brayton R. DAG-aware AIG rewriting a fresh look at combinational logic synthesis // 43rd annual Design Automation Conference (DAC '06). New York: ACM. 2006. P. 532–535. DOI: <https://doi.org/10.1145/1146909.1147048>.
9. P. Fišer, J. Schmidt, A comprehensive set of logic synthesis and optimization examples // 12th International Workshop on Boolean Problems (IWSBP'2016). Freiberg, Germany. 2016. P. 151–158.
10. Shpilka A. and Volkovich I. On the relation between polynomial identity testing and finding variable disjoint factors // 37th International Colloquium on Automata, Languages and Programming. Part 1 (ICALP'2010) / Lecture Notes in Computer Science. Berlin, New York: Springer, 2010. Vol. 6198. P. 408-419.
11. Somenzi F. CUDD: CU Decision Diagram Package. University of Colorado at Boulder. 2016. [Электронный ресурс]. URL: <https://github.com/ivmai/cudd> (дата обращения: 10.03.2022).

