

ISSN 2949-5598

ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ

Том 33 № 2 2026

Новосибирск
Издательство Института математики

О ВЫЧИСЛИТЕЛЬНОЙ СЛОЖНОСТИ НАХОЖДЕНИЯ ХРЕБТОВ БУЛЕВЫХ ФОРМУЛ

В. Ю. Попов

Уральский федеральный университет,
ул. Ленина, 51, 620083 Екатеринбург, Россия

E-mail: popovvvv@gmail.com

Аннотация. Хребты являются важными элементами решений ряда комбинаторных задач. В частности, хребты рассматривались для задачи коммивояжёра и различных вариантов задачи выполнимости. В данной статье рассматриваются хребты для задачи выполнимости. Хребты булевых формул представляют собой подмножества переменных, имеющие единственное допустимое значение. Существует ряд различных применений хребтов. Хребты широко используются для решения задачи выполнимости с помощью эвристических алгоритмов. В данной статье рассматривается вычислительная сложность задачи поиска хребта, а также задачи нахождения допустимых значений для хребтов. Библиогр. 38.

Ключевые слова: вычислительная сложность, задача выполнимости, NP-трудность, хребты.

Введение

Исследования пространств решений комбинаторных задач физическими методами выявили для некоторых задач феномен, называемый замораживанием переменных. В частности, наличие данного феномена обнаружено для задачи коммивояжёра в исследованиях, представленных в работе [1]. Замораживание переменных наблюдается и для задачи выполнимости [2–4].

Замораживание переменных проявляется в том, что некоторые части хороших наборов значений переменных остаются практически неизменными почти для всех хороших наборов значений переменных. Замораживание переменных указывает на закономерную неравномерность в пространстве решений: наличие областей повышенной плотности и значительных разреженных областей. Следует отметить, что многие современные криптографические алгоритмы в значительной степени опираются на разреженные или плотные данные (см., например, [5–7]). Это

естественным образом ведёт к разработке методов криптоанализа, которые направлены на поиск специфических закономерностей, связанных с разреженными и плотными данными [8–10]. В силу этого феномен замораживания переменных представляет существенный интерес как для криптографии, так и для криптоанализа.

Выявлено наличие тесной связи замораживания переменных с другим важным направлением теоретического исследования задачи выполнимости — фазовым переходом и пороговой гипотезой [11]. Явление фазового перехода указывает на то, что трудность задачи зависит не столько от размера исходных данных, сколько от некоторых их структурных особенностей. В последние годы исследования в области фазового перехода получили существенное развитие (см., например, [12]). В частности, пороговая гипотеза получила подтверждение как для задачи выполнимости [13, 14], так и в общем случае [15, 16].

Исследования в области фазового перехода показывают, что средние вычислительные затраты на поиск решения для задачи выполнимости растут линейно с увеличением числа переменных булевой формулы [17]. Однако для небольшой доли булевых формул наблюдается экспоненциальный рост затрат [17], что обусловлено присутствием замораживания переменных в булевых формулах [18]. Проявления феномена замораживания переменных для различных комбинаторных задач были формализованы на основе понятия хребта [18, 19].

От способности решателя учитывать структурные особенности булевых формул в значительной мере зависит его производительность [20]. Хребты рассматриваются как одна из основных структурных особенностей [21, 22]. Многие современные решатели используют поиск хребтов [23–25]. Экспериментально установленная зависимость между наличием хребта и трудностью анализа булевой формулы указывает на естественный путь к повышению производительности решателей: необходимо как можно раньше обнаружить хребет и элиминировать его из формулы, присваивая конкретные значения переменным. Данная работа посвящена вычислительной сложности соответствующих задач.

1. Основные определения

Как обычно, для произвольного множества M через $|M|$ будем обозначать его мощность, а для произвольной булевой формулы f через $V(f)$ — множество всех её переменных. Произвольный набор значений переменных из подмножества S множества $V(f)$ задаётся некоторым отображением $A: S \rightarrow \{0, 1\}$. Будем обозначать через $F[A(S)]$ булеву формулу, полученную из формулы f подстановкой вместо каждого вхождения переменной x её значения $A(x)$ для всех $x \in S$. Следуя [26], для булевой

формулы f множество $S \subseteq V(f)$ будем называть хребтом, если существует единственное отображение A такое, что формула $F[A(S)]$ выполнима при условии выполнимости формулы f .

2. Нахождение хребтов

Понятия и определения, которые не приводятся в настоящей статье непосредственно, можно найти в книге [27]. В [28] показано, что если $P \neq NP$, то не существует аппроксимационной процедуры, которая бы гарантировала нахождение достаточно большого хребта за полиномиальное время. Для произвольного множества булевых формул $\mathcal{H}$ и для любого действительного числа β , $0 < \beta < 1$, через $L(\mathcal{H}, \beta)$ обозначим язык, состоящий из всех булевых формул h , удовлетворяющих следующим условиям: $h \in \mathcal{H}$ и существует хребет $S \subseteq V(h)$ такой, что $|S| \geq \beta |V(h)|$. В [29] установлено, что для любого действительного числа β , $0 < \beta < 1$, существует генерируемое полиномиальным алгоритмом множество булевых формул $\mathcal{B}$ такое, что язык $L(\mathcal{B}, \beta)$ NP-полный. Результаты исследований [28, 29] указывают на то, что нахождение хребта может быть не менее трудной задачей, чем решение задачи выполнимости. Следующее утверждение показывает, что нахождение хребта может быть значительно труднее, чем решение задачи выполнимости.

Теорема 1. Для любого $\beta \in (0, 1)$ существует генерируемое полиномиальным алгоритмом множество булевых формул $\mathcal{F}$ такое, что язык $L(\mathcal{F}, \beta)$ DP-полный.

ДОКАЗАТЕЛЬСТВО. Для произвольной булевой формулы p обозначим через p^* булеву формулу, полученную из p заменой каждого вхождения переменной x на $\neg x$ для всех переменных $x \in V(p)$. Зафиксируем произвольное $\beta \in (0, 1)$. Пусть $\mathcal{F}$ — множество всевозможных булевых формул вида

$$\left(f \vee \bigwedge_{x \in W} x \right) \wedge (g \vee g^*), \quad (1)$$

удовлетворяющих следующим трём условиям:

- 1) булевы формулы f и g представлены в конъюнктивной нормальной форме, причём каждая дизъюнкция содержит по три литерала;
- 2) множества переменных $V(f)$, $V(g)$ и W попарно не пересекаются;
- 3) выполняются неравенства

$$|W| \geq \beta(|V(f)| + |V(g)| + |W|), \quad (2)$$

$$|V(f)| < \beta(|V(f)| + |V(g)| + |W|). \quad (3)$$

Легко понять, что множество булевых формул $\mathcal{F}$ рекурсивно. В частности, элементы множества $\mathcal{F}$ можно сгенерировать по их номерам. Заметим, что размер элементов множества $\mathcal{F}$ экспоненциально больше размера их естественных номеров в бинарной кодировке, поэтому для установления полиномиального соответствия между размерами входа и выхода при генерации естественно использовать стандартный в таких случаях приём, основанный на представлении номеров в унарной кодировке (см., например, [30] для случая уникальной задачи остановки). Легко понять, что по номерам, представленным в унарном виде, элементы множества $\mathcal{F}$ можно генерировать за полиномиальное время.

Доказательство DP-полноты языка $L(\mathcal{F}, \beta)$ построим на обосновании трудности распознавания принадлежности формул множества $\mathcal{F}$ языку $L(\mathcal{F}, \beta)$.

Непосредственно из определения хребта вытекает, что формула вида (1) имеет хребет лишь при условии, что формула $g \vee g^*$ выполняется. Выполнимость формулы $g \vee g^*$ равносильна выполнимости по крайней мере одной из формул g и g^* . Из определения формулы g^* следует, что $g[A(V(g))] = 1$ тогда и только тогда, когда $g^*[A^*(V(g))] = 1$, где $A(y) \neq A^*(y)$ для всех $y \in V(g)$. Следовательно, выполнимость формулы $g \vee g^*$ равносильна выполнимости формулы g . Кроме того, ни одна из переменных множества $V(g)$ не принадлежит хребту.

Допустим, что формула f выполнима. Легко понять, что если формула f выполнима, то ни одна из переменных множества W не принадлежит хребту: либо формула (1) не будет выполнимой, либо формула (1) выполняется для любых значений переменных из множества W , поэтому если формула f выполнима, то для любого хребта S справедливо неравенство

$$|S| \leq |V(f)|. \quad (4)$$

Из определения языка $L(\mathcal{H}, \beta)$ и системы неравенств (3) и (4) очевидным образом вытекает, что формула (1) не принадлежит языку $L(\mathcal{F}, \beta)$. Таким образом, чтобы формула (1) принадлежала языку $L(\mathcal{F}, \beta)$, необходимо, чтобы формула f не была выполнимой.

Рассмотрим следующую задачу.

Задача SAT-UNSAT (Выполнимость-невыполнимость). Даны булевы формулы f и g , каждая из которых представлена в конъюнктивной нормальной форме, причём каждая дизъюнкция содержит по три литерала. Верно ли, что формула g выполнима, а формула f выполнимой не является?

Хорошо известно, что задача SAT-UNSAT DP-полна (см., например, [27, теорема 17.1]). Легко понять, что если формула g выполнима, а формула f выполнимой не является, то формула (1) выполняется лишь

при условии, что все переменные множества W принимают значение 1, т. е. множество W является хребтом. Учитывая неравенство (2), формула (1) принадлежит языку $L(\mathcal{F}, \beta)$ тогда и только тогда, когда формула g выполнима, а формула f выполнимой не является. Легко понять, что для произвольных формул f и g , выбирая достаточно большое множество W , можно построить формулу вида (1) из множества $\mathcal{F}$. Отсюда и из DP-полноты задачи SAT-UNSAT очевидным образом вытекает DP-полнота языка $L(\mathcal{F}, \beta)$. Теорема 1 доказана.

Заметим, что обнаружение хребта позволяет существенно упростить анализ исходной формулы. Однако даже нахождение большого хребта не обеспечивает того, что полученная формула станет лёгкой для дальнейшего анализа, по крайней мере потому, что она тоже может содержать хребет. Тем самым представляет интерес исследование максимальных хребтов, т. е. хребтов S таких, что для любой переменной $x \notin S$ множество $S \cup \{x\}$ хребтом не является. Соответствующая задача в явном виде может быть сформулирована следующим образом.

Задача МВ (ПРОВЕРКА ХРЕБТА НА МАКСИМАЛЬНОСТЬ). *Даны булева формула f и хребет S формулы f . Верно ли, что хребет S максимален?*

Следующее утверждение показывает, что проверка найденного хребта на максимальность является вычислительно трудной задачей.

Теорема 2. *Задача МВ co-NP-трудна.*

ДОКАЗАТЕЛЬСТВО. Для произвольной булевой формулы g в конъюнктивной нормальной форме рассмотрим булеву формулу

$$\bigwedge_{x \in W} x \wedge ((g \vee g^*) \vee z), \quad (5)$$

предполагая, что z — переменная такая, что $z \notin W$, $z \notin V(g)$, а множества W и $V(g)$ не имеют общих элементов. Легко понять, что для любой формулы g формула (5) выполнима. Следовательно, множество W является хребтом. Кроме того, ни одна переменная множества $V(g)$ в хребет не входит. Следовательно, максимальным хребтом является либо множество W , либо $W \cup \{z\}$. Очевидно, что $W \cup \{z\}$ — максимальный хребет тогда и только тогда, когда булева формула g невыполнима. Отсюда в силу co-NP-полноты задачи SAT-COMPLEMENT (см., например, [27, п. 10.1]) вытекает co-NP-трудность проверки хребта на максимальность. Теорема 2 доказана.

Хотя теоремы 1 и 2 мы доказываем конструктивно, т. е. строим соответствующие множества булевых формул в явном виде, результаты этих

утверждений относятся к количественной оценке хребтов: существует ли достаточно большой хребет и можно ли известный хребет увеличить. Результаты работы [31] показывают, что, даже зная хребет, найти значения его переменных — вычислительно трудная задача. Количественные оценки ещё менее информативны. В частности, даже зная мощность максимального хребта, мы, вообще говоря, не имеем эффективного способа нахождения хребта.

Теорема 3. Если $P \neq NP \cap \text{co-NP}$, то для любого натурального k существует множество булевых формул $\mathcal{F}$ такое, что каждая формула $f \in \mathcal{F}$ имеет максимальный хребет мощности k , но не существует полиномиального алгоритма, позволяющего по произвольной формуле $f \in \mathcal{F}$ найти её хребет.

ДОКАЗАТЕЛЬСТВО. При доказательстве теоремы будем пользоваться подходом, предложенным в [32] и адаптированным для исследования хребтов в [33]. Предположим, что $P \neq NP \cap \text{co-NP}$. Тогда существует язык L в некотором алфавите Σ такой, что $L \in (NP \cap \text{co-NP}) \setminus P$. Поскольку $L \in \text{co-NP}$, язык $\Sigma^* \setminus L$, где Σ^* — множество всевозможных слов в алфавите Σ , принадлежит классу NP . Зафиксируем некоторое рекурсивное множество

$$\{\mathfrak{N}_i \mid i \in \mathbb{N}\} \quad (6)$$

недетерминированных машин Тьюринга, распознающих всевозможные языки из класса NP за полиномиальное время. Обычно (см., например, [34–36]) в качестве множества (6) используются недетерминированные машины Тьюринга с часами, предложенные в работе [37]. Поскольку $L \in NP$ и $\Sigma^* \setminus L \in NP$, в множестве (6) найдутся машины $\mathfrak{N}_s$ и $\mathfrak{N}_t$, распознающие языки L и $\Sigma^* \setminus L$ соответственно. Используя конструкцию Кука [38], можем получить множество булевых формул

$$\{f(\mathfrak{N}_i, w) \mid i \in \mathbb{N}, w \in \Sigma^*\} \quad (7)$$

такое, что формула $f(\mathfrak{N}_i, w)$ выполнима тогда и только тогда, когда слово w принадлежит языку, распознаваемому машиной $\mathfrak{N}_i$. Пусть $\mathcal{F}$ — подмножество множества (7), состоящее из формул вида

$$\left(\bigwedge_{x \in U} x \vee f(\mathfrak{N}_s, w) \vee f^*(\mathfrak{N}_s, w) \right) \wedge \left(\bigwedge_{y \in W} y \vee f(\mathfrak{N}_t, w) \vee f^*(\mathfrak{N}_t, w) \right) \quad (8)$$

для всех $w \in \Sigma^*$, где множества $U, W, V(f(\mathfrak{N}_s, w)), V(f(\mathfrak{N}_t, w))$ попарно не пересекаются, $|U| = |W| = k$.

Заметим, что формулы $f(\mathfrak{N}_s, w)$ и $f(\mathfrak{N}_t, w)$ выполнимы тогда и только тогда, когда выполнимы формулы $f^*(\mathfrak{N}_s, w)$ и $f^*(\mathfrak{N}_t, w)$ соответственно, поэтому переменные из множеств $V(f(\mathfrak{N}_s, w))$ и $V(f(\mathfrak{N}_t, w))$ не принадлежат хребту. Поскольку машина $\mathfrak{N}_s$ распознаёт язык L , а машина $\mathfrak{N}_t$ —

дополнение языка L , формула $f(\mathfrak{N}_s, w)$ выполнима тогда и только тогда, когда формула $f(\mathfrak{N}_t, w)$ выполнимой не является.

Допустим, что формула $f(\mathfrak{N}_s, w)$ выполнима. Тогда переменные множества U хребту не принадлежат, а множество W — максимальный хребет формулы (8). Если формула $f(\mathfrak{N}_t, w)$ выполнима, то переменные множества W хребту не принадлежат, а множество U является максимальным хребтом формулы (8). Таким образом, если $w \in L$, то множество W является максимальным хребтом формулы (8), а если $w \notin L$, то множество U является максимальным хребтом формулы (8). Поскольку $L \notin \mathbf{P}$, не существует полиномиального алгоритма, позволяющего по произвольной формуле $f \in \mathcal{F}$ найти её хребет. Теорема 3 доказана.

3. Нахождение значений хребтов

При анализе булевой формулы f нахождение хребта S само по себе позволяет получить важную информацию о структуре формулы f и пространстве решений. Однако для упрощения дальнейшего анализа формулы f необходимо нахождение значения хребта S , т. е. конкретных значений переменных из множества S , обеспечивающих выполнимость формулы f , что само по себе является вычислительно трудной задачей (см. [31]). Следующее утверждение является аналогом результатов работы [31] для случая максимальных хребтов.

Теорема 4. Если $\mathbf{P} \neq \mathbf{NP} \cap \text{co-NP}$, то для любого натурального k существует множество булевых формул $\mathcal{F}$ такое, что для каждой формулы $f \in \mathcal{F}$ известен максимальный хребет мощности k , но не существует полиномиального алгоритма, позволяющего по произвольной формуле $f \in \mathcal{F}$ найти значения хребта.

ДОКАЗАТЕЛЬСТВО осуществим модификацией рассуждений, проведённых при доказательстве теоремы 3. Рассмотрим множество $\mathcal{F}$, предполагая, что оно является подмножеством множества (7), состоящим из формул вида

$$\left(\bigwedge_{x \in U} x \vee f(\mathfrak{N}_s, w) \vee f^*(\mathfrak{N}_s, w) \right) \wedge \left(\bigwedge_{x \in U} \neg x \vee f(\mathfrak{N}_t, w) \vee f^*(\mathfrak{N}_t, w) \right) \quad (9)$$

для всех $w \in \Sigma^*$, где множества U , $V(f(\mathfrak{N}_s, w))$, $V(f(\mathfrak{N}_t, w))$ попарно не пересекаются, $|U| = k$. Легко понять, что множество U является максимальным хребтом формулы (9). Если $w \in L$, то формула (9) выполняется, когда все переменные множества U принимают значение 0. Если $w \notin L$, то формула (9) выполняется, когда все переменные множества U принимают значение 1. Поскольку $L \notin \mathbf{P}$, не существует полиномиального алгоритма, позволяющего по произвольной формуле $f \in \mathcal{F}$ найти значения хребта. Теорема 4 доказана.

Заключение

В данной статье рассмотрена вычислительная сложность задач, связанных с использованием хребтов при анализе булевых формул: существование хребта, нахождение хребта и нахождение значений хребта. Ранее уже было известно, что исследование булевых формул на основе хребтов может быть не менее трудной задачей, чем нахождение набора, на котором формула выполняется [28, 29, 33]. Мы показали, что подход с использованием хребтов может быть значительно более трудным, чем задача выполнимости.

Финансирование работы

Исследование выполнено за счёт бюджета Уральского федерального университета. Дополнительных грантов на проведение или руководство этим исследованием получено не было.

Конфликт интересов

Автор заявляет, что у него нет конфликта интересов.

Литература

1. **Kirkpatrick S., Toulouse G.** Configuration space analysis of travelling salesman problems // J. Phys. 1985. V. 45, No. 8. P. 1277–1292. DOI: 10.1051/jphys:019850046080127700.
2. **Monasson R., Zecchina R.** Entropy of the K -satisfiability problem // Phys. Rev. Lett. 1996. V. 76, No. 21. P. 3881–3885. DOI: 10.1103/PhysRevLett.76.3881.
3. **Monasson R., Zecchina R.** Statistical mechanics of the random K -satisfiability model // Phys. Rev. E. 1997. V. 56, No. 2. P. 1357–1370. DOI: 10.1103/PhysRevE.56.1357.
4. **Monasson R., Zecchina R., Kirkpatrick S., Selman B., Troyansky L.** 2+ p -SAT: Relation of typical-case complexity to the nature of the phase transition // Random Struct. Algorithms. 1999. V. 15, No. 3–4. P. 414–435. DOI: 10.1002/(SICI)1098-2418(199910/12)15:3/4<414::AID-RSA10>3.0.CO;2-G.
5. **Bohra D. D., Banerjee D. S., Sanadhya S.** Sparse-aware NTT: Accelerating lattice-based cryptography on FPGAs // 2025 IEEE Comput. Soc. Annu. Symp. Very Large Scale Integration (Kalamata, Greece, July 6–9, 2025). Piscataway: IEEE, 2025. Article ID 11130302. 6 p. DOI: 10.1109/ISVLSI65124.2025.11130302.
6. **Dao Q., Jain A.** Lossy cryptography from code-based assumptions dense-sparse LPN: A new subexponentially hard LPN variant in SZK // J. Cryptol. 2025. V. 38, No. 32. P. 1–46. DOI: 10.1007/s00145-025-09553-6.

7. **He P., Tu Y., Bao T., Koç Ç. Ç., Xie J.** HSPA: High-throughput sparse polynomial multiplication for code-based post-quantum cryptography // *ACM Trans. Embed. Comput. Syst.* 2024. V. 24, No. 1. Article ID 16. 24 p. DOI: 10.1145/3703837.
8. **Campa L., Roy A.** Gröbner basis cryptanalysis of Anemoi // *Advances in cryptology — EUROCRYPT 2025. Proc. 44th Annu. Int. Conf. Theory and Applications of Cryptographic Techniques* (Madrid, Spain, May 4–8, 2025). Pt. 1. Cham: Springer, 2025. P. 303–332. (Lect. Notes Comput. Sci.; V. 15601). DOI: 10.1007/978-3-031-91107-1_11.
9. **Dinur I., Keller N., Klein O.** Fine-grained cryptanalysis: Tight conditional bounds for dense k -SUM and k -XOR // *J. ACM.* 2024. V. 71, No. 3. Article ID 23. 41 p. DOI: 10.1145/3653014.
10. **Jain A., Lin H., Saha S.** A systematic study of sparse LWE // *Advances in cryptology — CRYPTO 2024. Proc. 44th Annu. Int. Cryptology Conf.* (Santa Barbara, USA, Aug. 18–22, 2024). Pt. 3. Cham: Springer, 2024. P. 210–245. (Lect. Notes Comput. Sci.; V. 14922). DOI: 10.1007/978-3-031-68382-4_7.
11. **Bollobás B., Borgs C., Chayes J. T., Kim J. H., Wilson D. B.** The scaling window of the 2-SAT transition // *Random Struct. Algorithms.* 2001. V. 18, No. 3. P. 201–256. DOI: 10.1002/rsa.1006.
12. **Perkins W.** Searching for (sharp) thresholds in random structures: Where are we now? // *Bull. Amer. Math. Soc.* 2025. V. 62, No. 1. P. 113–143. DOI: 10.1090/bull/1857.
13. **Ding J., Sly A., Sun N.** Satisfiability threshold for random regular NAE-SAT // *Commun. Math. Phys.* 2016. V. 341. P. 435–489. DOI: 10.1007/s00220-015-2492-8.
14. **Ding J., Sly A., Sun N.** Proof of the satisfiability conjecture for large k // *Ann. Math.* 2022. V. 196, No. 1. P. 1–388. DOI: 10.4007/annals.2022.196.1.1.
15. **Park J., Pham H. T.** A proof of the Kahn–Kalai conjecture // *J. Amer. Math. Soc.* 2024. V. 37. P. 235–243. DOI: 10.1090/jams/1028.
16. **Park B., Vondrák J.** A simple proof of the nonuniform Kahn–Kalai conjecture // *SIAM J. Discrete Math.* 2024. V. 38, No. 3. P. 2060–2073. DOI: 10.1137/23M1587075.
17. **Selman B., Kirkpatrick S.** Critical behavior in the computational cost of satisfiability testing // *Artif. Intell.* 1996. V. 81, No. 1–2. P. 273–295. DOI: 10.1016/0004-3702(95)00056-9.
18. **Monasson R., Zecchina R., Kirkpatrick S., Selman B., Troyansky L.** Determining computational complexity from characteristic “phase transitions” // *Nature.* 1999. V. 400. P. 133–137. DOI: 10.1038/22055.
19. **Schneider J., Froschhammer C., Morgenstern I., Husslein T., Singer J. M.** Searching for backbones — An efficient parallel algorithm for the traveling salesman problem // *Comput. Phys. Commun.* 1996. V. 96, No. 2–3. P. 173–188. DOI: 10.1016/0010-4655(96)00062-8.

20. **Alyahya T. N., Menai M. E. B., Mathkour H.** On the structure of the Boolean satisfiability problem: A survey // *ACM Comput. Surv.* 2022. V. 55, No. 3. Article ID 46. 34 p. DOI: 10.1145/3491210.
21. **Dubois O., Dequen G.** A backbone-search heuristic for efficient solving of hard 3-SAT formulae // *Proc. 17th Int. Jt. Conf. Artificial Intelligence* (Seattle, USA, Aug. 4–10, 2001). V. 1. San Francisco: Morgan Kaufmann Publ., 2001. P. 248–253.
22. **Menai M. E. B., Batouche M.** A backbone-based co-evolutionary heuristic for partial MAX-SAT // *Artificial evolution. Rev. Sel. Pap. 7th Int. Conf.* (Lille, France, Oct. 26–28, 2005). Heidelberg: Springer, 2006. P. 155–166. (Lect. Notes Comput. Sci.; V. 3871). DOI: 10.1007/11740698_14.
23. **Biere A., Faller T., Fazekas K., Fleury M., Froleys N., Pollitt F.** CaDiCaL 2.0 // *Computer aided verification. Proc. 36th Int. Conf.* (Montreal, Canada, July 24–27, 2024). Pt. 1. Cham: Springer, 2024. P. 133–152. (Lect. Notes Comput. Sci.; V. 14681). DOI: 10.1007/978-3-031-65627-9_7.
24. **Froleys N., Yu E., Biere A.** BIG backbones // *Proc. 23rd Conf. Formal Methods in Computer-Aided Design* (Ames, USA, Oct. 23–27, 2023). Wien: TU Wien Acad. Press, 2023. P. 162–167. DOI: 10.34727/2023/isbn.978-3-85448-060-0_24.
25. **Schreiber D., Rigi-Luperti N., Biere A.** Streamlining distributed SAT solver design // *Proc. 28th Int. Conf. Theory and Applications of Satisfiability Testing* (Glasgow, UK, Aug. 12–15, 2025). Wadern: Leibniz-Zentrum Inform., 2025. P. 27:1–27:23. (Leibniz Int. Proc. Inform.; V. 341). DOI: 10.4230/LIPIcs.SAT.2025.27.
26. **Williams R., Gomes C. P., Selman B.** Backdoors to typical case complexity // *Proc. 18th Int. Jt. Conf. Artificial Intelligence* (Acapulco, Mexico, Aug. 9–15, 2003). San Francisco: Morgan Kaufmann Publ., 2003. P. 1173–1178.
27. **Papadimitriou C. H.** Computational complexity. Reading, MA: Addison-Wesley Publ., 1994. 524 p.
28. **Kilby P., Slaney J., Thiebaux S., Walsh T.** Backbones and backdoors in satisfiability // *Proc. 20th Nat. Conf. Artificial Intelligence* (Pittsburgh, USA, July 9–13, 2005). Washington: AAAI Press, 2005. P. 1368–1373.
29. **Hemaspaandra L. A., Narváez D. E.** Existence versus exploitation: The opacity of backdoors and backbones // *Prog. Artif. Intell.* 2021. V. 10. P. 297–308. DOI: 10.1007/s13748-021-00234-6.
30. **Blass A., Gurevich Y.** On the unique satisfiability problem // *Inf. Control.* 1982. V. 55, No. 1–3. P. 80–88. DOI: 10.1016/S0019-9958(82)90439-9.
31. **Hemaspaandra L. A., Narváez D. E.** The opacity of backbones // *Proc. 31st AAAI Conf. Artificial Intelligence* (San Francisco, USA, Feb. 4–9, 2017). Washington: AAAI Press, 2017. P. 3900–3906.
32. **Borodin A. B., Demers A. J.** Some comments on functional self-reducibility and the NP hierarchy: Tech. rep. Ithaca, NY: Cornell Univ., 1976. 22 p.

-
33. **Hemaspaandra L. A., Narváez D. E.** The opacity of backbones // *Inf. Comput.* 2021. V. 281. Article ID 104772. 10 p. DOI: 10.1016/j.ic.2021.104772.
34. **Landweber L. H., Lipton R. J., Robertson E. L.** On the structure of sets in NP and other complexity classes // *Theor. Comput. Sci.* 1981. V. 15, No. 2. P. 181–200. DOI: 10.1016/0304-3975(81)90069-4.
35. **Hartmanis J., Immerman N.** On complete problems for $NP \cap CoNP$ // *Automata, languages and programming. Proc. 12th Int. Colloq. (Nafplion, Greece, July 15–19, 1985)*. Heidelberg: Springer, 1985. P. 250–259. (Lect. Notes Comput. Sci.; V. 194). DOI: 10.1007/BFb0015750.
36. **Hartmanis J., Hemachandra L. A.** Complexity classes without machines: On complete languages for UP // *Theor. Comput. Sci.* 1988. V. 58, No. 1–3. P. 129–142. DOI: 10.1016/0304-3975(88)90022-9.
37. **Baker T., Gill J., Solovay R.** Relativizations of the $P = ? NP$ question // *SIAM J. Comput.* 1975. V. 4, No. 4. P. 431–442. DOI: 10.1137/0204037.
38. **Cook S. A.** The complexity of theorem-proving procedures // *Proc. 3rd Annu. ACM Symp. Theory of Computing (Shaker Heights, USA, May, 3–5, 1971)*. New York: ACM, 1971. P. 151–158. DOI: 10.1145/800157.805047.

Попов Владимир Юрьевич

Статья поступила

5 ноября 2025 г.

После доработки —

11 февраля 2026 г.

Принята к публикации

23 марта 2026 г.

ON COMPUTATIONAL COMPLEXITY OF FINDING BACKBONES OF BOOLEAN FORMULAS

V. Yu. Popov

Ural Federal University,
51 Lenin Street, 620083 Yekaterinburg, Russia
E-mail: popovvvv@gmail.com

Abstract. Backbones are important parts of solutions for a number of combinatorial problems. In particular, backbones have been considered for the traveling salesman problem and various modifications of the satisfiability problem. In this paper, we consider backbones for the latter. The backbones of Boolean formulas are subsets of variables with unique proper assignment. There is a number of different applications of backbones. They are extensively used to solve the satisfiability problem with heuristic algorithms. In this paper, we consider the computational complexity of the problem of finding backbones and of the problem of finding proper assignments for backbones. Bibliogr. 38.

Keywords: computational complexity, satisfiability problem, NP-hardness, backbones.

References

1. **S. Kirkpatrick** and **G. Toulouse**, Configuration space analysis of travelling salesman problems, *J. Phys.* **45** (8), 1277–1292 (1985), DOI: 10.1051/jphys:019850046080127700.
2. **R. Monasson** and **R. Zecchina**, Entropy of the K -satisfiability problem, *Phys. Rev. Lett.* **76** (21), 3881–3885 (1996), DOI: 10.1103/PhysRevLett.76.3881.
3. **R. Monasson** and **R. Zecchina**, Statistical mechanics of the random K -satisfiability model, *Phys. Rev. E* **56** (2), 1357–1370 (1997), DOI: 10.1103/PhysRevE.56.1357.
4. **R. Monasson**, **R. Zecchina**, **S. Kirkpatrick**, **B. Selman**, and **L. Troyansky**, $2+p$ -SAT: Relation of typical-case complexity to the nature of the phase transition, *Random Struct. Algorithms* **15** (3–4), 414–435 (1999), DOI: 10.1002/(SICI)1098-2418(199910/12)15:3/4<414::AID-RSA10>3.0.CO;2-G.

English transl.: *Journal of Applied and Industrial Mathematics* **20** (2) (2026).

5. **D. D. Bohra, D. S. Banerjee, and S. Sanadhya**, Sparse-aware NTT: Accelerating lattice-based cryptography on FPGAs, in *2025 IEEE Comput. Soc. Annu. Symp. Very Large Scale Integration* (Kalamata, Greece, July 6–9, 2025) (IEEE, Piscataway, 2025), ID 11130302, DOI: 10.1109/ISVLSI65124.2025.11130302.
6. **Q. Dao and A. Jain**, Lossy cryptography from code-based assumptions dense-sparse LPN: A new subexponentially hard LPN variant in SZK, *J. Cryptol.* **38** (32), 1–46 (2025), DOI: 10.1007/s00145-025-09553-6.
7. **P. He, Y. Tu, T. Bao, Ç. Ç. Koç and J. Xie**, HSPA: High-throughput sparse polynomial multiplication for code-based post-quantum cryptography, *ACM Trans. Embed. Comput. Syst.* **24** (1), ID 16 (2024), DOI: 10.1145/3703837.
8. **L. Campa and A. Roy**, Gröbner basis cryptanalysis of Anemoi, in *Advances in Cryptology — EUROCRYPT 2025*, Proc. 44th Annu. Int. Conf. Theory and Applications of Cryptographic Techniques (Madrid, Spain, May 4–8, 2025), Pt. 1 (Springer, Cham, 2025), pp. 303–332 (Lect. Notes Comput. Sci., Vol. 15601), DOI: 10.1007/978-3-031-91107-1_11.
9. **I. Dinur, N. Keller, and O. Klein**, Fine-grained cryptanalysis: Tight conditional bounds for dense k -SUM and k -XOR, *J. ACM* **71** (3), ID 23 (2024), DOI: 10.1145/3653014.
10. **A. Jain, H. Lin, and S. Saha**, A systematic study of sparse LWE, in *Advances in Cryptology — CRYPTO 2024*, Proc. 44th Annu. Int. Cryptology Conf. (Santa Barbara, USA, Aug. 18–22, 2024), Pt. 3 (Springer, Cham, 2024), pp. 210–245 (Lect. Notes Comput. Sci., Vol. 14922), DOI: 10.1007/978-3-031-68382-4_7.
11. **B. Bollobás, C. Borgs, J. T. Chayes, J. H. Kim, and D. B. Wilson**, The scaling window of the 2-SAT transition, *Random Struct. Algorithms* **18** (3), 201–256 (2001), DOI: 10.1002/rsa.1006.
12. **W. Perkins**, Searching for (sharp) thresholds in random structures: Where are we now?, *Bull. Amer. Math. Soc.* **62** (1), 113–143 (2025), DOI: 10.1090/bull/1857.
13. **J. Ding, A. Sly, and N. Sun**, Satisfiability threshold for random regular NAE-SAT, *Commun. Math. Phys.* **341**, 435–489 (2016), DOI: 10.1007/s00220-015-2492-8.
14. **J. Ding, A. Sly, and N. Sun**, Proof of the satisfiability conjecture for large k , *Ann. Math.* **196** (1), 1–388 (2022), DOI: 10.4007/annals.2022.196.1.1.
15. **J. Park and H. T. Pham**, A proof of the Kahn–Kalai conjecture, *J. Amer. Math. Soc.* **37**, 235–243 (2024), DOI: 10.1090/jams/1028.
16. **B. Park and J. Vondrák**, A simple proof of the nonuniform Kahn–Kalai conjecture, *SIAM J. Discrete Math.* **38** (3), 2060–2073 (2024), DOI: 10.1137/23M1587075.
17. **B. Selman and S. Kirkpatrick**, Critical behavior in the computational cost of satisfiability testing, *Artif. Intell.* **81** (1–2), 273–295 (1996), DOI: 10.1016/0004-3702(95)00056-9.

18. **R. Monasson, R. Zecchina, S. Kirkpatrick, B. Selman, and L. Troyansky**, Determining computational complexity from characteristic “phase transitions”, *Nature* **400**, 133–137 (1999), DOI: 10.1038/22055.
19. **J. Schneider, C. Froschhammer, I. Morgenstern, T. Husslein, and J. M. Singer**, Searching for backbones — An efficient parallel algorithm for the traveling salesman problem, *Comput. Phys. Commun.* **96** (2–3), 173–188 (1996), DOI: 10.1016/0010-4655(96)00062-8.
20. **T. N. Alyahya, M. E. B. Menai, and H. Mathkour**, On the structure of the Boolean satisfiability problem: A survey, *ACM Comput. Surv.* **55** (3), ID 46 (2022), DOI: 10.1145/3491210.
21. **O. Dubois and G. Dequen**, A backbone-search heuristic for efficient solving of hard 3-SAT formulae, in *Proc. 17th Int. Jt. Conf. Artificial Intelligence* (Seattle, USA, Aug. 4–10, 2001), Vol. 1 (Morgan Kaufmann Publ., San Francisco, 2001), pp. 248–253.
22. **M. E. B. Menaï and M. Batouche**, A backbone-based co-evolutionary heuristic for partial MAX-SAT, in *Artificial Evolution*, Rev. Sel. Pap. 7th Int. Conf. (Lille, France, Oct. 26–28, 2005) (Springer, Heidelberg, 2006), pp. 155–166 (Lect. Notes Comput. Sci., Vol. 3871), DOI: 10.1007/11740698_14.
23. **A. Biere, T. Faller, K. Fazekas, M. Fleury, N. Froleys, and F. Pollitt**, CaDiCaL 2.0, in *Computer Aided Verification*, Proc. 36th Int. Conf. (Montreal, Canada, July 24–27, 2024), Pt. 1 (Springer, Cham, 2024), pp. 133–152 (Lect. Notes Comput. Sci., Vol. 14681), DOI: 10.1007/978-3-031-65627-9_7.
24. **N. Froleys, E. Yu, and A. Biere**, BIG backbones, in *Proc. 23rd Conf. Formal Methods in Computer-Aided Design* (Ames, USA, Oct. 23–27, 2023) (TU Wien Acad. Press, Wien, 2023), pp. 162–167, DOI: 10.34727/2023/isbn.978-3-85448-060-0_24.
25. **D. Schreiber, N Rigi-Luperti, and A. Biere**, Streamlining distributed SAT solver design, in *Proc. 28th Int. Conf. Theory and Applications of Satisfiability Testing* (Glasgow, UK, Aug. 12–15, 2025) (Leibniz-Zentrum Inform., Wadern, 2025), pp. 27:1–27:23 (Leibniz Int. Proc. Inform., Vol. 341), DOI: 10.4230/LIPIcs.SAT.2025.27.
26. **R. Williams, C. P. Gomes, and B. Selman**, Backdoors to typical case complexity, in *Proc. 18th Int. Jt. Conf. Artificial Intelligence* (Acapulco, Mexico, Aug. 9–15, 2003) (Morgan Kaufmann Publ., San Francisco, 2003), pp. 1173–1178.
27. **C. H. Papadimitriou**, *Computational Complexity* (Addison-Wesley Publ., Reading, MA, 1994).
28. **P. Kilby, J. Slaney, S. Thiebaux, and T. Walsh**, Backbones and backdoors in satisfiability, in *Proc. 20th Nat. Conf. Artificial Intelligence* (Pittsburgh, USA, July 9–13, 2005) (AAAI Press, Washington, 2005), pp. 1368–1373.
29. **L. A. Hemaspaandra and D. E. Narváez**, Existence versus exploitation: The opacity of backdoors and backbones, *Prog. Artif. Intell.* **10**, 297–308 (2021), DOI: 10.1007/s13748-021-00234-6.

-
- 30. **A. Blass** and **Y. Gurevich**, On the unique satisfiability problem, *Inf. Control.* **55** (1–3), 80–88 (1982), DOI: 10.1016/S0019-9958(82)90439-9.
 - 31. **L. A. Hemaspaandra** and **D. E. Narváez**, The opacity of backbones, in *Proc. 31st AAAI Conf. Artificial Intelligence* (San Francisco, USA, Feb. 4–9, 2017) (AAAI Press, Washington, 2017), pp. 3900–3906.
 - 32. **A. B. Borodin** and **A. J. Demers**, Some comments on functional self-reducibility and the NP hierarchy, *Tech. Rep.* (Cornell Univ., Ithaca, NY, 1976).
 - 33. **L. A. Hemaspaandra** and **D. E. Narváez**, The opacity of backbones, *Inf. Comput.* **281**, ID 104772 (2021), DOI: 10.1016/j.ic.2021.104772.
 - 34. **L. H. Landweber**, **R. J. Lipton**, and **E. L. Robertson**, On the structure of sets in NP and other complexity classes, *Theor. Comput. Sci.* **15** (2), 181–200 (1981), DOI: 10.1016/0304-3975(81)90069-4.
 - 35. **J. Hartmanis** and **N. Immerman**, On complete problems for $NP \cap CoNP$, in *Automata, Languages and Programming*, Proc. 12th Int. Colloq. (Nafplion, Greece, July 15–19, 1985) (Springer, Heidelberg, 1985), pp. 250–259 (Lect. Notes Comput. Sci., Vol. 194), DOI: 10.1007/BFb0015750.
 - 36. **J. Hartmanis** and **L. A. Hemachandra**, Complexity classes without machines: On complete languages for UP, *Theor. Comput. Sci.* **58** (1–3), 129–142 (1988), DOI: 10.1016/0304-3975(88)90022-9.
 - 37. **T. Baker**, **J. Gill**, and **R. Solovay**, Relativizations of the $P = ? NP$ question, *SIAM J. Comput.* **4** (4), 431–442 (1975), DOI: 10.1137/0204037.
 - 38. **S. A. Cook**, The complexity of theorem-proving procedures, in *Proc. 3rd Annu. ACM Symp. Theory of Computing* (Shaker Heights, USA, May, 3–5, 1971) (ACM, New York, 1971), pp. 151–158, DOI: 10.1145/800157.805047.

Vladimir Yu. Popov

Received November 5, 2025

Revised February 11, 2026

Accepted March 23, 2026