ИНФОРМАТИКА

УДК 004.89 DOI: https://doi.org/10.54341/20778708_2025_2_63_97
EDN: FGHDOK

ДИСКРЕТНАЯ ОПТИМИЗАЦИЯ ДЛЯ ЗАДАЧИ ФАКТОРИЗАЦИИ

А.А. Чагочкин

Гомельский государственный университет имени Франциска Скорины

DISCRETE OPTIMIZATION FOR THE FACTORIZATION PROBLEM

A.A. Chagochkin

Francisk Skorina Gomel State University

Аннотация. Рассмотрена задача факторизации натуральных чисел на простые множители в контексте дискретной оптимизации и машинного обучения. Предложен подход с разложением на слагаемые и связанная функция для использования с генетическими алгоритмами (в качестве фитнес функции) и нейронными сетями (в качестве функции ошибки). Проведен статистический анализ изменений функции дискретного преобразования оптимального делителя с целью аппроксимации области оптимальных дискретных преобразований для пробного делителя.

Ключевые слова: *факторизация больших чисел*, *дискретная оптимизация*, *машинное обучение*, *нейронные сети*, *генетические алгоритмы.*

Для цитирования: *Чагочкин*, *А.А*. Дискретная оптимизация для задачи факторизации / А.А. Чагочкин // Проблемы физики, математики и техники. – 2025. – № 2 (63). – С. 97–100. – DOI: https://doi.org/10.54341/20778708_2025_2_63_97. – EDN: FGHDOK

Abstract. The paper addresses the factorization of natural numbers into prime factors task in the context of discrete optimization and machine learning. The approach with decomposition into summands and the associated function for using with genetic algorithms (as fitness function) and neural networks (as error function) is proposed. The statistical analysis of changes in the discrete transformation function of the optimal divisor is performed in order to approximate the scope of optimal discrete transformations for a trial divisor.

Keywords: *large numbers factoring*, *discrete optimization*, *machine learning*, *neural networks*, *genetic algorithms.*

For citation: *Chagochkin*, *A.A*. Discrete optimization for the factorization problem / A.A. Chagochkin // Problems of Physics, Mathematics and Technics. – 2025. – № 2 (63). – P. 97–100. – DOI: https://doi.org/10.54341/20778708_2025_2_63_97 (in Russian). – EDN: FGHDOK

Введение

Задача факторизации больших чисел имеет критическую значимость для современной цифровой инфраструктуры, особенно в вычислительной математике и криптографии.

Общий (General Number Field Sieve, GNFS) и специальный (Special Number Field Sieve, SNFS) методы решета числового поля признаны самыми быстрыми методами факторизации натуральных чисел [1], если не считать теоретические оценки квантового алгоритма Шора [2]. Время факторизации десятичного числа, состоящего из 512 знаков с использованием GNFS исчисляется годами и десятками лет для современных вычислительных кластеров, что представляет собой одну из фундаментальных проблем теории чисел на сегодняшний день [3]. Факторизация больших натуральных чисел с использованием GNFS имеет экспоненциальную алгоритмическую сложность [4]

$$O\left(\exp\left(\frac{64}{9}\ln N\right)^{\frac{1}{3}}\left(\ln\ln N\right)^{\frac{2}{3}}\right)$$

и не выполнима в приемлемое время – часы или дни для больших чисел. Поиск альтернативных подходов, позволяющих существенно снизить сложность вычислений, является важной практической задачей.

В данной работе факторизация рассматривается как задача дискретной оптимизации: приведение делителей факторизуемого числа к дискретному виду, то есть к натуральным целым числам, с использованием средств машинного обучения. При делении факторизуемого числа на оптимальный делитель дробный остаток отсутствует.

Задачу целочисленной факторизации числа в контексте дискретной оптимизации в простейшем виде можно интерпретировать как размещение прямоугольника на дискретной сетке, где площадь прямоугольника соответствует факторизуемому числу, а стороны – целочисленным факторам или делителям. В начале процесса оптимизации мы имеем первый пробный не оптимальный натуральный делитель, из которого вычисляется второй действительный путем деления факторизуемого числа на первый. В действительном

делителе выделяется целая и дробная части. Цель оптимизации: минимизировать дробную часть при соблюдении исходного ограничения: отличие делителей от 1 и самого факторизуемого числа. При таком наборе ограничений детерминированные методы оптимизации не могут предоставить значимые результаты для решения проблемы.

Для уменьшения неопределенности процесса оптимизации предлагается разложение факторизуемого числа на слагаемые. При делении слагаемых факторизуемого числа на оптимальный делитель сумма дробных частей остатков от деления также дискретна, то есть является целым числом.

Разложение на слагаемые изменяет интерпретацию задачи. При делении каждого из слагаемых на первый делитель (фактор) мы имеем целую и дробную части. В этом случае на дискретной сетке необходимо расположить прямоугольники, площадь которых соответствует произведению целых частей и первого фактора, и объединить в один прямоугольники, площадь которых соответствует произведению дробных частей и первого фактора. Пример оптимального решения с разложением на 3 слагаемых представлен в таблице 0.1.

Таблица 0.1 – Оптимальное решение задачи факторизации через слагаемые для дискретной сетки

Выражение	Описание
$S = ab$	S – факторизуемое число; a, b – натуральные факторы
$S = S_1 + S_2 + S_3$	раскладываем S на слагаемые S_1, S_2 и S_3
$\frac{S_1}{a} = n_1 + r_1$	находим целую n_1 и дробную r_1 части от деления S_1 на a
$\frac{S_2}{a} = n_2 + r_2$	находим целую n_2 и дробную r_2 части от деления S_2 на a
$\frac{S_3}{a} = n_3 + r_3$	находим целую n_3 и дробную r_3 части от деления S_3 на a
$P_1 = an_1$	первый дискретный прямоугольник
$P_2 = an_2$	второй дискретный прямоугольник
$P_3 = an_3$	третий дискретный прямоугольник
$P_4 = a(r_1 + r_2 + r_3)$	четвертый составной дискретный прямоугольник

Разложение на слагаемые увеличивает контроль над оптимизируемой функцией дискретного преобразования делителя за счет возможности оценки ошибки на шаге оптимизации для каждого r. В этом случае не детерминированные методы способны быстрее аппроксимировать выигрышные преобразования.

При сборе эмпирических данных использован генетический алгоритм eaSimple [6] для факторизации числа. Использование генетического алгоритма обусловлено его универсальностью и способностью к решению задач на ограниченных данных, в частности, на одном экземпляре.

Недостаток этого подхода – в высокой требовательности к ресурсам при увеличении популяций, в «наивной» реализации он не сопоставим с GNFS по производительности. Для его применения на практике требуется сбор эмпирических данных, аналитические расчеты алгоритмической сложности для разных значений вероятностных и количественных параметров генетического алгоритма, разрядности числа и количества слагаемых, адаптация архитектуры и параметров генетического алгоритма.

Искусственные нейронные сети демонстрируют более высокую производительность обученных моделей для отдельных задач по сравнению с генетическими алгоритмами. Декодерные архитектуры, такие как Generative Pre-trained Transformer (GPT) с авторегрессионным процессом обучения и Generative Adversarial Network (GAN), могут представлять интерес для решения оптимизационной задачи факторизации, они хорошо документированы и соответствуют специфике задачи. Одним из важных аспектов, определяющих успех популярных декодерных моделей, является этап предобучения [5]. Декодерные архитектуры нуждаются в достаточном количестве данных для корректной аппроксимации. Для этапа предобучения нейронной сети, выполняющей целочисленную факторизацию на размеченных и неразмеченных данных, доступны большие массивы производных данных из простых чисел. Генерация больших простых чисел и арифметические операции с ними не являются актуальной проблемой с точки зрения вычислительной сложности.

В работе предложена функция ошибки (1.1) с использованием разложения на слагаемые для нейронных сетей и генетических алгоритмов.

Специфика проблемы больших чисел состоит в колоссальных объемах возможных комбинаций. Для сужения пространства решений проведен анализ зависимости дискретных преобразований делителя с его оптимальностью. Изучение таких связей на известных данных позволяет вводить дополнительные условия и ограничения для процесса оптимизации с использованием нейронных сетей и генетических алгоритмов.

1 Факторизация с использованием генетического алгоритма

При сравнительном тестировании использованы следующие настройки генетического алгоритма: исходная популяция 10000, вероятность образования пары 0.6, вероятность мутации индивида 0.3 и разрядность факторов до 6 знаков. Результаты приведены в таблице 1.1.

Таблица 1.1 – Результаты сравнительного тестирования фитнес функций генетического алгоритма с разложением на множители и без для задачи факторизации

№	S	Шаг успеха оптимизатора без разложения (множитель)	Шаг успеха оптимизатора с разложением (множитель)
1	49118670253	NA	3504 (897817)
2	51933092507	9418 (673747)	NA
3	850266266951	5136 (864757)	446 (864757)
4	261100975771	NA	3980 (755057)
5	375343367551	NA	7459 (593071)
6	72714690589	6164 (138637)	NA
7	13587085747	8612 (61331)	NA
8	828847571509	2279 (230819)	NA
9	828885787931	NA	2146 (841793)
10	69922171051	NA	6959 (83339)
11	96301017707	9058 (869543)	NA
12	320253630097	NA	429 (323077)
13	347866742453	9930 (576881)	5341 (576881)
14	313151820431	6316 (743689)	NA
15	334591178759	NA	5108 (463363)
16	635309551183	5282 (881729)	NA
17	203988146111	137 (721783)	NA
18	608722442407	NA	7067 (719633)
19	238632650083	3168 (429329)	46 (429329)
20	576891026099	NA	1775 (646147)

В общем случае, без разложения, фитнес функция возвращает дробную часть результата деления для минимизации $\frac{S}{a} - \left[\frac{S}{a}\right]$.

В задаче со слагаемыми добавляется штраф, если не происходит уменьшения ошибки по конкретному слагаемому. Для этого вычисляются общая ошибка R как сумма дробных частей r по каждому слагаемому $R = \sum_{r=1}^{i} r$, минимизируемая ошибка дискретизации $D = [R] - R$, и непосредственно штраф:

$$z = pR. \qquad (1.1)$$

Если штраф меньше ранее наложенных на это слагаемое, то его не применяем, если больше – добавляем к D. В зависимости от среды исполнения показатели могут варьироваться, но общая тенденция сохраняется: при увеличении разрядности чисел алгоритм с разложением стабильно демонстрирует более высокую эффективность при сохранении параметров оптимизатора: + 33%. Итерации, в которых оба подхода не возвратили корректный фактор за выделенное количество поколений, в таблице 0.1 не приводятся.

2 Связь дискретных преобразований делителя с его оптимальностью

Полиномиальное представление используется в машинном обучении и многих аналитических методах, в том числе GNFS. В задаче дискретной оптимизации для факторизации в общем случае также подразумевается полиномиальное представление делителя A:

$$A = \sum_{i=0}^{n} a_i \cdot 10^i, \text{ где } a_i \in 0,...,9.$$

Оптимизируемое дискретное преобразование для делителей: $A_i = T(A_{i-1})$.

Для анализа связи между дискретным преобразованием делителя и его оптимальностью был подготовлен источник данных из 2000 произведений 2 простых чисел. Разрядность произведений от 5 до 10 знаков. К источнику данных добавлено разложение на 3 слагаемых S_1, S_2, S_3 по старшим разрядам, где n соответствует порядку самого старшего разряда:

$S_1 = a_n \cdot 10^{n-1}$, $S_2 = a_{n-1} \cdot 10^{n-2}$, $S_3 = S - (S_1 + S_2)$.

Аналогичный подход использовался в примере с генетическим алгоритмом, но с 4 слагаемыми.

К источнику данных добавлены 4 преобразованных делителя через дискретные преобразования T_1, T_2, T_3, T_4 над оптимальным делителем:

T_1 соответствует смещению цифры среднего разряда вверх на единицу: 1 → 2, 2 → 3, …, 9 → 0;

T_2 – смещению цифр среднего и соседних разрядов вверх на единицу, то есть происходит равномерное смещение трёх разрядов в середине;

T_3 – смещению цифр среднего разряда на 2 единицы, а соседних разрядов на единицу вверх, то есть как и в T_2 происходит смещение трёх разрядов, но неравномерное;

T_4 – целой части результата деления на 2:

$$T(a) = \left[\frac{a}{2}\right],$$

где a оптимальный делитель.

Первые три преобразования позиционированы, незначительны, с плавным увеличением отличия от оптимального делителя. Четвертое значительное, с существенным отличием от оптимального делителя.

Для каждого преобразованного делителя рассчитан коэффициент смещения l как отношение суммы поразрядных смещений к общему числу разрядов n:

$$l(T_j) = \frac{1}{n}\sum_{i=0}^{n}\left|b_j - a_j\right|, \qquad (2.1)$$

где T_j – текущее преобразование, a_j знак оптимального делителя в разряде i, b_j – знак преобразованного делителя в разряде i.

Для каждого преобразованного делителя b_t рассчитаны: дробная часть r результата деления общего произведения S на b_t, дробные части r_1, r_2, r_3 результатов деления слагаемых S_1, S_2, S_3 на b_t, сумма дробных частей: $\sum_{i=1}^{3} r_i$.

Анализ зависимости преобразований оптимального делителя через коэффициент смещения (2.1) с остатками от деления выполнен с использованием коэффициента Пирсона [7] и представлен в таблице 2.1:

$$K = P(L, R), \qquad (2.2)$$

где $L = \{l(T_1), l(T_2), l(T_3), l(T_4)\}$, $R = \{r, r_1, r_2, r_3, r_1 + r_2 + r_3\}$, P – функция расчёта коэффициента Пирсона, $K = \{k_1, k_2, k_{03}, k_4\}$ – коэффициенты корреляции.

Таблица 2.1 – Коэффициенты корреляции для смещений делителя (k)

	k_1	k_2	k_3	k_4
r	0,0050973	0,0390797	0,0060046	–0,05226
r_1	0,1121576	–0,2394642	–0,0110255	–0,0404442
r_2	0,0262286	–0,2239337	–0,0251826	–0,0091951
r_3	0,0206891	0,0301176	0,0061416	–0,0238543
$\sum_{i=1}^{3} r_i$	0,0906496	–0,2417458	–0,0179779	–0,0390467

Статистическая интерпретация говорит об отсутствии значимых линейных связей между выбранными показателями, но выделяются общие закономерности:

1) усиление корреляционных связей при добавлении расчетов со слагаемыми, что косвенно объясняет результаты тестирования генетического алгоритма из таблицы 1.1;

2) строго положительные коэффициенты при минимальном преобразовании T_1;

3) строго отрицательные коэффициенты для максимального преобразования T_4;

4) усиление отрицательной корреляции для преобразования T_2 близкое к значимому.

Заключение

В работе задача факторизации натуральных чисел на простые множители рассматривается в контексте дискретной оптимизации и машинного обучения. Для генетических алгоритмов и нейронных сетей предложен подход с разложением на слагаемые, который вместе со связанной функцией для генетических алгоритмов (фитнес функция) и нейронных сетей (функция ошибки) (1.1) на примере с генетическим алгоритмом показал 33% прирост эффективности.

Рассчитан коэффициент Пирсона для изменений функции дискретного преобразования оптимального делителя и минимизируемых дробных частей (2.2) через коэффициент смещения (2.1). Это позволяет аппроксимировать область оптимальных дискретных преобразований через вычисление коэффициента корреляции Пирсона для случайного коэффициента смещения (2.1) и дробных частей от пробного делителя вместо оптимального как в (2.2). В этом случае увеличение количества и уточнение анализируемых показателей, усиление их статистической значимости повышает качество аппроксимации и эффективность выбранных средств.

ЛИТЕРАТУРА

1. *Lenstra*, *A*. The Development of the Number Field Sieve / A. Lenstra and H. Lenstra (eds.). – Lect. Not. in Math. 1554. – Springer – Verlag, Berlin, 1993. – 139 p

2. *Sun*, *D*. Optimization and Performance Analysis of Shor's Algorithm in Qiskit / D. Sun, N. Zhang, F. Franchetti. – Carnegie Mellon University, Pittsburgh, USA, 2023. – Mode of access: https://users.ece.cmu.edu/~franzf/papers/hpec_2023_Quantum_final.pdf.

3. *Zimmermann*, *P*. Cado NFS report Factorization of RSA-250 / P. Zimmermann. – Mode of access: https://gitlab.inria.fr/cado-nfs/cado-nfs.

4. *Pomerance*, *C.A*. Tale of Two Sieves / C.A. Pomerance // Notices of the AMS. – 1996. – Vol. 43, № 12. – P. 1473–1485.

5. *Improving Language Understanding by Generative Pre-Training* / A. Radford, K. Narasimhan, T. Salimans, I. Sutskever // OpenAI, 2018. – 12 p.

6. Документация DEAP library 1.4.1 – Режим доступа: https://deap.readthedocs.io/en/stable/api/algo.html

7. *Sammut*, *C*. Encyclopedia of Machine Learning / C. Sammut, G.I. Webb. – Springer Science + Business Media LLC, 2011. – 831 p.

Поступила в редакцию 07.02.2025.

Информация об авторах

Чагочкин Александр Александрович – магистрант