

Трухин В. О., Анисич А. И., Лобанова Э. А., Нефедев К. В.
V. O. Ttukhin, A. I. Anisich, E. A. Lobanova, K. V. Nefedev

РЕАЛИЗАЦИЯ НЕКЛАССИЧЕСКОГО АЛГОРИТМА ПОЛНОГО ПЕРЕБОРА МОДЕЛИ ИЗИНГА С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ CUDA

REALIZATION OF NON-CLASSICAL ALGORITHM OF FULL BRUTE-FORCE SEARCH OF ISING MODEL USING CUDA TECHNOLOGY

Трухин Вячеслав Олегович – старший преподаватель Дальневосточного федерального университета, младший научный сотрудник Института прикладной математики ДВО РАН (Россия, Владивосток).

Viacheslav O. Trukhin – Senior Lecturer, Far Eastern Federal University, Junior Researcher, Institute of Applied Mathematics, Far Eastern Branch of RAS (Russia, Vladivostok).

Анисич Александр Игоревич – студент Дальневосточного федерального университета (Россия, Владивосток).

Alexandr I. Anisich – Student, Far Eastern Federal University (Russia, Vladivostok).

Лобанова Элиза Александровна – студент Дальневосточного федерального университета, лаборант Института прикладной математики ДВО РАН (Россия, Владивосток).

Eliza A. Lobanova – Student, Far Eastern Federal University, Laboratory Assistant, Institute of Applied Mathematics, Far Eastern branch of RAS (Russia, Vladivostok).

Нефедев Константин Валентинович – доктор физико-математических наук, профессор департамента теоретической физики и интеллектуальных технологий ИНТиПМ Дальневосточного федерального университета, главный научный сотрудник ИПМ ДВО РАН (Россия, Владивосток). E-mail: nefedev.kv@dvfu.ru.

Konstantin V. Nefedev – Holder of an Advanced Doctorate (Doctor of Science) in Physical and Mathematical Sciences, Full Professor, Professor at the Department of Theoretical Physics and Intelligent Technologies, Institute of High Technologies and Advanced Materials, Far Eastern Federal University (Russia, Vladivostok). E-mail: nefedev.kv@dvfu.ru.

Аннотация. В статье представлен алгоритм полного перебора модели Изинга. Основные методы, применяемые при расчётах, это использование технологии CUDA и параллелизация на архитектуре графического процессора. Подробно описываются структура алгоритма и его применение к решению задач в статистической термодинамике. Уделено внимание сравнению скорости работы алгоритма относительно классического переборного алгоритма на Python и C++. Предложенный подход позволяет проводить точные расчёты для систем квадратной решётки спинов модели Изинга со случайным распределением обменных констант.

Summary. The paper presents an algorithm for complete enumeration of the Ising model. The main methods used in the calculations are the use of CUDA technology and parallelization on GPU architecture. The structure of the algorithm and its application to the solution of problems in statistical thermodynamics are described in detail. Attention is paid to the comparison of the speed of the algorithm with respect to the classical brute-force algorithm in Python and C++. The proposed approach allows carrying out accurate calculations for systems of the square lattice of spins of the Ising model with a random distribution of exchange constants.

Ключевые слова: статистическая физика, модель Изинга, точное решение, алгоритмы, параллельные вычисления, графический процессор.

Key words: statistical physics, Ising model, exact solution, algorithms, parallel computing, GPU.

Исследование выполнено за счёт гранта Российского научного фонда № 24-71-10069, <https://rscf.ru/project/24-71-10069/>.

Введение. Поиск точных решений для статистических моделей, с одной стороны, является критически важным для прогнозирования или описания реальных событий и экспериментов, с другой стороны, это вычислительно сложная проблема и может служить мерой эффективности как программного, так и аппаратного обеспечения.

В данной работе показан алгоритм поиска точного решения, которое представлено в виде плотности всех возможных состояний для квадратной решётки спинов модели Изинга (МИ) со случайным распределением обменных констант [1; 2], пример приведён на рис. 1. Эта модель не ограничивается лишь статистической физикой и может быть представлена в виде графа. Таким образом, решения данной модели вызывают интерес в различных областях науки. В настоящее время активно ведутся работы по оптимизации таких решений [3].

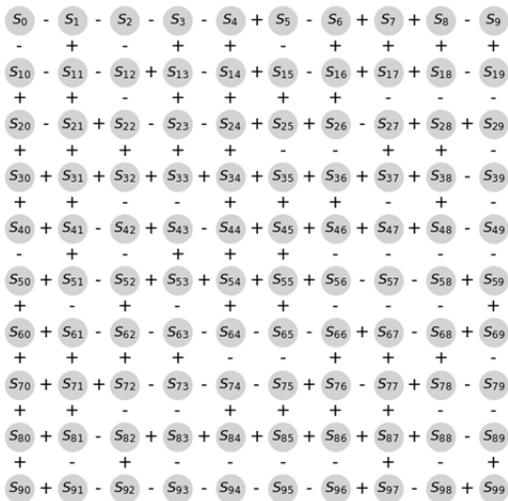


Рис. 1. Пример решётки со случайным распределением обменных констант

Модель Изинга. Простое математическое описание модели Изинга [9] вместе с отсутствием её точного решения делает её эталоном в теории сложности, т. к. любая NP-сложная задача может быть сведена к задаче нахождения вектора состояний для МИ со случайным распределением обменных констант [10; 11; 12].

В работе полная энергия взаимодействия рассчитывается по формуле

$$E = - \sum_{\langle i,j \rangle} J_{ij} S_i S_j,$$

где $S_i \in \{-1; 1\}$, $J_{ij} \in \{-1; 1\}$, $\langle i, j \rangle$ – суммирование по ближайшим соседям.

Алгоритм. На вход алгоритма мы подаём матрицу связей J и линейный размер системы L , в результате работы мы хотим получить данные, которые опишут систему во всех её состояниях, т. е. энергия – спиновый избыток – вырождение.

На первом этапе берутся нулевой и первый столбцы, рассчитывается их плотность состояний как независимых одномерных цепочек и для каждой заполняется 4-мерный массив: первая ось – это удвоенная максимальная энергия +1, вторая ось – удвоенный спиновый избыток +1, третья ось – это конфигурации, её размер 2^L , и четвёртая ось имеет размерность количества простых чисел, остатки от деления на которые записываются по этим координатам, размер этой оси также увеличен на 1 для проверки на совпадение. Избежать добавления этой оси можно только в случае размерности системы, не превышающей 8×8 , в ином случае вырождение может выходить за рам-

Термодинамические системы с фиксированным количеством частиц могут быть охарактеризованы канонической статистической суммой. Значение этой функции заключается в том, что производные от неё позволяют вычислить важные термодинамические параметры системы, такие как свободная энергия и теплоёмкость. Точное решение для статистической суммы есть для двумерной МИ [4], однако это решение не позволяет моделировать поведение системы со случайным распределением обменных констант или с заданным магнитным полем, но является хорошим подспорьем в проверке точности различных алгоритмов. Для решения задачи нахождения минимума в настоящее время существуют различные алгоритмы, такие как алгоритмы машинного обучения [5], квантовый алгоритм адиабатического отжига [6], нейронные сети [7] и множество стохастических алгоритмов [8]. Однако, в отличие от упомянутых методов, стохастические алгоритмы могут тратить слишком много времени на термализацию.

ки целочисленного встроенного типа данных большинства низкоуровневых языков. Устройство функции представлено на рис. 2.

Второй этап заключается в соединении двух массивов состояний, полученных на предыдущем шаге, в один новый, размерность которого будет учитывать новое количество спинов. Устройство функции представлено на рис. 3.

Третий этап закликивает первую и вторую функции до конца решётки, формулы для расчёта размерности на каждом этапе весьма тривиальны. В последнем приращении у финального массива состояний можно не задавать ось конфигураций.

```
-----  
Функция generator EMC(L, EMC, S, E_max, M_max, m)  
  x := blockIdx.x * blockDim.x + threadIdx.x  
  length_dev := 2 ^ L  
  E_add := (E_max - 1) / 2  
  M_add := (M_max - 1) / 2  
  
  Для i от x до length_dev с шагом blockDim.x * gridDim.x  
    E := 0  
    M := 0  
    Для j от 0 до L - 2  
      E := E - J_vertical[j * L + m] * S[i * L + j] * S[i * L + j + 1]  
      M := M + S[i * L + j]  
    КонечДля  
    M := M + S[i * L + L - 1]  
    E := E + E_add  
    M := M + M_add  
    atomicAdd(EMC[i * E_max * M_max * prime_n_dev + E * M_max *  
prime_n_dev + M * prime_n_dev], 1)  
    Для j от 1 до prime_n_dev - 1  
      atomicAdd(EMC[i * E_max * M_max * prime_n_dev + E * M_max *  
prime_n_dev + M * prime_n_dev + j], 1)  
    КонечДля  
  КонечДля  
КонечФункции  
-----
```

Рис. 2. Алгоритм перебора одномерной цепочки и заполнения массива

На последнем этапе мы должны раскодировать массив просуммированных остатков от деления на простые числа в массив вырождений, делаем это, основываясь на китайской теореме об остатках и расширенном алгоритме Эвклида [13].

Ускорение вычислений достигается за счёт расчёта уникального индекса для каждого CUDA-ядра [14], что позволяет каждому потоку обрабатывать свой участок данных. Это достигается путём распараллеливания внешнего цикла, где индексы вычисляются как $x = \text{blockIdx.x} \times \text{blockDim.x} + \text{threadIdx.x}$, и итерации по массиву данных с шагом $\text{blockDim.x} \times \text{gridDim.x}$, что обеспечивает эффективное использование ресурсов GPU и минимизирует время выполнения.

Быстродействие. В рамках анализа быстродействия алгоритма было проведено сравнение времени работы алгоритма с алгоритмами прямого перебора, написанными на Python и C++, с технологиями параллелизации по ядрам процессора (см. табл. 1). Использование метода прямого перебора для сравнения аргументировано тем, что именно его можно взять за эталон сложности для данных методов. Таким образом, мы смогли наблюдать кратное превосходство разработанного метода, что развивает интерес дальнейших исследований в этой области. В строке «CUDA» представлен полученный алгоритм перебора.

```

-----
Функция unifying(L, EMC_l, E_max_l, M_max_l, EMC_r, E_max_r, M_max_r, EMC_created, E_max_created,
M_max_created, S, column_number)
  x := blockIdx.x * blockDim.x + threadIdx.x
  length_dev := 2 ^ L
  E_add_l := (E_max_l - 1) / 2
  M_add_l := (M_max_l - 1) / 2
  E_add_r := (E_max_r - 1) / 2
  M_add_r := (M_max_r - 1) / 2
  E_add_created := (E_max_created - 1) / 2
  M_add_created := (M_max_created - 1) / 2

  Для conf_l от x до length_dev с шагом blockDim.x * gridDim.x
    Для E_l от 0 до E_max_l - 1
      Для M_l от 0 до M_max_l - 1
        Для conf_r от 0 до length_dev - 1
          Для E_r от 0 до E_max_r - 1
            Для M_r от 0 до M_max_r - 1
              Если (EMC_l[conf_l * E_max_l * M_max_l * prime_n_dev + E_l * M_max_l *
                prime_n_dev + M_l * prime_n_dev] > 0
                И EMC_r[conf_r * E_max_r * M_max_r * prime_n_dev + E_r * M_max_r *
                prime_n_dev + M_r * prime_n_dev] > 0) Тогда

                E := 0
                M := 0

                Для j от 0 до L - 1
                  E := E - J_horizontal[j * (L - 1) + column_number] * S[conf_l
                    *
                    L + j] * S[conf_r * L + j]
                КонецДля

                E := E + E_l - E_add_l + E_r - E_add_r + E_add_created
                M := M + M_l - M_add_l + M_r - M_add_r + M_add_created

                atomicAdd(&EMC_created[conf_r * E_max_created * M_max_created *
                  prime_n_dev + E * M_max_created * prime_n_dev + M * prime_n_dev],
                  1)

                Для j от 1 до prime_n_dev - 1
                  atomicAdd(&EMC_created[conf_r * E_max_created * M_max_created *
                    prime_n_dev + E * M_max_created * prime_n_dev + M * prime_n_dev
                    + j], EMC_l[conf_l * E_max_l * M_max_l * prime_n_dev +
                    E_l * M_max_l * prime_n_dev + M_l * prime_n_dev + j])
                КонецДля
              КонецЕсли
            КонецДля
          КонецДля
        КонецДля
      КонецДля
    КонецДля
  КонецФункции
-----

```

Рис. 3. Алгоритм перебора соседних цепочек спинов и заполнение соответствующего массива

Таблица 1

Сравнение времён, затрачиваемых на полный перебор систем спинов Изинга с помощью различных программных средств (значения указаны в секундах)

Технология	Размерность				
	3x3	4x4	5x5	6x6	7x7
Один поток Python	0	4,09	-	-	-
Python + numba	0	0,046	66	-	-
Один поток на C++	0,003	0,201	204,564	-	-
C++ + OpenMP	-	-	23,443	-	-
CUDA	0,305	0,602	2,385	15,153	89,114

Заключение. В результате работы алгоритма мы получаем уникальные данные, характеризующие исследуемую систему. Уникальность заключается в том, что можно определить, глобальный ли минимум был найден с помощью алгоритмов, не являющихся точными, например, Монте-Карло или генетического алгоритма [15]. Также это позволяет определить оптимальное количество шагов для метода Метрополиса, необходимых для нахождения глобального минимума.

Полученные с помощью алгоритма характеристики состояний позволяют рассчитать свойства спиновых систем, которые без полной плотности состояний получить нельзя. В том числе удалось классифицировать решётку Эдвардса – Андерсена на ферромагнитное, антиферромагнитное состояния и состояние спинового стекла, а также провести численный эксперимент по определению фазовых свойств таких решёток в поле.

ЛИТЕРАТУРА

- Roma F and Risau-Gusman, S and Ramirez-Pastor, AJ and Nieto, F and Vogel, EE Physical Review B-Condensed Matter and Materials Physics Ground-state topology of the Edwards-Anderson J spin glass model 82 21 214-401 2010.
- Katzgraber, Helmut G and Lee, Lek Wee, Correlation length of the two dimensional Ising spin glass with bimodal interactions, Physical Revive B – condensed Matter and Materials Physics, 71:13, (2005), 134-404.
- Romero, Joshua and Bisson, Mauro and Fatica, Massimiliano and Bernaschi, Massimo Computer Physics Communicationspaper High performance implementations of the 2D Ising model on GPUs (2020) 256.
- Onsager, Lars. (1944). Crystal statistics. I. A two-dimensional model with an order-disorder transition. Physical Review, 65(3-4), 117. APS.
- Maren, Alianna J. (1991). A logical topology of neural networks. In Proceedings of the Second Workshop on Neural Networks.
- Grant, Erica K, Humble, Travis S. (2020). Adiabatic quantum computing and quantum annealing. In Oxford Research Encyclopedia of Physics.
- Korol, Alyona Olegovna, Captain, Vitaly Yurievich. (2021). Neural network for determining the Curie temperature of the two-dimensional Ising model. Far Eastern Mathematical Journal, 21(1), 51-60. Institute of Applied Mathematics, Far East Branch of RAS.
- Janke, Wolfhard. (2008). Monte Carlo methods in classical statistical physics. In Computational many-particle physics (pp. 79-140). Springer.
- E. Ising, Z. Phys. XXXI (1925).
- Markovich, LA. (2019). Parallel algorithm based on the Ising model for solving combinatorial optimization problems. In Information Technology and Systems 2019 (p. 350-358).
- C. H. Papadimitriou, The Euclidean travelling salesman problem is np-complete, Theoretical computer sciences 4 (3) (1977).
- Karp, Richard M. (2010). Reducibility among combinatorial problems. Springer.
- Katz, Victor J. (2007). The Mathematics of Egypt, Mesopotamia, China, India, and Islam: A Sourcebook. Princeton University Press.
- NVIDIA CUDA, [HTTPS://DEVELOPER.NVIDIA.COM/CUDA-ZONE](https://developer.nvidia.com/cuda-zone).
- Panchenko, Tatyana Vyacheslavovna, Tarasevich, Yuri Yurievich. (2007). Comparative analysis of the efficiency of application of genetic algorithms and Metropolis algorithm in problems of solid state physics. Computational Methods and Programming, 8, 77-87.