Дальневосточный математический журнал. 2025. Т. 25. № 1. С. 102–112

УДК 519.17+УДК 519.11+519.684+519.683
MSC2020 68R10 + 82B30 + 65Y05

© В. О. Трухин[1,2], Э. А. Лобанова[1,2], А. И. Анисич[1], К. В. Макарова[1,2], А. Г. Макаров[1,2], К. В. Нефедев[1,2]

Переборный алгоритм декомпозиции для решения модели Изинга

В работе представлен авторский алгоритм исчерпывающего перечисления конфигураций спинов в модели Изинга на квадратной решетке. Уделено внимание параллельной алгоритмизации вычислений на центральном процессоре (CPU) с использованием OpenMP и на графических процессорах (GPU) с использованием CUDA. Описана структура алгоритма, а также его применение к решению задач статистической термодинамики, в частности, к вычислению плотности состояний. Проведено сравнение производительности предложенного алгоритма с последовательными переборными алгоритмами, реализованными на языках программирования Python и C. Результаты показывают, что предложенная декомпозиция модели существенно ускоряет вычисления и позволяет эффективно анализировать системы квадратной решетки спинов модели Изинга размером до 10×10 узлов (100 спинов) с произвольным распределением обменных констант.

Ключевые слова: *алгоритм декомпозиции, модель Изинга, параллельные вычисления.*

DOI: https://doi.org/10.47910/FEMJ202509

Введение

Полный перебор состояний для моделей (поиск точных решений) в статистической физике является важной задачей прогнозирования или описания реальных событий и экспериментов. Кроме того, это вычислительно сложная проблема. Как утверждает Р. Бэкстер в основополагающей работе «Точно решаемые модели в статистической механике» [1], двумерные «точно решаемые модели» представляют большую ценность для проверки общих теорий и предположений. Общепринято считать, что точное решение может служить эталоном для проверки решений, полученных другими алгоритмами, например, гибридного алгоритма Монте-Карло [2] и квантового отжига [3].

[1] Дальневосточный федеральный университет, 690922, г. Владивосток, о. Русский, п. Аякс, 10.
[2] Институт прикладной математики ДВО РАН, 690041, г. Владивосток, ул. Радио, 7.
Электронная почта: trukhin.vo@dvfu.ru (В. О. Трухин).

Модель Изинга — это модель взаимодействующих спинов (магнитных моментов), которые имеют только две возможные ориентации, условно «вверх» и «вниз». Взаимное расположение магнитных моментов на кристаллической или квазикристаллической решетке [4], а также возможность выбора закона взаимодействия значительно расширяют количество различных систем многих взаимодействующих частиц в рамках модели Изинга. Существует чрезвычайно малое число точных решений модели Изинга на решетках, а для спинового стекла поиск точного решения модели Изинга до сих пор является актуальным [5]. Существующими приближенными методами Монте-Карло, нейронными сетями и другими статистическими методами можно посчитать лишь небольшую часть из общего числа 2^N. Под точным решением модели Изинга понимается вычисление энергии E, спинового избытка M, а также вырождения $G(E, M)$ для всех 2^N возможных состояний N спинов для систем конечного числа частиц. Если хотя бы одно из 2^N-состояний не будет вычислено, то это уже будет не точное решение, а приближенное. Задача о поиске основного состояния резко сокращает возможность применения приближенных методов. Решение Онсагера для 2D-модели Изинга на простой квадратной решетке можно считать точным только условно, потому что он использовал предельные переходы и получил решение без внешнего магнитного поля, т.е. без спинового избытка [1,6]. В случае использования приближенных методов неизбежно возникает необходимость проверки получаемых результатов на достоверность. Проверку достоверности можно выполнить в том числе сравнением с точным решением.

Приближенными вычислительными методами, а также «точными» аналитическими методами, использующими предельные переходы, нельзя решить задачу о нахождении всех 2^N-значений E и M. На сегодняшний день отсутствуют аналитические решения для 2D- и 3D-моделей Изинга даже в самом простом ферромагнитном случае в поле. Задача о 3D является не только NP-сложной, но также и экспоненциально ресурсоемкой. Экспоненциально возрастают требования не только к производительности алгоритмов и аппаратной части, но и к хранению данных.

Модель Изинга не ограничивается лишь статистической физикой и вызывает интерес в различных областях науки. В настоящее время активно ведутся работы по оптимизации алгоритмических решений [7]. Свойства термодинамических систем могут быть вычислены, если известна статистическая сумма. Производные от статистической суммы позволяют вычислить важные термодинамические параметры системы, такие как свободная энергия, теплоемкость и магнитная восприимчивость и другие. Аналитическое решение для статистической суммы есть для двумерной модели Изинга [6], но это решение не позволяет вычислять поведение системы с произвольным распределением обменных констант или с заданным магнитным полем, хотя является хорошим подспорьем в проверке точности различных алгоритмов. Для решения задачи нахождения минимума энергии взаимодействия в дискретных моделях в настоящее время существуют различные алгоритмы, такие как алгоритмы машинного обучения [8], квантовый алгоритм адиабатического отжига [9], нейронные сети [10] и множество стохастических алгоритмов [11]. Однако, в отличие от упомянутых методов, стохастические алгоритмы могут затрачивать слишком много времени на достижение термодинамического равновесия (релаксацию). Общий

принцип расчета статистической суммы для конечного числа спинов Изинга неизвестен. Алгоритм перебора может позволить для небольших N расчитать E и M, чтобы построить статистическую сумму.

В данной работе предлагается переборный алгоритм декомпозиции задачи на подзадачи, который позволяет не только увеличить расчетное число спинов, но и получить точное решение для любого распределения обменных констант.

1. Задача о модели спинового стекла Изинга

Задача заключается в ускорении полного перебора всех конфигураций для модели Изинга.

Простое математическое описание модели Изинга вместе с отсутствием ее точного аналитического решения в общем виде делает ее эталоном в теории сложности, так как любая NP-сложная задача может быть сведена к задаче нахождения вектора состояний для модели Изинга с произвольным распределением обменных констант [12–14].

В работе энергия взаимодействия спинов Изинга рассчитывается по формуле

$$E = -\sum_{<i,j>} J_{ij} S_i S_j,$$

где S_i, S_j — спиновые переменные, которые могут принимать значения $+1$ (вверх) и -1 (вниз), J_{ij} — константа обмена между взаимодействующими спинами в модели спинового стекла Эдвардса – Андерсона, $\langle i,j \rangle$ — суммирование по ближайшим соседям. Это модель Изинга с произвольными константами обмена между парами спинов, т.е. $J_{ij}{=}1$ — ферромагнитное упорядочение, $J_{ij}{=}{-}1$ — антиферромагнитное.

Спиновый избыток конфигурации системы определяется как разность между количеством спинов, направленных вверх, и количеством спинов, направленных вниз

$$M = N_+ - N_- = \sum_{i=1}^{N} S_i.$$

Численный расчет всех 2^N значений энергии и спинового избытка для каждого состояния можно оптимизировать, если выполнять по отдельности вычисление энергии взаимодействия цепочек спинов, см. рис. 1, а затем отдельно производить вычисление энергии взаимодействия между спинами отдельных цепочек.

В данной работе представлен алгоритм поиска точного решения, т.е. плотности состояний $G(E,M)/2^N$ для квадратной решетки спинов модели Изинга с произвольным распределением обменных констант [15, 16] со свободными (открытыми) граничными условиями (пример решетки представлен на рис. 2). Под свободными граничными условиями понимается, что количество соседей у спинов будет зависеть от расположения на решетке, т.е. внутри решетки будет по четыре соседа у каждого, по углам — два, а гранях — по три. Например, спин S_0 будет иметь соседей S_1 и S_{10}, спин S_1 имеет соседей S_0, S_2, и S_{11}, а спин S_{11} имеет соседей S_1, S_{10}, S_{12} и S_{21} (рис. 2). Алгоритм использует исчерпывающее перечисление всех 2^N конфигураций спинов, каждая из которых характеризуется своими значениями E и M. При

этом за счет декомпозиции двумерной модели на одномерные цепочки существенно сокращено количество вычислений для присоединяемых элементов.

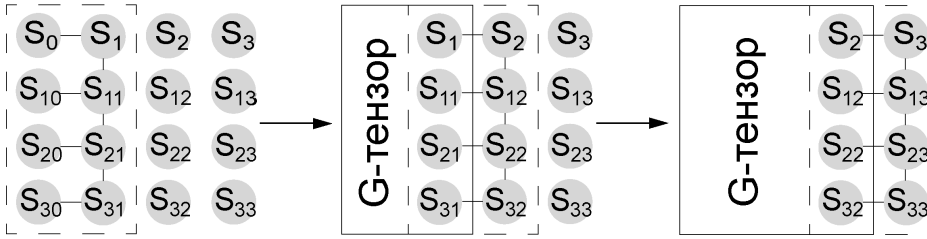

Рис. 1. Принцип объединения цепочек спинов в работе алгоритма декомпозиции. Пунктиром обозначены связи, которые учитываются на каждом шаге.

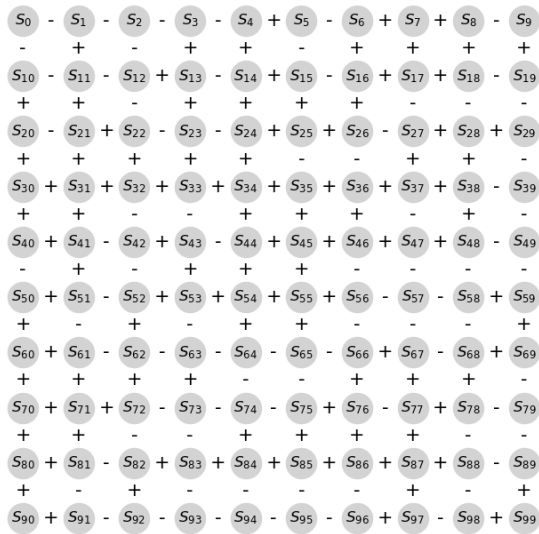

Рис. 2. Пример решетки спинов Изинга с открытыми граничными условиями размером 10×10 узлов (100 спинов) с произвольным распределением обменных констант. Знаки «+»и «−»представляют собой значения обменных констант $J = +1$ и $J = -1$ соответственно. Под спиновой цепочкой понимается строка (например, $\{S_0, \ldots, S_9\}$) или столбец (например, $\{S_0, \ldots, S_{90}\}$) спиновых переменных.

2. Алгоритм декомпозиции

Идея алгоритма заключается в декомпозиции расчёта конфигураций системы спинов (кратность вырождения конфигураций с одинаковой энергией и спиновым избытком). Алгоритм декомпозиции состоит в разбиении большой решетки на части (цепочки спинов).

2.1. Программная реализация алгоритма

Общий принцип работы алгоритма представлен на рис. 1 и в виде псевдокода.

Algorithm 1 Переборный алгоритм расчета плотности состояний методом декомпозиции.

ВВОД: Линейный размер решетки и распределение обменных интегралов.
ВЫВОД: Полная плотность состояний.
Создание массива конфигураций одномерной цепочки методом битового сдвига.
Создание G-тензоров для основной решетки, для добавляемой цепочки и для записи результата суммирования: G[конфигурация цепочки][энергия системы][спиновый избыток][набор остатков от деления на простые числа].
Параллельно для каждой концфигурации крайней цепочки решётки:
for Для всех энергий крайней цепочки **do**
  for Для всех спиновых избытков крайней цепочки **do**
    for Для каждой конфигурации добавляемой цепочки спинов **do**
      for Для всех энергий добавляемой цепочки спинов **do**
        for Для всех спиновых избытков добавляемой цепочки спинов **do**
          Расчет энергии и спинового избытка. Атомарное добавление полученного значения вырождения состояния базовой решетки в G-тензор по индексам кофигурации, энергий и спинового избытка добавляемой цепочки.
        end for
      end for
    end for
  end for
end for
Расшифровка вырождений из набора остатков от деления на набор простых простых чисел с помощью китайской теоремы об остатках и расширенного алгоритма Евклида.
Переформирование данных из полученного G-тензора для всей решетки в плотность состояний: вырождение, энергия, спиновый избыток.

На вход алгоритма подается линейный размер системы L и матрица связей J. В результате работы алгоритма получаем данные, которые описывают систему во всех ее состояниях, т.е. энергию E, спиновый избыток M и вырождение состояний G.

На первом этапе для первой цепочки рассчитывается плотность состояний. Заполняется четырехмерный массив: первая ось — это удвоенная максимальная энергия + 1; вторая ось — удвоенный спиновый избыток + 1; третья ось — это конфигурации, ее размер 2^L; четвертая ось имеет размерность количества простых чисел, остатки от деления на которые записываются по этим координатам. Поскольку при разбиении на остатки от деления на простые числа в любом остатке может оказаться ноль, размер этой оси также увеличен на 1 для заполнения единицей, если значение вырождения ненулевое. Избежать добавления этой оси можно только в случае размерности системы, не превышающей 8×8, в ином случае вырождение

может выходить за рамки целочисленного встроенного типа данных большинства низкоуровневых языков. Такое хранение информации избавляет от необходимости перевыделения памяти и сортировки.

Второй этап подразумевает собой то же самое для следующей цепочки спинов.

Третий этап заключается в соединении двух массивов состояний, полученных на предыдущих шагах, в двумерную решетку: перебираются все состояния левой и правой цепочки и рассчитывается энергия между ними (рис. слева).

Четвертый этап — расчет присоединения еще одной одномерной цепочки.

Пятый этап повторяет третий, но перебирается теперь только крайний слой двумерной цепочки центральный случай).

Далее повторяются действия четвертого и пятого этапов, пока не пройдем по всем столбцам. В последнем приращении у финального массива состояний можно не задавать ось конфигураций.

На последнем этапе мы раскодируем результат из набора остатков от деления на простые числа вырождения, используя китайскую теорему об остатках и расширенный алгоритм Эвклида [17].

В результате для квадратной решетки размером $L \times L = N$ вычислительная сложность исчерпывающего перечисления при использовании декомпозиции снижается с 2^N до $L \times 2^L + (L-1) \times 2^L$. Таким образом, теоретический прирост производительности составляет приблизительно 92% для решетки из 9 спинов и достигает 27 порядков величины для 100 спинов. Однако использование сильно разряженных матриц приводит к большой задержке работы с памятью.

Ускорение обработки достигается за счет исключения операции хранения информации об уникальных состояниях уже обработанной части решетки. Это приводит к невозможности обработки решетки с периодическими граничными условиями, поскольку для расчёта взаимодействия последней цепочки с первой необходимо знать все состояния первой. В вычислительных экспериментах, проведенных с представленным алгоритмом, анализ размерных эффектов при масштабировании, проведенный в работе [5], показал, что влияние поверхностных (граничных) спинов на полученные результаты незначительно. Таким образом, полученные результаты применимы для регулярной решетки.

2.2. Реализация алгоритма на OpenMP и CUDA

Параллельная алгоритмизация вычислений на центральном процессоре осуществлялась с использованием OpenMP. Для этого директива `#pragma omp for` была размещена над внешним циклом функции присоединения, что позволило каждому потоку независимо перебирать собственную конфигурацию в отношении присоединяемых. Чтобы корректно учитывать приращение вырождений, на каждом шаге необходимо применять атомарные операции при обновлении массива плотности состояний. Это связано с тем, что, начиная с присоединения третьей цепочки спинов, может возникать конкуренция потоков за доступ к одним и тем же ячейкам памяти. Для предотвращения состояния гонки (race condition) использовалась директива `#pragma omp atomic`, что гарантировало корректное обновление массива плотности состояний при параллельном выполнении.

При реализации алгоритма с помощью технологии параллельного программирования CUDA (рис. 3) ускорение вычислений достигается за счет расчета уникального индекса для каждого CUDA-ядра, что позволяет каждому потоку обрабатывать свой участок данных. Это обеспечивается путем распараллеливания внешнего цикла, где индексы вычисляются как `x = blockIdx.x × blockDim.x + threadIdx.x`, и итерации по массиву данных с шагом `blockDim.x × gridDim.x`, что обеспечивает эффективное использование ресурсов GPU и минимизирует время выполнения. Для предотвращения конкуренции потоков использовалась атомарная операция *atomicAdd*(). Дополнительное ускорение работы достигалось за счет перемещения небольшой по размерам, но часто используемой матрицы обменных интегралов в константную память.

3. Анализ быстродействия алгоритмов перебора

В рамках анализа быстродействия алгоритма был проведен сравнительный эксперимент: время работы алгоритма сопоставили с реализациями метода прямого перебора на Python и C, использующими параллелизацию по ядрам процессора (таблица 1). Метод прямого перебора выбран для сравнения, так как его можно счи-

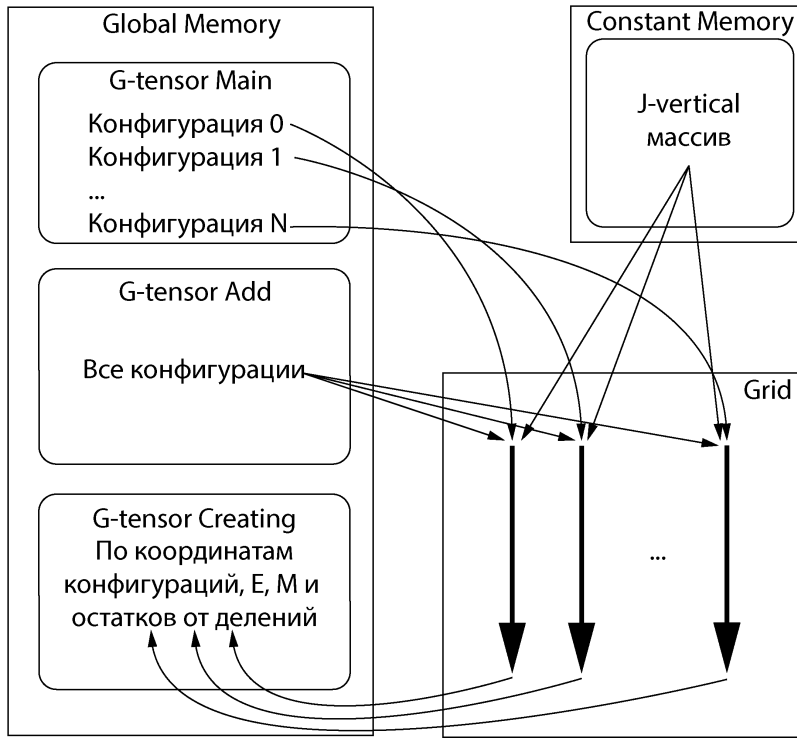

Рис. 3. Параллелизация алгоритма декомпозиции на GPU.

тать эталоном сложности для рассматриваемых методов. В результате наблюдается значительное превосходство предложенного алгоритма, что делает перспективным дальнейшее исследование в этой области.

Вычисления CUDA проводились на графическом процессоре Nvidia Tesla A100 в соотношении одна решетка — одно GPU. Всего было задействовано 6912 CUDA-ядер. OpenMP, Numba, а также все однопоточные вычисления выполнялись на 26-ядерных процессорах Intel Xeon Gold 6230R (2.1 ГГц). Многопоточные приложения использовали 104 потока.

В таблице 1 представлено сравнение времени выполнения предложенного алгоритма перебора с классическими алгоритмами перебора в различных реализациях. Под классическим алгоритмом полного перебора (brute force, исчерпывающего перечисления) понимается алгоритм, в котором каждая новая конфигурация, для которой считается E и M, отличается от предыдущей только одним перевернутым спином, т.е. классический алгоритм является «односпиновым». Таким образом, используемый нами переборный алгоритм декомпозиции не является классическим в том смысле, что для расчета используются данные от предыдущих вычислений частей системы.

Таблица 1. Сравнение времени, затрачиваемого на полный перебор систем спинов Изинга в зависимости от реализации алгоритма и размера системы (значения указаны в секундах).

	4×4	5×5	6×6	7×7	8×8	9×9	10×10
Классический перебор							
Python	7.91	5610	—	—	—	—	—
Python (+Numba)	0.88	121	—	—	—	—	—
C (один поток)	0.006	3.9	—	—	—	—	—
C (+ OpenMP)	0.008	0.28	499	—	—	—	—
Алгоритм декомпозиции							
Python	2	41.47	719	9183	—	—	—
Python (+Numba)	3.24	3.97	6.31	28	248	2379	21753
C (один поток)	0.06	0.21	1.08	20.62	234	2379	34366
C (+ OpenMP)	0.06	0.18	0.45	1.74	9.58	75	1007
CUDA	0.112	0.5	1.942	10.07	50	252	2086

Результаты экспериментов (таблица 1) показывают, что реализация на Python без оптимизаций оказалась наименее эффективной, поскольку демонстрирует высокие временные затраты даже для малых размерностей. Однако применение оптимизирующего компилятора Numba позволило значительно улучшить производительность, снизив время выполнения на порядок. Тем не менее даже оптимизированная версия алгоритма на Python остается существенно медленнее решений, реализованных на языке C.

Использование многопоточных вычислений на CPU (OpenMP) обеспечивает значительное ускорение по сравнению с однопоточной реализацией. В частности, для

системы размерностью 10×10 (примерно 1.27×10^{30} конфигураций) применение OpenMP сокращает время выполнения с 34366 до 1007 секунд (с 9.5 часов до 16 минут), что соответствует ускорению более чем в 30 раз, однако рост сложности системы (начиная с 6×6) приводит к экспоненциальному увеличению вычислительных затрат из-за коллизий.

Реализация на GPU с использованием CUDA также демонстрирует значительное ускорение, особенно на больших размерностях. Однако для малых систем использование GPU может быть менее эффективно из-за накладных расходов на передачу данных между памятью и вычислительными блоками. Таким образом, использование параллельных вычислений (OpenMP, CUDA) дает значительный выигрыш по времени, особенно для крупных систем.

На рис. 4 представлена зависимость времени выполнения для двух наиболее эффективных и перспективных реализаций. Использование логарифмической шкалы позволяет четко выделить различия между реализациями.

Разработанный алгоритм декомпозиции продемонстрировал высокую эффективность для исчерпывающего перечисления конфигураций спинов в модели Изинга. Использование технологии OpenMP обеспечило оптимальный баланс между вычислительной сложностью и производительностью, позволяя значительно сократить время расчетов. Полученные результаты подтверждают перспективность дальнейшего изучения гибридных подходов, сочетающих многопоточные вычисления и графические ускорители для решения задач комбинаторной сложности.

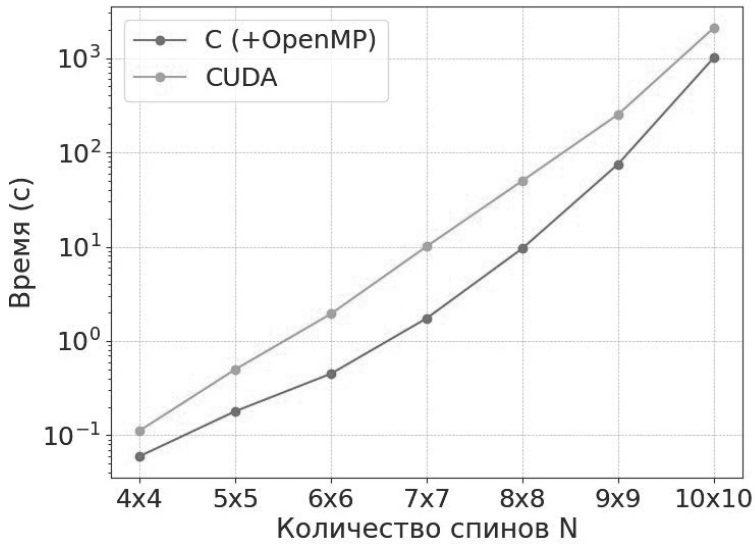

Рис. 4. Сравнение времени перебора систем спинов Изинга алгоритмом декомпозиции для многопоточной реализации (логарифмическая шкала).

Заключение

Предложенный алгоритм декомпозиции показывает лучшие результаты по сравнению с классическим перебором, особенно при использовании многопоточных вычислений и аппаратного ускорения.

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 [18]. Также это позволяет определить оптимальное количество шагов для алгоритма Метрополиса, необходимое для нахождения глобального минимума. Полученные с помощью алгоритма исчерпывающего перечисления характеристики состояний позволяют рассчитать свойства спиновых систем, которые без полной плотности состояний получить нельзя. Это, в частности, необходимо для классификации состояний решётки Эдвардса – Андерсона на ферромагнитное, антиферромагнитное или состояния спинового стекла, а также для проведения численных экспериментов по исследованию фазовых свойств таких решёток во внешнем поле.

Практическое применение показало что использование декомпозиции решетки для вычислений дает значительный выигрыш по времени в сравнении классическим перебором, особенно для крупных систем. Это подтверждает приведенный в параграфе 2.1 теоретический прирост производительности.

Список литературы

[1] Бэкстер Р., *Точно решаемые модели в статистической механике*, Мир, М., 1985.

[2] Makarova K., Makarov A., Strongin V., Titovets Iu., “Canonical Monte Carlo multispin cluster method”, *Journal of Computational and Applied Mathematics*, **427**, (2023), 115153.

[3] Jałowiecki K., Rams M., Gardas B., “Brute-forcing spin-glass problems with CUDA”, *Computer Physics Communications*, **260**, (2021), 107728.

[4] Shevchenko Yu., Makarov A., Andriushchenko P., Nefedev K., “Multicanonical sampling of the space of states of H(2,n)-vector models”, *Journal of Experimental and Theoretical Physics*, **124**, (2017), 982–993.

[5] Trukhin T. V., Strongin V. S., Chesnokov M. A., Makarov A. G., “Thermodynamic equilibrium of ±J Ising model on square lattice”, *Physica A: Statistical Mechanics and its Applications*, **655**, (2024), 130172.

[6] Onsager L., “Crystal statistics. I. A two-dimensional model with an order-disorder transition”, *Physical Review*, **65**, (1944), 117.

[7] Romero J., Bisson M., Fatica M., Bernaschi M., “High performance implementations of the 2D Ising model on GPUs”, *Computer Physics Communications*, **256**, (2020).

[8] Maren A. J., “A logical topology of neural networks”, *Proceedings of the Second Workshop on Neural Networks*, 1991.

[9] Grant E. K., Humble T. S., “Adiabatic quantum computing and quantum annealing”, *Oxford Research Encyclopedia of Physics*, 2020.

[10] Король А. О., Капитан В. Ю., “Нейронная сеть для определения температуры Кюри двумерной модели Изинга”, *Дальневост. матем. журн.*, **21**:1, (2021), 51–60.

[11] Janke W., "Monte Carlo methods in classical statistical physics", *Computational many-particle physics*, 2008, 79–140.

[12] Markovich L. A., "Parallel algorithm based on the Ising model for solving combinatorial optimization problems", *Information Technology and Systems*, 2019, 350–358.

[13] Papadimitriou C. H., "The Euclidean travelling salesman problem is NP-complete", *Theoretical computer science*, **4**:3, (1977), 237–244.

[14] Karp R. M., "Reducibility among combinatorial problems", *Springer*, 2010.

[15] Romá F., Risau-Gusman S., Ramirez-Pastor A. J., Nieto F., "Ground-state topology of the Edwards-Anderson±J spin glass model", *Physical Review B—Condensed Matter and Materials Physics*, **82**:21, (2010), 214–401.

[16] Katzgraber H. G., Lee L. W., "Correlation length of the two-dimensional Ising spin glass with bimodal interactions", *Physical Review B—Condensed Matter and Materials Physics*, **71**:13, (2005), 134–404.

[17] Katz V. J., "The Mathematics of Egypt, Mesopotamia, China, India, and Islam: A Sourcebook", *Physics Letters A*, 2007.

[18] Panchenko T. V., Tarasevich Y. Y., "Comparative analysis of the efficiency of application of genetic algorithms and Metropolis algorithm in problems of solid state physics", *Computational methods and programming*, **8**, (2007), 77–87.

Поступила в редакцию
26 февраля 2025 г.

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

Trukhin V. O.[1,2]*; Lobanova E. A.*[1,2]*; Anisich A. I.*[1]*; Makarova K. V.*[1,2]*; Makarov A. G.*[1,2]*; Nefedev K. V.*[1,2] An enumeration decomposition algorithm for solving the Ising model. *Far Eastern Mathematical Journal.* 2025. V. 25. No 1. P. 102–112.

[1] Far Eastern Federal University
[2] Institute for Applied Mathematics, Far Eastern Branch, Russian Academy of Sciences

ABSTRACT

The article presents a novel algorithm for the complete enumeration of spin configurations in the Ising model on a square lattice. Particular attention is given to the parallel algorithmization of computations on central processing units (CPUs) using OpenMP and on graphics processing units (GPUs) using CUDA. The structure of the algorithm is described, including its main implementation steps, as well as its application to solving problems in statistical thermodynamics, specifically the calculation of the density of states. A performance comparison is conducted between the proposed algorithm and sequential enumeration algorithms implemented in Python and C. The results demonstrate that the proposed approach significantly accelerates computations and enables efficient analysis of square-lattice spin systems in the Ising model with sizes up to 10×10 nodes (100 spins) and with an arbitrary distributions of exchange constants.

Key words: *decomposition algorithm, Ising model, parallel computing.*