

УДК 511.21+517.965+517.547.582

MSC2020 11B37 + 33E05

© М. А. Падалко^{1,2}; Ю. А. Шевченко^{1,2}

Параллельные вычисления модели Эдвардса – Андерсона

Приводится алгоритм параллельного точного вычисления основного состояния двумерной модели Эдвардса – Андерсона со свободными граничными условиями. Время работы алгоритма экспоненциально растет с увеличением стороны квадрата решетки. При фиксации одной из сторон решетки время работы растет полиномиально с увеличением размера другой стороны. Метод может найти применение в теории спиновых стекол, в области вычислений на квантовых компьютерах. Приводятся данные производительности для бимодального распределения. Распределение связей спинов может быть как бимодальным, так и гауссовым. Метод дает возможность осуществлять расчет систем вплоть до размеров 40x40.

Ключевые слова: модель Эдвардса – Андерсона, спиновые стекла, основное состояние, высокопроизводительные вычисления, квантовые вычисления.

DOI: <https://doi.org/10.47910/FEMJ202120>

1. Введение

Модель Эдвардса – Андерсона была предложена Эдвардсом и Андерсоном в 1975 году для описания физики спиновых стекол [1, 2], она представляет собой набор спинов, расположенных в узлах решетки. Каждый спин может принимать значение +1 или -1 (или направление вверх или вниз в некоем одномерном изотопическом пространстве). Все соседние спины взаимодействуют друг с другом. Энергия взаимодействия спинов решетки рассчитывается по формуле

$$E = \sum_{\langle i,j \rangle} J_{ij} s_i s_j, \quad (1)$$

¹ Институт прикладной математики ДВО РАН, 690000, г. Владивосток, ул. Радио, 7.

² Дальневосточный федеральный университет, 690922, г. Владивосток, остров Русский, п. Аякс, д. 10.

Электронная почта: padalkoma1992ofacc@gmail.com (М. А. Падалко), shevchenko.ya@dvfu.ru (Ю. А. Шевченко).

где символ суммы с индексом $\langle i, j \rangle$ обозначает суммирование по всем соседним узлам решетки, J_{ij} – переменные константы связи между спинами, принимают значения $+1$ или -1 с равной вероятностью для бимодального распределения и любые значения, имеющие вероятность

$$P(J_{ij}) = \frac{1}{\sqrt{2\pi J_0}} e^{-\frac{J_{ij}}{2J_0}} \quad (2)$$

в гауссовом распределении.

Спиновые стекла имеют сложный энергетический ландшафт и обладают свойством ультраметричности [3]. Модель спинового стекла используется в таксономии, задачах классификаций, теории информации, биологии, биоинформатики, свертывании и замораживании белков [3], в нейронных сетях, в оценках эффективности работы квантовых компьютеров, в задачах оптимизации. В процессе развития теории нейронных сетей, в частности, после создания сети Хопфилда [4–6], ограниченной машины Больцмана [7], глубокой машины Больцмана [8] интерес к спиновым стеклам значительно усилился. В сети Хопфилда поиск образа по искаженному входному сводится к задаче поиска метастабильных состояний с максимальной корреляцией входного образа с обучаемым образом. Множество локальных минимумов состояний соответствует множеству запоминаемых образов. Задавая должным образом матрицы связей между спинами, можно контролировать величину барьеров между состояниями, количество локальных минимумов, ширину долин и т.д. [9] Этим объясняется множество публикаций, посвященных анализу и расчету количества метастабильных состояний [10–12]. Знание физики спинового стекла позволяет контролировать производительность, количество ошибок и объем обрабатываемых данных нейронной сети, емкость памяти. Развитием сети Хопфилда является машина Больцмана, работа которой имеет вероятностный характер. Машина Больцмана позволяет преодолевать относительно небольшие локальные минимумы и оставаться в более глубоких. Это делает ее работу более устойчивой и эффективной в практических приложениях.

Одно из важных практических применений методов поиска основного состояния – тестирование квантовых компьютеров. Алгоритмы поиска основного состояния могут служить бенчмарком для квантовых компьютеров. Например, компьютеры DWave приспособлены для решения задачи поиска основного состояния спинового стекла [13, 14]. Из-за несовершенства технологий в работе квантовых компьютеров происходят ошибки, приводящие к неверным результатам. Отсюда возникает задача построения априори известного основного состояния для тестирования квантового компьютера [15]. Интересный подход к решению этой проблемы приведен в [16]. В его основе лежит сведение задачи решения системы линейных алгебраических уравнений (с алгебраическими операциями по модулю два) к задаче поиска основного состояния. Стоит отметить, что помимо квантовых компьютеров разрабатываются и другие аналоговые устройства для решения задачи оптимизации модели Эдвардса – Андерсона [15–17].

Теория спиновых стекол тесно связана с задачами оптимизации, в частности, с проблемой равенства P и NP классов. Точный полиномиальный по времени метод

поиска основного состояния не найден до сих пор. Но существует множество вероятностных подходов посвященных этой проблеме. Широко применяются Монте-Карло методы [18, 19], алгоритмы имитации отжига [21, 22], параллельного темперинга, мультисканонического Монте-Карло сэмплирования [24].

Особо стоит отметить точный метод, базирующийся на вычислении Пфаффиана и методах модулярной арифметики [25]. Он способен работать в том числе с решетками вплоть до размера 100×100 .

Хартманом и другими авторами было разработано множество подходов для вычисления основного состояния, базирующихся на генетических алгоритмах [26], кластерных алгоритмах [27], алгоритмах поиска кратчайших путей, связывающих фрустрированные плакетки [28]. Последний осуществляет расчет за полиномиальное время и способен рассчитывать решетки размерами до 1448×1448 .

В [29] представлен алгоритм на основе матриц Пфаффиана, способный рассчитывать решетку до 3000×3000 со свободными граничными условиями в одном направлении. Метод использует ФКТ-теорему.

В настоящей работе представлен точный метод получения основного состояния модели Эдвардса – Андерсона с использованием технологий высокопроизводительных параллельных вычислений. Хотя алгоритм экспоненциален по времени, тем не менее он способен выполнять расчеты на вычислительном кластере решетки размером 40×40 за время порядка нескольких суток при распараллеливании по 64 потокам с использованием около 400 Гб оперативной памяти. Предельный размер решетки, доступный алгоритму, меньше, чем в некоторых упомянутых выше работах. Однако в предлагаемом подходе есть ряд достоинств. Во-первых, полиномиальный рост времени расчета (без роста объема оперативной памяти) при фиксации размера одной стороны и увеличении размера другой, что дает возможность считать решетки 40×100 , 40×1000 , 40×10000 и т.д. Данная возможность недоступна методу, представленному в [25]. Во-вторых, метод — точный. В-третьих, алгоритм может работать с различными распределениями (не только с бимодальным). Есть возможность учитывать примеси и неоднородности. И, наконец, еще одним достоинством является простота реализации.

Важной проблемой компьютерных вычислений и математического моделирования является распараллеливание задачи, т.к. это дает возможность использовать мощности современных высокопроизводительных систем и, соответственно, рассчитывать более сложные и большие решетки спинов. Основной материал статьи будет посвящен именно проблеме распараллеливания.

Метод позволяет находить как основное состояние, так и его энергию.

Статья организована следующим образом: в первой главе приводится описание метода и схема распараллеливания. Во второй главе приводятся данные производительности для бимодального распределения. Далее следует заключение.

2. Схема работы алгоритма

Для поиска минимальной энергии необходимо перебрать все возможные состояния спинов решетки, вычислить их энергию и сравнить между собой. Чтобы каждый

раз ее не вычислять для всех спинов, можно вычислять один раз энергии состояний подсистемы (части исходной системы) с заданными границами, а затем расширять подсистему добавлением спинов к ней. Знание состояния границы дает возможность вычислять изменения энергии основного состояния при добавлении спинов: $E \rightarrow E + \Delta E$. Достраивая подсистему добавлением одного спина на каждом шаге до тех пор, пока подсистема не будет достроена до исходной системы, вычисляем на каждом шаге новый минимум энергии. Вычисленная энергия основного состояния на последнем шаге будет являться конечным ответом. Данный подход довольно распространен и используется, в частности, в методе трансфер-матриц в расчетах модели Изинга [30] и квантовых моделей задач многих тел [31,32]. Далее опишем схему распараллеливания, предварительно введем некоторые обозначения и формализуем задачу.

Пусть дана решетка размером $L_y \times L_x$. Описанная выше схема расчета энергии при заполнении спинов одной строки решетки (будем расширять систему строка за строкой) может быть представлена следующим образом:

$$\begin{pmatrix} E(s_1 = -1, s_2 = -1, \dots, s_{L_x} = -1)_{m-1, L_x} \\ E(s_1 = +1, s_2 = -1, \dots, s_{L_x} = -1)_{m-1, L_x} \\ \dots \\ E(s_1 = +1, s_2 = +1, \dots, s_{L_x} = +1)_{m-1, L_x} \end{pmatrix} \xrightarrow{\text{доб. } s_{m,1}} \begin{pmatrix} E(s_1 = -1, s_2 = -1, \dots, s_{L_x} = -1)_{m,1} \\ E(s_1 = +1, s_2 = -1, \dots, s_{L_x} = -1)_{m,1} \\ \dots \\ E(s_1 = +1, s_2 = +1, \dots, s_{L_x} = +1)_{m,1} \end{pmatrix}$$

$$\xrightarrow{\text{доб. } s_{m,2}} \dots \xrightarrow{\text{доб. } s_{m,L_x}} \begin{pmatrix} E(s_1 = -1, s_2 = -1, \dots, s_{L_x} = -1)_{m, L_x} \\ E(s_1 = +1, s_2 = -1, \dots, s_{L_x} = -1)_{m, L_x} \\ \dots \\ E(s_1 = +1, s_2 = +1, \dots, s_{L_x} = +1)_{m, L_x} \end{pmatrix}.$$

В схеме значения минимальной энергии подсистемы с заданной границей из L_x спинов записываются в векторы-столбцы. Каждой компоненте столбца ставится в соответствие заданная граница подсистемы. Состояние границы задается набором $(s_1, s_2, \dots, s_{L_x})$, в столбце имеется 2^{L_x} компонент. Два нижних индекса y, x справа после скобки $E(\dots)_{y,x}$ обозначают подсистему, состоящую из y строк с x спинами в верхней строке y . Мы ее расширяем на каждом шаге, добавляя по одному спину слева направо, поэтому данные индексы позволяют понять, с какой подсистемой мы имеем дело на текущем шаге. Систему обозначений можно понять на примере решетки с $L_x = 4$ на рис. 1. После заполнения одной строки повторяем процесс для следующей, пока не получим исходную систему.

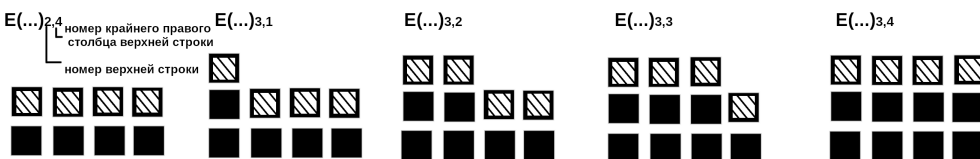


Рис. 1. Пример обозначения энергий подсистем для $L_x = 4$ для различных длин верхней строки, заполняемой слева направо. (Верхняя граница обозначена заштрихованными квадратами (спинами)).

Рассмотрим детально расчет E . Знание границы и всех возможных состояний добавляемого спина $s_{m,n}$ (индексы m и n обозначают позицию спина по вертикали и горизонтали соответственно) позволяет рассчитать минимальное значение E двух подсистем с границами, отличающимися состоянием только одного спина, по формулам

$$E(s_1, s_2, \dots, s_n=1, \dots, s_{L_x})_{m,n} = \quad (3)$$

$$= \min \left(E(s_1, \dots, s_n = -1, \dots, s_{L_x})_{m,n-1} + \Delta E((s_1, \dots, s_n = -1, \dots, s_{L_x})_{m,n-1}, s_{m,n} = -1), \right.$$

$$E(s_1, \dots, s_n = +1, \dots, s_{L_x})_{m,n-1} + \Delta E((s_1, \dots, s_n = +1, \dots, s_{L_x})_{m,n-1}, s_{m,n} = -1) \left. \right)$$

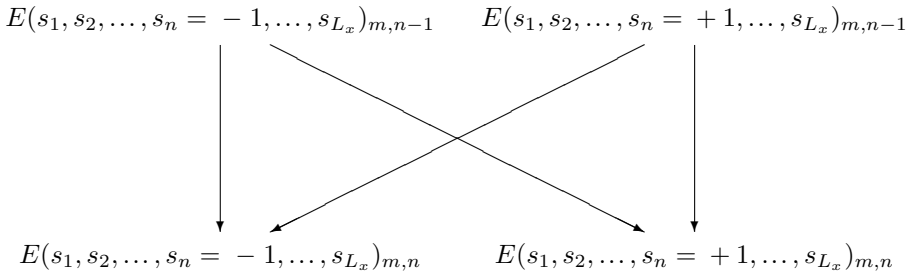
и

$$E(s_1, s_2, \dots, s_n = +1, \dots, s_{L_x})_{m,n} = \quad (4)$$

$$= \min \left(E(s_1, \dots, s_n = -1, \dots, s_{L_x})_{m,n-1} + \Delta E((s_1, \dots, s_n = -1, \dots, s_{L_x})_{m,n-1}, s_{m,n} = +1), \right.$$

$$E(s_1, \dots, s_n = +1, \dots, s_{L_x})_{m,n-1} + \Delta E((s_1, \dots, s_n = +1, \dots, s_{L_x})_{m,n-1}, s_{m,n} = +1) \left. \right).$$

Здесь ΔE — изменение энергии после добавления спина, первый аргумент ΔE в скобках обозначает состояние границы, второй — состояние добавляемого спина. Минимальная энергия подсистемы нового набора для одного состояния границы рассчитывается с использованием только двух значений E предыдущего набора, отличающихся состоянием единственного спина с номером столбца n (к данному столбцу с номером n добавляется спин):



Таким образом, для расчета нового набора минимальных энергий следует сгруппировать все состояния по парам, в каждой из которых состояния граничных спинов на позиции n различаются, на остальных местах — совпадают. После добавления следующего спина уже на позицию $n+1$ столбца конфигурации границ спинов каждой пары будут отличаться только состоянием $n+1$ спина. Пример разбиения на пары набора граничных спинов для решетки с $L_x = 5$ для всех пяти шагов заполнения строки приведен на рис. 2. Энергии пары данного набора E определяются энергиями только одной пары предыдущего набора. Количество состояний граничных спинов равно 2^5 , следовательно, количество пар будет 2^4 . В листинге 1 приведен пример псевдокода разбиения состояний граничных строк на пары.

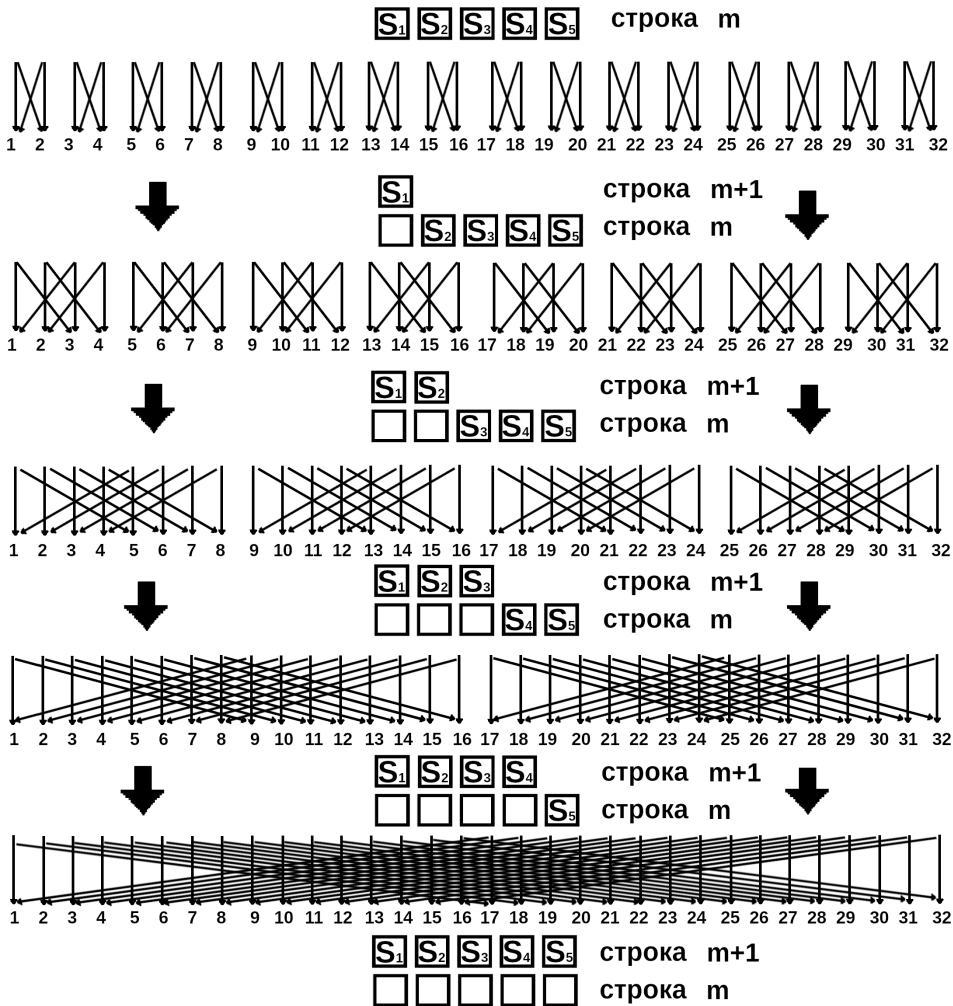


Рис. 2. Разбиение состояний граничных спинов решетки с $L_x = 5$ на пары при достройке строки $m + 1$. (Для достройки строки необходимо добавить 5 спинов.)

В соответствии со схемой на рис. 2 для разбиения на пары элементов необходимы два вложенных друг в друга цикла. В листинге эти циклы осуществляются по переменным *index_out* и *index_in*. В переменные *state1* и *state2* записываются номера состояний границ пары. Энергии состояний хранятся в массиве $E[...]$. Можно заметить, что с увеличением номера столбца расстояние между номерами спинов также растет. Из-за того что при добавлении спина связи меняются, становится понятно, что в схеме распараллеливания придется осуществлять обмен данными между потоками нетривиальным образом. Для того чтобы понять, как это нужно сделать, перепишем схему разбиения на пары в другом виде. Будем записывать номера состояний спинов пары вместе (одно под другим) так, как это сделано на рис. 3.

```

for (int row=0; row < Ly; row++)
{
int max_step_out=(2^Lx)/2, max_step_in=1;
int state1=0, state2=max_step_in;
for (int col=0; col < Lx; col++)
{
for (int index_out=0; index_out < max_step_out; index_out++)
{
for (int index_in=0; index_in < max_step_in; index_in++)
{
state1=index_out*max_step_in*2+index_in;
state2=state1+max_step_in;
.....
calc_E(...)
// (E[state1], E[state2]) => (E[state1], E[state2])
.....
}
}
max_step_out=max_step_out/2; max_step_in=max_step_in*2;
//
}
}

```

Листинг 1. Псевдокод перебора строк и столбцов, разделения состояний на пары, вычисления новых минимумов энергии.

Если бы мы хотели, чтобы переменные с энергиями спаренных состояний находились на соседних участках памяти (т.е. парами), то при каждом изменении подсистемы

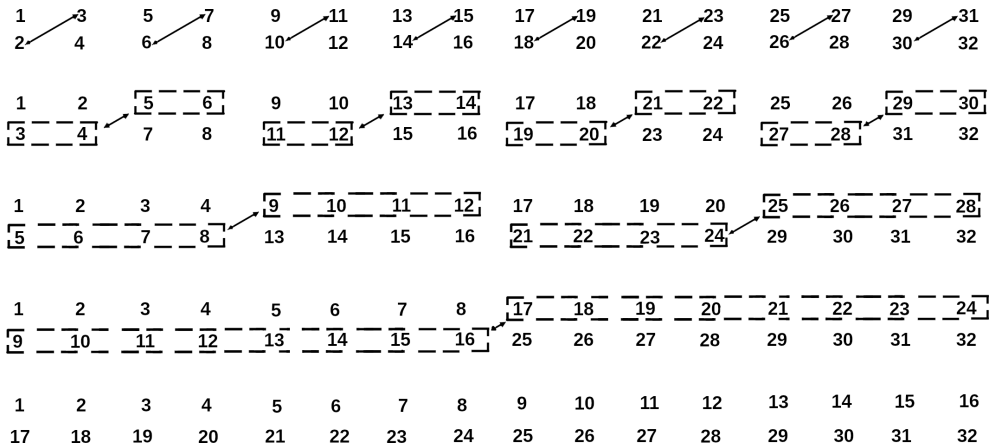


Рис. 3. Разбиение состояний граничных спинов решетки с $L_x = 5$ на пары при достройке строки $m + 1$. В данной схеме спины каждой пары записаны вместе. Стрелки обозначают направления перемещения данных. Каждая пара строк соответствует подсистеме с заданным количеством спинов самой верхней строки (вид всех 5 подсистем приведен на рис. 2). Для $L_x = 5$ имеем 5 возможных подсистем различных форм со спинами, заполняемыми слева направо (с одним верхним спином, двумя и т.д.).

мы бы перемещали данные так, как это изображено на схеме стрелками.

По схеме на рис. 3 можно понять, как перемещать данные между потоками при распараллеливании. С ростом номера столбца разница номеров состояний пары увеличивается, поэтому, начиная с определенного номера заполняемого столбца n_{exch} , необходимо перемещать данные между потоками в соответствии со схемой 3. Можно заметить, что в блоках с данными, обведенными на рисунке пунктирной линией, номера состояний расположены упорядоченно. Это позволяет построить следующую схему обмена данными. При работе с подсистемами с количеством спинов верхней строки $n < n_{exch}$ в пределах потока данные не перемещаются. Разбиение элементов на пары и расчет энергии осуществляется по алгоритму листинга 1. Далее, когда номер заполняемого столбца $n \geq n_{exch}$, осуществляем обмен данными между потоками так, чтобы спаренные элементы располагались так же, как на схеме рис. 3. Исходя из этого, схема обмена данными между потоками будет выглядеть так, как это изображено на рис. 4. Данная схема выполнена для 16 потоков. Схема согласуется со

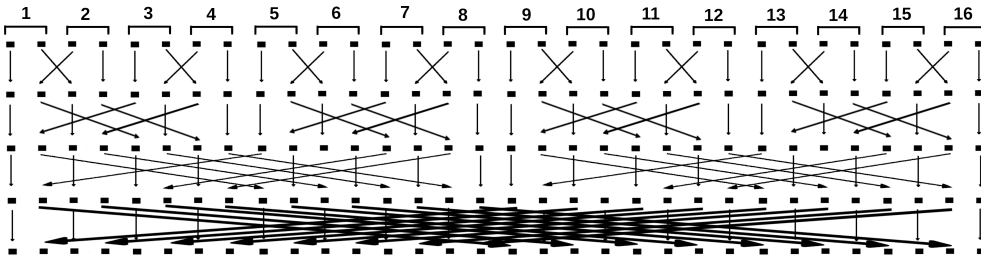


Рис. 4. Схема обмена данными между 16 потоками. В каждом потоке память делится на две части. Одна остается, другая передается другому потоку, а на ее место записывается половина памяти другого потока. Числа обозначают номер потока, закрашенные квадраты — блок памяти.

схемой рис. 3. Для 16 потоков будет осуществляться 4 операции обмена для последовательного образования четырех новых подсистем. Внутри потока после операции обмена энергия будет рассчитываться по схеме рис. 2 с двумя предпоследними строками (каждый элемент верхней строки будет взаимодействовать с нижележащим элементом). Код расчета минимумов энергий внутри потока приведен в листинге 2. В отличие от кода листинга 1, здесь номера элементов пар не меняются при добавлении спина с номером n по горизонтали таким, что $n \geq n_{exch}$.

```

int max_step_in=(2^Lx)/Nthr/2;
int state1=0, state2=max_step_in;
for (int index_in=0; index_in < max_step_in; index_in++)
{
state1=startstate1+max_step_in;
state2=startstate2+max_step_in;
.....
calc_E(...)
// (E[state1], E[state2]) => (E[state1], E[state2])
.....
}
    
```

Листинг 2. Псевдокод разделения состояний на пары, вычисление новых минимумов

энергии для подсистем с номером столбца $n \geq n_{exch}$. $Nthr$ — число потоков. $startstate1$ и $startstate2$ — номера стартовых состояний, хранящихся внутри блоков памяти и передающиеся потоками.

Т.к. согласно схеме рис. 3 данные внутри блоков расположены по порядку, то для облегчения работы с вычислениями номеров состояний можно передавать также и начальные номера состояний в блоках. Перебирая смещения, можно восстановить нужное состояние в соответствии со схемой на рис. 3. В листинге этот процесс продемонстрирован при вычислении переменных $state1$ и $state2$. Можно заметить, что в схеме рис. 3 обмен данными между потоками можно сделать при помощи двух вложенных циклов, как это было сделано в листинге 1. Основной псевдокод, вычисляющий минимумы энергии для различных подсистем, приведен в листинге 3.

```

for (int row=1; row < Ly; row++)
{
  for (int col=0; col < n_exch; col++)
  {
    for (int thrd_num=0; thrd_num < thrd_amount; thrd_num++)
    {calc_E(min_E_array, col, Lx);}
  }
  int interval_amount_of_thrds=2, interval_amount_of_thrds_div_2=1,
  amount_of_bunchs_of_connected_thrds=N_thrd/2;
  for (int col=n_exch; col < Lx; col++)
  {
    // mem swap      ===== v
    for (int thrd_num_out=0; thrd_num_out <
        amount_of_bunchs_of_connected_thrds; thrd_num_out++)
    {
      for (int thrd_num_in=0; thrd_num_in <
          interval_amount_of_thrds_div_2; thrd_num_in++)
      {
        int thrd_num_1=thrd_num_out*interval_amount_of_thrds+thrd_num_in,
            thrd_num_2=thrd_num_1+interval_amount_of_thrds_div_2;
        swap_mem(min_E_array, thrd_num_1, thrd_num_2);      // ! ! !
      }
    }
    // mem swap      ===== ^
    for (int thrd_num=0; thrd_num < N_thrd; thrd_num++)
    {calc_E(min_E_array, col, Lx);}
  }
  ...
  mem_restore(...);
  ...
}

```

Листинг 3. Псевдокод перебора строк, столбцов, обмена памятью между потоками, вычисления энергии для случаев $col < n_{exch}$ и $col \geq n_{exch}$. После перебора столбцов участки памяти перемещаются в обратном направлении. В коде не рассматривается расчет подсистемы состоящей только из самой нижней строки, т.к. в этом случае обмена данными между потоками нет, и ее расчет тривиален. Обмен данными между потоками осуществляется при помощи двух вложенных циклов по переменным $thrd_num_out$ и $thrd_num_in$. При расчете энергии подпрограммой $calc_E(..)$ необходимо передать в качестве аргументов массив энергий, номер заполняемого столбца, длину стороны Lx .

Зная минимальную энергию для данной системы и энергии для каждого состояния граничных спинов Ly -й строки, можно получить конфигурации спинов с минимальной энергией, осуществив $Ly \cdot Lx - 1$ запусков программы, уменьшая количество спинов на единицу каждый раз и задавая верхнюю границу подсистемы, соответствующую минимуму для текущей уменьшенной решетки. Это следует из формул 3 и 4, где $E(s_1, s_2, \dots, s_n = -1, \dots, s_{L_x})_{m,n}$ и $E(s_1, s_2, \dots, s_n = +1, \dots, s_{L_x})_{m,n}$ мы полагаем известными величинами, а $E(s_1, s_2, \dots, s_n = -1, \dots, s_{L_x})_{m,n-1}$ и $E(s_1, s_2, \dots, s_n = +1, \dots, s_{L_x})_{m,n-1}$ — неизвестными. При расчете минимума энергии мы увеличивали подсистему, дстраивая ее каждый раз и начиная с подсистемы, состоящей из самой нижней строки. При расчете же спиновой конфигурации, соответствующей минимуму энергии, мы двигаемся “в обратном направлении”, уменьшая подсистему на каждом шаге и начиная расчет с подсистемы, равной исходной системе. При этом значения минимумов энергии для каждого состояния граничных спинов для подсистем вычисляются на каждом шаге с использованием значений энергий предыдущего шага. Эти значения нужны для отбора конфигураций граничных спинов, соответствующих минимуму энергии.

3. Производительность

В таблице 1 приведены параметры работы параллельного алгоритма поиска основного состояния для решеток 20×20 , 25×25 , 30×30 , 35×35 . Распараллеливание проводилось на 64 потоках (нитях). Отношения времен также приведены в таблице. Можно заметить, что время экспоненциально растет с увеличением линейного размера квадратной решетки. Зная отношения времен для решеток 35×35 и 40×40 , равное 43.07, можно оценить время расчета решетки 40×40 . Оно составит, примерно, 5 ч.

Таблица 1. Производительность параллельного алгоритма расчета fbc решетки спинов для решеток 20×20 , 25×25 , 30×30 , 35×35 на 64 потоках.

номер расчета i	решетка	время расчета t_i	отношение времен t_i к t_{i-1}	размер памяти
1	20×20	81 мс		384 кб
2	25×25	4 с 282 мс	52,86	12 Мб
3	30×30	3 мин 15 с	48,75	384 Мб
4	35×35	2 ч 20 мин	43,07	12 Гб

Далее в таблице 2 приводятся данные производительности алгоритма расчетов решеток с длиной стороны $Lx = 35$ при различных длинах стороны Ly . Распараллеливание осуществлялось по 64 потокам. Из данных таблицы видно, что время растет линейно с увеличением длины одной из сторон решетки.

Таблица 2. Производительность параллельного алгоритма расчета fbc решетки спинов для решеток 35x35, 50x35, 100x35, 250x35, 500x35 на 64 потоках.

номер расчета i	Ly	отношение сторон Ly_i к Ly_{i-1}	время расчета t_i	отношение времен t_i к t_{i-1}
1	35		2,33 ч	
2	50	1,4	3,46 ч	1,49
3	100	2,0	6,89 ч	1,99
4	250	2,5	17,17 ч	2,49
5	500	2,0	35,00 ч	2,03

4. Заключение

В работе представлен алгоритм параллельного расчета двумерной модели спинового стекла со взаимодействием ближайших соседей (Эдвардса – Андерсона). Приведены данные производительности. Алгоритм позволяет осуществлять обмен данными между потоками таким образом, чтобы данные одной нити можно было перенаправить в любую другую нить. Это следует из схемы на рис. 4. Из любой нити, начиная с верхней строки, правильно выбирая нужные стрелки, двигаясь вдоль них, мы обязательно попадем в любую нужную нам нить. Это позволяет использовать алгоритм в создании квантового симулятора. При использовании 64 нитей возможно создание симулятора приблизительно для 40 кубит. Время расчета примерно соответствовало бы данным таблиц 1 и 2, где Lx равнялось бы числу кубитов, а Ly — числу однокубитовых и двухкубитовых операторов. Алгоритм можно использовать в качестве бенчмарка для оценки работы квантовых компьютеров, в нейронных сетях, например, в машине Больцмана. Полиномиальное время роста времени работы при увеличении размера только одной стороны может быть полезным свойством для такой сети. Код программы расчета на языке C++ доступен для скачивания по ссылке ¹. В нем реализован как поиск минимальной энергии, так и поиск конфигураций с минимальной энергией.

Поступила в редакцию
15 октября 2021 г.

Исследование выполнено при финансовой поддержке ГЗ МНВО РФ (проект № 2297-21 а) в рамках программы DAAD "Михаил Ломоносов и Иммануил Кант".

¹https://github.com/kukurbitX29MIH/Edwards_Anderson_parallel_2D

Padalko M. A.^{1,2}, *Shevchenko Yu. A.*^{1,2} Parallel computing of Edwards – Anderson model. *Far Eastern Mathematical Journal*. 2021. V. 21. No 2. P. 234–246.

¹ Institute for Applied Mathematics, Far Eastern Branch, Russian Academy of Sciences

² The Far Eastern Federal University, Russia

ABSTRACT

An algorithm for parallel exact calculation of the ground state of a two-dimensional Edwards – Anderson model with free boundary conditions is given. The running time of the algorithm grows exponentially as the side of the lattice square increases. If one side of the lattice is fixed, the running time grows polynomially with increasing size of the other side. The method may find application in the theory of spin glasses, in the field of quantum computing. Performance data for the bimodal distribution is given. The distribution of spin bonds can be either bimodal or Gaussian. The method makes it possible to compute systems up to a size of 40x40.

Key words: *Edwards – Anderson model, spin glass, ground state, high performance computing, quantum computing.*

References

- [1] J. L. van Hemmen, “Classical Spin-Glass Model”, *Phys. Rev. Lett.*, **49**:6 (1982), 409–412.
- [2] S. F. Edwards, P. W. Anderson, “Theory of spin glasses”, *Phys. F : Metal Phys.*, **5** (1975), 965–74.
- [3] R. Rammal, G. Toulouse, M. A. Virasoro, “Ultrametricity for physicists”, *Rev. Mod. Phys.*, **58** (1986), 765.
- [4] J. J. Hopfield, D. W. Tank, ““Neural” Computation of Decisions in Optimization Problems”, *Biol. Cybern.*, **52** (1985), 141–152.
- [5] R. McEliece, E. Posner, E. Rodemich, S. Venkatesh, “The capacity of the Hopfield associative memory”, *IEEE Transactions on Information Theory*, **33**:4 (1987), 461–482.
- [6] J. L. van Hemmen, “Spin-glass models of a neural network”, *Phys. Rev. A*, **34**:4 (1986), 3435–3445.
- [7] G. S. Hartnett, E. Parker, E. Geist, “Replica symmetry breaking in bipartite spin glasses and neural networks”, *Phys. Rev. E*, **98**:2 (2018), 022116.
- [8] R. Salakhutdinov, G. Hinton, “Deep Boltzmann machines”, *Phys. Rev. E*, **5**:2 (2009), 448–455.
- [9] C. Amoruso, A. K. Hartmann, M. A. Moore, “Determining energy barriers by iterated optimization: The two-dimensional Ising spin glass”, *Phys. Rev. B*, **73**:18 (2006), 184405.
- [10] B. Waclaw, Z. Burda, “Counting metastable states of Ising spin glasses on arbitrary graphs”, *Phys. Rev. E*, **77**:4 (2008), 041114.
- [11] Z. Burda, A. Krzywicki, O. C. Martin, Z. Tabor, “From simple to complex networks: Inherent structures, barriers, and valleys in the context of spin glasses”, *Phys. Rev. E*, **73**:3 (2006), 036110.
- [12] S. Schnabel, W. Janke, “Distribution of metastable states of Ising spin glasses”, *Phys. Rev. E*, **97**:17 (2018), 174204.
- [13] M. W. Johnson , M. H. Amin, S. Gildert , T. Lanting , F. Hamze, N. Dickson, R. Harris,

- A. J. Berkley, J. Johansson, P. Bunyk, E. M. Chapple, C. Enderud, J. P. Hilton, K. Karimi, E. Ladizinsky, N. Ladizinsky, T. Oh, I. Perminov, C. Rich, M. C. Thom, E. Tolkacheva, C. J. Truncik, S. Uchaikin, J. Wang, B. Wilson, G. Rose, “Quantum annealing with manufactured spins”, *Nature*, **473**:7346 (2011), 194–8.
- [14] P. I. Bunyk, E. M. Hoskinson, M. W. Johnson, E. Tolkacheva, F. Altomare, A. J. Berkley, R. Harris, J. P. Hilton, T. Lanting, A. J. Przybysz, J. Whittaker, “Quantum annealing with manufactured spins”, *IEEE Transactions on Applied Superconductivity*, **24**:4 (2014), 1–20.
- [15] D. Perera, F. Hamze, J. Raymond, M. Weigel, H. Katzgraber, “Computational hardness of spin-glass problems with tile-planted solutions”, *Phys. Rev. E*, **101**:2 (2020), 023316.
- [16] I. Hen, “Equation Planting: A Tool for Benchmarking Ising Machines”, *Phys. Rev. Applied*, **12**:1 (2019), 011003.
- [17] D. Pierangeli, M. Rafayelyan, C. Conti, S. Gigan, “Scalable Spin-Glass Optical Simulator”, *Phys. Rev. Applied*, **15**:3 (2019), 034087.
- [18] T. Kadowaki, H. Nishimori, “Quantum annealing in the transverse Ising model”, *Phys. Rev. E*, **58**:5 (2019), 5355–5363.
- [19] G. E. Santoro, R. Martonak, E. Tosatti, R. Car, “Theory of Quantum Annealing of an Ising Spin Glass”, *Phys. Rev. Applied*, **295**:5564 (2002), 2427–2430.
- [20] J. Machta, “Population annealing with weighted averages: A Monte Carlo method for rough free-energy landscapes”, *Phys. Rev. E*, **82**:2 (2010), 026704.
- [21] J. Houdayer, O. C. Martin, “Hierarchical approach for computing spin glass ground states”, *Phys. Rev. E*, **64**:5 (2001), 056704.
- [22] W. Wang, J. Machta, H. G. Katzgraber, “Population annealing: Theory and application in spin glasses”, *Phys. Rev. E*, **92**:6 (2015), 063307.
- [23] D. P. Morelo, A. Ramirez-Pastor, F. Rom, “Ground-state energy and entropy of the two-dimensional Edwards–Anderson”, *Physica A: Statistical Mechanics and its Applications*, **391** (2011).
- [24] N. Hatano, “Evidence for the double degeneracy of the ground state in the three-dimensional $\pm J$ spin glass”, *Phys. Rev. B*, **66** (2002).
- [25] A. Galluccio, “New Algorithm for the Ising Problem: Partition Function for Finite Lattice Graphs”, *Phys. Rev. Lett.*, **84**:26 (2000), 5924–5927.
- [26] A. K. Hartmann, H. Rieger, *New Optimization Algorithms in Physics*, Wiley-VCH, Berlin, 2004.
- [27] A. K. Hartmann, “Cluster-exact approximation of spin glass ground states”, *Physica A*, **224**:480 (1996).
- [28] A. K. Hartmann, “Ground States of Two-Dimensional Ising Spin Glasses: Fast Algorithms, Recent Developments and a Ferromagnet-Spin Glass Mixture”, *J Stat Phys*, **144**:519 (2011).
- [29] G. Pardella, F. Liers, “Exact Ground States of Large Two-Dimensional Planar Ising Spin Glasses”, *Physical Review. E, Statistical, nonlinear, and soft matter physics*, **78** (2011), 056705.
- [30] B. Kaufman, “Crystal statistics. ii. partition function evaluated by spinor analysis”, *Phys. Rev.*, **78** (1949), 1232–1243.
- [31] M. Suzuki, “Transfer-matrix method and Monte Carlo simulation in quantum spin systems”, *Phys. Rev. B.*, **31**:5 (1985), 2957–2965.
- [32] M. Suzuki, “Generalized Trotter’s Formula and Systematic Approximants of Exponential Operators and Inner Derivations with Applications to Many-Body Problems. Commun”, *Math. Phys.*, **51** (1976), 183–190.