

УДК 519.68:[5/6]; 538

## СРАВНИТЕЛЬНЫЙ АНАЛИЗ ЭФФЕКТИВНОСТИ ПРИМЕНЕНИЯ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ И АЛГОРИТМА МЕТРОПОЛИСА В ЗАДАЧАХ ФИЗИКИ ТВЕРДОГО ТЕЛА

Т. В. Панченко<sup>1</sup>, Ю. Ю. Тарасевич<sup>1</sup>

Рассматривается применение генетических алгоритмов к оптимизационным задачам физики твердого тела. Приводится сравнительный анализ эффективности работы генетического алгоритма и алгоритма Метрополиса для модели Изинга с абсолютным значением спина атомов, равным единице. Обсуждаются основные проблемы, возникающие при применении генетических алгоритмов в задачах физики твердого тела.

**Ключевые слова:** генетические алгоритмы, алгоритм Метрополиса, модель Изинга, физика твердого тела, оптимизация.

**1. Введение. Оптимизационные задачи.** Как известно, оптимизационные задачи заключаются в нахождении минимума (максимума) заданной функции. Такую функцию называют функцией цели или целевой функцией. Как правило, функция цели — сложная функция, зависящая от некоторых параметров. В оптимизационной задаче требуется найти значения параметров, при которых функция цели достигает минимального (максимального) значения.

Оптимизационные методы можно разделить на два класса: методы, использующие понятие производной (градиентные методы), и стохастические методы (например, методы группы Монте-Карло). С их помощью можно найти экстремальное значение функции цели, но не всегда можно быть уверенным, что получено значение глобального экстремума. Нахождение локального экстремума вместо глобального называется преждевременной сходимостью. Помимо проблемы преждевременной сходимости существует другая проблема — существенное увеличение времени вычислений при решении сложных задач или использовании более точных оптимизационных методов.

Для устранения указанных проблем проводится поиск новых оптимизационных методов. Предложенные Дж. Холландом генетические алгоритмы [1] основаны на принципах естественного отбора Ч. Дарвина и относятся к классу стохастических методов. Они успешно применяются в различных областях деятельности [2–5], в частности в оптимизационных задачах физики твердого тела [6].

Генетические алгоритмы успешно применяют для решения NP-сложных задач. Для задач класса NP, как известно, нельзя найти полиномиальный алгоритм решения, но можно предложить полиномиальный алгоритм проверки некоторого решения рассматриваемой задачи [7] (полиномиальные алгоритмы — это алгоритмы, временная сложность которых составляет  $O(p)$ , где  $p(n)$  — полином степени  $n$  [7], обычно  $n < 6$ ). Алгоритмы, с помощью которых решаются NP-сложные задачи с использованием генерации случайных решений, называются недетерминированными. На практике часто приходится решать NP-сложные оптимизационные задачи, например, задачи, связанные с перевозками или расписанием, задачи физики твердого тела и т.п. Наибольшее распространение получили недетерминированные алгоритмы группы Монте-Карло и генетические алгоритмы.

**2. Задачи физики твердого тела, приводящие к задачам оптимизации, и методы их решения.**

**2.1. Определение основного состояния вещества.** В теории физики твердого тела одной из оптимизационных задач является определение основного состояния вещества. Основным состоянием называется конфигурация системы, обладающая минимальной энергией [6].

Для описания магнитных свойств кристаллической решетки применяется модель Изинга и модель Гейзенберга. Каждый атом характеризуется проекцией спина. В модели Изинга таких проекций две. Например, если абсолютное значение спина атома равно единице, то проекции принимают значения  $+1$  и  $-1$ . В модели Гейзенберга для атома с абсолютным значением спина, равным единице, возможны следующие проекции спина:  $1, 1/2, 0, -1/2, -1$ .

<sup>1</sup> Астраханский государственный университет, факультет математики и информационных технологий, ул. Татищева, д. 20 а, 414056, г. Астрахань; e-mail: tanuapanchenko@mail.ru, tarasevich@aspu.ru

Для определения полной энергии будем считать, что взаимодействуют только ближайшие соседние атомы. Энергия магнитного взаимодействия двух атомов в модели Изинга вычисляется по формуле  $E_{ij} = -J_{ij}\sigma_i\sigma_j$ , где  $\sigma_i, \sigma_j$  — значения проекций спинов ( $i \neq j$ ), а  $J_{ij}$  — константа обменного взаимодействия. Тогда полная энергия кристаллической решетки в отсутствие магнитного поля вычисляется как сумма энергий по всем парам соседних атомов:

$$E = - \sum_{ij} J_{ij}\sigma_i\sigma_j. \quad (1)$$

В плоской квадратной кристаллической решетке каждый атом взаимодействует с четырьмя соседними, а в кубической решетке количество соседних атомов увеличивается до шести. Даже если для моделирования взять только один моль вещества, то придется рассматривать  $N_A = 6.02 \times 10^{23}$  атомов, поэтому при моделировании рассматривается сильно уменьшенный фрагмент кристаллической решетки, состоящий всего из нескольких сотен или тысяч атомов. Однако даже в этом случае анализ всех возможных спиновых конфигураций модельной системы невозможно провести за разумное время.

Например, для ферромагнетика минимальная энергия достигается в системе атомов с одинаково направленными спинами. Для определения основного состояния необходимо найти минимум энергии системы. В соответствии с определением, приведенным в [8], в качестве внешних параметров в данной задаче полагаются константы и характеристики внешней среды (температура и напряженность магнитного поля), а в качестве внутренних — значения проекций спинов атомов кристаллической решетки. Минимизируемой (целевой) функцией будет полная энергия системы  $E$ . Соответственно, выходными параметрами могут служить такие величины моделируемой кристаллической решетки, как намагниченность, удельная теплоемкость, магнитная восприимчивость и т.п. Целевая функция  $E$  поставленной задачи непосредственно зависит от температуры и спиновой конфигурации атомов. Для некоторых соединений существует несколько основных состояний (такие соединения называются спиновыми стеклами), поэтому задача об определении минимума энергии является многоэкстремальной.

Если система (кристаллическая решетка) состоит из  $N$  атомов, то процесс перебора состояний потребует порядка  $O(N!)$  операций вычисления энергии. При больших  $N$  время, затрачиваемое на решение, становится очень большим, например, для квадратной решетки с линейным размером 16 существует  $(16^2)! \approx 2 \times 10^{13}$  конфигураций атомов! Поэтому для решения рассматриваемой задачи используют недетерминированные алгоритмы.

**2.2. Моделирование фазовых переходов.** Другим примером задачи физики твердого тела, в которой необходимо находить значение минимума энергии, является моделирование магнитных фазовых переходов в кристаллической решетке. При нагревании магнитного образца происходит разрушение магнитного порядка — направление спинов атомов становится неупорядоченным. При достижении критического значения температуры  $T_c$  происходит фазовый переход, т.е. при температурах выше  $T_c$  образец перестает проявлять магнитные свойства. При критическом значении температуры значение удельной теплоемкости максимально. Так как теплоемкость является функцией от энергии, то для каждого значения температуры необходимо определить энергию состояния системы. В результате задача сводится к определению минимума энергии для конкретных значений температуры и, как было показано выше, требует большого количества вычислений. В связи с этим задача о фазовых переходах (как и задача об основных состояниях) относится к классу NP-задач и решается с помощью недетерминированных алгоритмов.

Теплоемкость, как и другие характеристики системы, вычисляется с помощью усреднения по каноническому ансамблю, находящемуся в состоянии равновесия при некоторой температуре. Канонический ансамбль состоит из большого числа систем, каждая из которых представляет собой некоторую конфигурацию атомов кристаллической решетки. Такой ансамбль должен быть построен по принципам статистической механики Гиббса, согласно которым все системы ансамбля должны быть равновероятны. Из принципов Гиббса следует, что вероятность пребывания системы в некотором состоянии определяется только энергией состояния, а именно, вероятность принадлежности системы с энергией  $E$  к каноническому ансамблю пропорциональна величине  $\exp(-E/kT)$ , где  $k$  — постоянная Больцмана. Поэтому наиболее вероятными будут конфигурации атомов с наименьшей энергией (системы с большой энергией также возможны, но менее вероятны). Равновесию в каноническом ансамбле соответствует наиболее вероятное распределение систем по состояниям (расположениям спинов атомов). Следовательно, для построения канонического ансамбля необходимо определить системы, наиболее вероятные для заданного значения температуры. Таким образом, поиск равновесного канонического ансамбля приводит к задаче оптимизации с целевой (минимизируемой) функцией  $E$ . Внешние и внутренние параметры в этой задаче те же, что и в задаче об основных состояниях.

Для получения хотя бы одной системы равновесного канонического ансамбля необходимо перебрать все возможные ее состояния, соответствующие различным расположениям спинов атомов в решетке. Для каждого состояния необходимо вычислить энергию, согласно которой оно может принадлежать или не принадлежать ансамблю.

Таким образом, рассмотренные выше оптимизационные задачи физики твердого тела не могут быть решены полиномиальными алгоритмами и, тем самым, относятся к классу NP-сложных задач.

**2.3. Алгоритм Метрополиса.** Одним из часто употребляемых в физике твердого тела недетерминированных алгоритмов является метод Монте-Карло. В этом методе решение получается за приемлемое время с помощью исследования случайно выбранных состояний системы [9]. На каждом шаге строится некоторая система канонического ансамбля при условии выполнения принципов статистической механики. На этапе инициализации расположение элементов в системе задается случайным образом.

Пусть система, построенная на шаге  $n$ , соответствует точке в фазовом пространстве; тогда новой системе в ансамбле на шаге  $n + 1$  соответствует перемещение в соседнюю точку фазового пространства. Канонический ансамбль состоит из всех состояний от  $n = 1$  до  $n = m$ , через которые прошла система ( $m$  — количество систем в ансамбле — должно быть большим числом). Существует большое число возможностей выбора следующего состояния на шаге  $n' = n + 1$  исходя из некоторого состояния системы на шаге  $n$ . Выбор производится случайно, и в результате осуществляется переход в одно из множества состояний. Очевидно, что после небольшого числа шагов ансамбль может содержать очень большое разнообразие систем. Необходимо построить канонический ансамбль так, чтобы частота появления систем, образующих ансамбль, была пропорциональна вероятности Больцмана

$$p = \exp(-\Delta E/kT), \tag{2}$$

где  $\Delta E = E_{n+1} - E_n$ . Такое распределение по ансамблю можно получить, решая вопрос, является ли вновь полученная система допустимой или нет. Если  $\Delta E < 0$ , т.е. перемещение приводит к системе с меньшей энергией, то система включается в ансамбль. С другой стороны, если  $\Delta E > 0$ , т.е. образовалась система с большей энергией, то система включается в ансамбль лишь с вероятностью Больцмана  $p$ . Для этого выбирается случайное число  $r \in (0, 1)$ , и если  $r < p$ , то новая система присоединяется к ансамблю; если  $r > p$ , то новая система отбрасывается и к ансамблю добавляется еще один раз старая система. На этом данный шаг процесса завершен; на следующем шаге рассматривается другое состояние системы. Очевидно, что чем больше систем содержится в ансамбле, тем лучше будет решение. В общем случае метод усреднения по каноническому ансамблю, построенному с помощью алгоритма Монте-Карло, значительно превосходит по быстродействию прямой метод усреднения по времени с использованием детерминистических уравнений.

Таким образом, алгоритмы группы Монте-Карло состоят в итерационном выполнении двух этапов: создание случайным образом решения (пробного решения) и проверка его на наличие критерия, по которому можно установить, что данное решение правильное. Одним из алгоритмов этой группы является алгоритм, предложенный Метрополисом [10]. Алгоритм Метрополиса используется при моделировании фазовых переходов. Критерий правильного решения в алгоритме Метрополиса зависит от управляющего параметра — “температуры”. В алгоритме целевую функцию обычно называют энергией, а принятие решения осуществляется согласно вероятности Глаубера

$$p = \frac{1}{1 + \exp(-\Delta E/kT)}. \tag{3}$$

Общая схема алгоритма Метрополиса может быть представлена следующим образом.

1. Инициализация. Моделируем вектор (строку) решения со случайными значениями элементов. Вычисляем значение целевой функции — энергии  $E$ .
2. Случайным образом выбираем компоненту вектора и меняем ее значение. Вычисляем энергию получившегося решения  $E'$ , при этом энергия системы меняется на величину  $\Delta E = E' - E$ .
3. Если изменение компоненты приводит к понижению энергии, то новая конфигурация системы (вектор решения), полученная на предыдущем шаге, принимается  $E = E'$ ; иначе она принимается с вероятностью (3).
4. Этапы 2 и 3 повторяются до тех пор, пока не достигается минимум (стабилизация) энергии системы.
5. Изменяем  $T$  в формуле (3), повторяем все еще раз, но с новой начальной конфигурацией атомов.
6. Усредняем по многим опытам.

Обычно алгоритм Метрополиса начинает работу с высоких температур и постепенно их понижает, поэтому процесс решения иногда называют отжигом.

Пусть магнитные свойства кристаллической решетки описываются моделью Изинга с абсолютным значением спина атомов, равным единице. Тогда на этапе инициализации будет сгенерирована кристаллическая решетка со случайным распределением спинов атомов, значения которых равны 1 и  $-1$ . В качестве целевой функции будет рассматриваться энергия, вычисленная по формуле (1), а новые конфигурации будут получаться с помощью изменения значения спина случайно выбранного атома на противоположное значение.

Результатом работы алгоритма Метрополиса будет зависимость энергии системы от температуры, причем параллельно энергии можно вычислять и магнитные характеристики, например такие, как намагниченность. Температуру фазового перехода можно определить по полученным данным несколькими способами, например, как максимум теплоемкости, рассчитанной по формуле  $c = \frac{\langle E^2 \rangle - \langle E \rangle^2}{kT^2}$  или  $c = \frac{\partial E}{\partial T}$ .

Алгоритм Метрополиса часто используют в NP-сложной задаче коробейника. В этой задаче задан набор городов и расстояние между любыми двумя из них (иногда вместо расстояния рассматривают стоимость проезда). Коробейник путешествует из города в город, посещая города только по одному разу, и в итоге возвращается в исходный город. Пусть совокупность городов образует множество вершин полного графа. Тогда каждому ребру приписывается вес, равный расстоянию между двумя соответствующими городами. Если между городами не существует пути, то соответствующему ребру приписывается заведомо большой вес ( $\infty$ ). Необходимо определить кратчайший путь на взвешенном графе, что соответствует наименьшему пути, пройденному коробейником за время путешествия по всем городам. Аналогичная задача возникает, например, при выборе кратчайшего пути распространения информации по всем узлам компьютерной сети.

Применение детерминированного метода перебора всех потенциально реализуемых посещений невозможно при большом количестве городов. Пусть требуется посетить  $n$  городов; тогда выходной вектор  $\mathbf{Y}$ , имитирующий последовательность посещения городов, будет состоять из  $n + 1$  целых чисел, соответствующих номерам городов, причем первый и последний компоненты вектора будут одинаковыми (условие возвращения в исходный город). Для конкретного вектора  $\mathbf{Y}$  можно вычислить энергию (в данном случае суммарное расстояние, пройденное коробейником). Тем самым, становится возможным применение алгоритма Метрополиса по указанной выше схеме.

**2.4. Генетические алгоритмы.** Генетический алгоритм впервые был предложен Холландом [1]. Впоследствии этот алгоритм, получивший название “репродуктивный план Холланда”, активно использовался в качестве базового в эволюционных вычислениях. В дальнейшем генетические алгоритмы были развиты в работах Гольдберга [11], Де Йонга (De Jong) и их учеников. Генетические алгоритмы — это адаптивные методы поиска, которые в последнее время используются для решения задач функциональной оптимизации. В них применяются аналоги механизмов генетического наследования и естественного отбора, при этом используется биологическая терминология в упрощенном виде и сохраняются основные понятия линейной алгебры.

*Вектор* — упорядоченный набор чисел, называемых *компонентами* вектора. Поскольку вектор можно представить себе в виде строки его координат, то понятия вектора и строки считаются идентичными.

*Булев вектор* — вектор, компоненты которого принимают значения из двухэлементного (булева) множества, например  $\{0, 1\}$  или  $\{-1, 1\}$ .

*Хеммингово расстояние* (или просто *расстояние*) — используется для булевых векторов и равно числу различающихся в обоих векторах компонент.

*Хеммингово пространство* — пространство булевых векторов с введенным на нем расстоянием (метрикой) Хемминга. В случае булевых векторов длины  $n$  это пространство представляет собой множество вершин  $n$ -мерного гиперкуба с хемминговой метрикой. Расстояние между двумя вершинами определяется длиной кратчайшего соединяющего их пути, измеренной вдоль ребер.

*Хромосома* — вектор (или строка) из каких-либо чисел. Если этот вектор представлен бинарной строкой из нулей и единиц, например 1010011, то такой вектор получен с использованием либо *двоичного кодирования*, либо *кода Грея*. Каждая позиция (бит) хромосомы называется *геном* [12].

*Индивидуум* (генетический код, особь) — набор хромосом (вариант решения задачи). Обычно особь здесь состоит из одной хромосомы, поэтому в дальнейшем особь и хромосома — идентичные понятия.

*Репродукция* (селекция) — операция отбора наиболее приспособленных особей для участия в воспроизводстве (репродукции) новых особей-потомков.

*Кроссинговер* (кроссовер) — операция, при которой две хромосомы обмениваются своими частями, например 1100&1010  $\rightarrow$  1110&1000.

*Мутация* — случайное изменение одной или нескольких позиций в хромосоме, например 1010011 → 1010001.

*Инверсия* — изменение порядка следования битов в хромосоме или в ее фрагменте, например 1100 → 0011.

*Популяция* — совокупность индивидуумов.

*Пригодность* (приспособленность) — критерий или функция, экстремум которой следует найти.

*Локус* — позиция гена в хромосоме.

*Аллель* — совокупность подряд идущих генов.

*Эпистаз* — влияние гена на приспособленность индивидуума в зависимости от значения гена, присутствующего в другом месте. Ген считают эпистатическим, когда его присутствие подавляет влияние гена в другом локусе. Эпистатические гены из-за их влияния на другие гены иногда называют ингибирующими. Подавление проявления гена неаллельным ему геном называется гипостазом, а сам подавляемый ген — гипостатическим.

Рассмотрим основные этапы работы генетического алгоритма (рис. 1).

1. Генерируем случайным образом начальную популяцию из  $n$  хромосом.
2. Вычисляем для каждой хромосомы ее приспособленность.
3. Выбираем пару хромосом-родителей с помощью одного из способов отбора.
4. Проводим кроссинговер двух родителей с вероятностью  $p_c$ , производя двух потомков.
5. Проводим мутацию потомков с вероятностью  $p_m$ .
6. Повторяем этапы 3–5, пока не будет сгенерировано новое поколение популяции, содержащее  $n$  хромосом.
7. Повторяем этапы 2–6, пока не будет достигнут критерий окончания процесса.

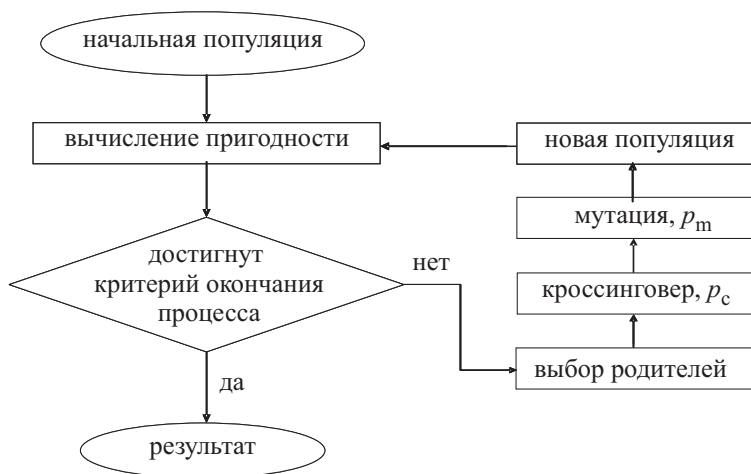


Рис. 1. Схема простого генетического алгоритма

Отбор хромосом в новую популяцию может происходить по-разному, например, хромосомы-потомки могут заменить менее приспособленные хромосомы предыдущей популяции. Процессы кроссинговера, мутации и отбора хромосом в новую популяцию повторяются несколько раз. В итоге в получившейся популяции все хромосомы будут иметь практически одинаковое значение приспособленности (т.е. популяция сойдется к одному шаблону строк) и, тем самым, будет найден вектор решения, отвечающий минимуму функции приспособленности. Для построения эффективного генетического алгоритма необходимо правильно выбрать кодирование чисел в строки, типы операторов создания родительских пар, кроссинговера, мутации и отбора в новую популяцию.

**2.5. Задачи физики твердого тела, в которых генетические алгоритмы уже применялись.**

**2.5.1. Определение энергии основного состояния примеси.** Генетические алгоритмы применялись для определения энергии основного состояния примесей в сферической квантовой точке радиуса  $R$  с бесконечным граничным потенциалом [13]  $V(r) = \begin{cases} 0, & r < R, \\ -\infty, & r > R. \end{cases}$  Известно, что любая квантовомеханическая система описывается некоторой волновой функцией, в соответствии с которой вычисляются значения энергии основных состояний. В рассматриваемой задаче квантовомеханическая система описывается волновой функцией

$$\psi(r) = \begin{cases} 0, & r > R, \\ \frac{A}{r} \sin(kr) \exp(-\lambda r), & r < R. \end{cases} \tag{4}$$

где  $k = \pi/R$  и  $\lambda$  — изменяемый параметр. Пусть  $\psi^*(r)$  — комплексно-сопряженная функция для волновой функции. Амплитуда  $A$  вычисляется из условия нормировки для каждого значения параметра  $\lambda$ :

$$\int \psi^*(r)\psi(r) dr = 1, \tag{5}$$

Для нахождения минимума энергии необходимо решить следующую задачу минимизации:

$$E = \min_{\lambda} \frac{\int_Z \psi^*(\lambda, r) \widehat{H} \psi(\lambda, r) dz}{\int_Z \psi^*(\lambda, r) \psi(\lambda, r) dz}.$$

Для решения этой задачи рассматривались два генетических алгоритма. Идея первого алгоритма заключалась в том, что в качестве внутреннего параметра рассматривался параметр  $\lambda$ . В таком генетическом алгоритме в качестве особи выбиралось кодированное в двоичной системе значение  $\lambda$ . На этапе инициализации создавалась популяция из ста особей, соответствующих случайно выбранному значению  $\lambda$ . Для всех особей вычислялась энергия и значение функции приспособленности  $\text{Fitness}(\psi_i) = \exp\left(-\beta(E_i - |\langle E \rangle|)\right)$ , где  $\beta$  — константа. Процесс репродукции заключался в отборе особей для создания новой популяции. В качестве оператора репродукции применялся метод рулеточного отбора. Реализация рулеточного отбора родительских особей заключается в следующем: каждой особи в популяции соответствует зона на колесе рулетки, пропорциональная функции приспособленности [12]. Тогда при повороте колеса рулетки каждый элемент имеет некоторую вероятность выбора для репродукции, причем особь с большим значением функции приспособленности имеет большую вероятность для выбора. Для создания новых особей использовался одноточечный кроссинговер, в котором для родительской пары случайным образом выбирался один ген разреза, и затем между особями происходил обмен полученными фрагментами справа относительно точки разреза. В результате одного кроссинговера получаются два новых значения параметра  $\lambda$ , записанных в двоичном коде. Полученные особи мутируют с очень низкой вероятностью. Применение мутации заключается в инвертировании (изменении значения 1 на 0 и наоборот) одного гена и является особенно необходимым в тех случаях, когда система сходится к одному из локальных минимумов вместо глобального. Мутации с большой вероятностью не используются, так как они нарушают сходимость алгоритма.

Второй генетический алгоритм основывался на оптимизации волновой функции. В генетический код особи записывалась волновая функция  $\psi(r)$ . Популяция состояла из ста особей. На этапе инициализации для случайно сгенерированных значений  $\lambda$  вычислялись соответствующие значения волновой функции согласно формуле (4). Затем значение волновой функции нормализовывалось по формуле (5). Значения энергии особей вычислялись по формуле  $E_k = \int_Z \psi^*(\lambda, r) \widehat{H} \psi(\lambda, r) dz$ . В качестве оператора репродукции выбирался рулеточный метод отбора, и отобранные особи участвовали в кроссинговере. Новые волновые функции создавались по следующему правилу:

$$\psi'_1(r) = \psi_1(r)S(r) + \psi_2(r)[1 - S(r)], \quad \psi'_2(r) = \psi_2(r)S(r) + \psi_1(r)[1 - S(r)],$$

где  $S(r)$  — размытая ступенчатая функция, определяемая формулой  $S(r) = \frac{1}{2} \left[ 1 + \text{th} \left( \frac{r - r_0}{w} \right) \right]$ , где  $r_0$  — случайное число на интервале  $(0, R)$  и  $w$  — параметр, определяющий резкость кроссинговера [13]. Множитель  $1/w^2$  соответствует степени размытости ступеней: чем меньше  $w$ , тем больше график гиперболического тангенса похож на ступенчатую функцию. Поэтому параметр  $w$  называется резкостью кроссинговера. Вероятность кроссинговера в данной задаче принималась равной  $P_c = 0.95$ . Для созданных в результате кроссинговера особей вычислялось значение функции приспособленности (т.е. энергии). На следующем шаге к особи исходной популяции с вероятностью  $P_m = 0.05$  применялся оператор мутации: генерировалось произвольное значение параметра  $\lambda$  и соответствующая этому параметру волновая функция добавлялась к существующему значению, т.е.  $\psi'_k(z) = \psi_k(z) + \psi_M(z)$ . Из полученных в результате предыдущих действий особей выбирались сто наиболее приспособленных особей в новую популяцию. Сходимость алгоритма достигается уже за двадцать итераций.

Генетические алгоритмы работают в данной задаче гораздо быстрее, чем детерминированные методы. Что касается эффективности рассмотренных выше вариантов генетических алгоритмов, то следует отметить, что алгоритм, работающий с волновыми функциями, оказывается точнее, чем алгоритм, оперирующий со значениями параметра  $\lambda$ .

Подобный квантовый генетический алгоритм, оперирующий с волновыми функциями, применялся при самосогласованных вычислениях гетеропереходов в полупроводниках [14].

**2.5.2. Применение генетических алгоритмов к задаче поиска основных состояний спиновых стекол.** Генетические алгоритмы использовались для нахождения основных состояний спиновых стекол [6]. Особенностью таких алгоритмов является строение операторов кроссинговера и селекции. В качестве функции приспособленности выбирают функцию энергии (1).

Целью генетического алгоритма является определение конфигурации спиновой решетки, при которой значение энергии минимально. Начальная популяция состоит из бинарных хромосом, гены которых случайным образом принимают значения  $-1$  и  $+1$ . Каждая хромосома соответствует определенному расположению спинов в решетке. Приспособленностью хромосомы является значение полной энергии, соответствующей конфигурации атомов. Перед отбором родительских пар популяцию сортируют по возрастанию значений приспособленности. В кроссинговере участвует первая более приспособленная половина всех хромосом. Родительские пары формируются из соседних хромосом. Каждая выбранная хромосома может состоять в одной–двух родительских парах. Количество пар обуславливается заданной вероятностью кроссинговера. Получающиеся при кроссинговере потомки заменяют хромосомы во второй половине популяции.

Хартманн [6] предлагает в первом поколении при кроссинговере не скрещивать родительские хромосомы, а дублировать их в качестве потомков. Таким образом после первого применения оператора кроссинговера из начальной популяции выбрасываются заведомо непригодные хромосомы. Начиная со второго поколения, для получения новых особей применяется триадный кроссинговер для соседних по популяции особей–родителей. Триадный кроссинговер использует бинарную строку–маску для скрещивания хромосом. Маска представляет собой хромосому со случайным распределением спинов. Потомки строятся побитово: если  $i$ -е гены первой хромосомы и маски совпадают, то  $i$ -м геном первого (второго) потомка будет  $i$ -й ген первого (второго) родителя, иначе таким геном станет  $i$ -й ген второго (первого) родителя. Тем самым триадный кроссинговер позволяет получить новое спиновое состояние и сократить количество мутаций. Другие виды кроссинговера менее пригодны, так как при их использовании получение новых комбинаций генов для больших хромосом затруднено. После кроссинговера потомки подвергаются мутации с очень маленькой вероятностью. Отбор в новое поколение происходит согласно формуле Глаубера (3) или Больцмана (2).

Описанный выше генетический алгоритм применяется на сравнительно маленьких решетках с большим количеством итераций. Хартманн [6] начинал с большой популяции и, используя элитизм для отбора хромосом, в каждом поколении усекал ее в два раза, получая на выходе единственную хромосому (основное спиновое состояние).

**3. Проблемы, возникающие при применении генетических алгоритмов в задачах физики твердого тела.** При применении генетических алгоритмов в задаче о фазовых переходах возникает несколько проблем, которые затрудняют создание большого ансамбля с больцмановским распределением. Первая из них связана с возникновением зеркального распределения спинов в хромосомах, т.е. таких конфигураций, в которых атомы со спинами разных знаков располагаются на одинаковых позициях.

Объяснить существо проблемы удобнее всего с использованием формализма шим. Шаблон (шимой, схемой) в генетическом алгоритме называется подмножество хромосомы (некоторая совокупность подряд идущих генов), содержащее знаки “\*”. На основании одного шаблона может быть построено некоторое подмножество дочерних хромосом, в которых знак “\*” заменяется значением 0 или 1, если хромосома кодируется бинарной строкой [12]. Порядок шаблона — число закрепленных позиций (в двоичном алфавите это число нулей и единиц, представленных в шаблоне). Например, для шаблона  $H = (0 * 1 * *)$  порядок шаблона  $o(H) = 2$ , а для шаблона  $H = (* * 1 * *)$  —  $o(H) = 1$ . Для каждого шаблона можно оценить его приспособленность. Согласно правилу репродукции Холланда шаблон со значением функции приспособленности выше среднего копируется в следующую популяцию, а шаблон с функцией приспособленности ниже среднего — отбрасывается. Предположим, что в хромосоме выделены несколько шаблонов высокого порядка.

В модели Изинга хромосома кодируется бинарной строкой, состоящей из 1 и  $-1$ , а в качестве функции приспособленности полагается полная энергия системы. Тогда наиболее приспособленные шаблоны будут состоять из одинаковых генов. Рассмотрим два однородных (или однотипных, т.е. содержащих только 1 или  $-1$ ) шаблона одинакового порядка, но отличающихся тем, что на тех позициях, на которых у первого шаблона стоят 1, у второго — стоят  $-1$ . Энергия для обоих шаблонов будет одинакова. Если эти шаблоны высокого порядка, то они обязательно попадут в новую популяцию. В новой популяции они могут:

- стать компонентами одной хромосомы; тогда приспособленность хромосомы ухудшится (энергия увеличится) в связи с взаимодействием противоположных по знаку спинов;
- рассматриваемые шаблоны могут быть разрушены в результате кроссинговера — обмен фрагмен-

тами таких шаблонов приведет к созданию непригодных шаблонов высокого порядка (рис. 2).

Существование симметричных шаблонов замедляет сходимость алгоритма и в итоге приводит к ложному результату. Чтобы избежать подобной ситуации, можно рассматривать в ферромагнитных системах в качестве функции приспособленности полную энергию системы с учетом внешнего магнитного поля  $H$ , позволяющего снять вырождение системы:

$$E = - \sum_{ij} J_{ij} \sigma_i \sigma_j - H \sum_i \sigma_i. \quad (6)$$

При использовании данной функции приспособленности шаблоны с отрицательными значениями генов становятся непригодными и, следовательно, отбрасываются при формировании нового поколения. Другим вариантом функции приспособленности может служить функция  $E = - \sum_{ij} J_{ij} \sigma_i \sigma_j \times H \sum_i \sigma_i$ . Ее применение также влечет удаление из популяции непригодных шаблонов с тем отличием, что это происходит гораздо быстрее, чем в первом случае (разность произведений больше, чем разность сумм).

Вторая проблема связана с преждевременной сходимостью. В этом случае в генетическом алгоритме перестают создаваться принципиально новые решения и шаблоны, ведущие к увеличению приспособленности (т.е. алгоритм сходится к некоторому виду хромосомы-решения). Вообще, принципиально новые хромосомы могут быть созданы посредством применения мутаций. С другой стороны, мутации разрушают приспособленные шаблоны хромосом. В связи с этим необходимо найти способ управления количеством мутаций для задачи с определенным размером кристаллической решетки. Это делается путем определения динамических мутаций, способных менять свою вероятность в процессе решения задачи [12].

Третья проблема связана с увеличением числа пересчета энергии, вызванным наличием кроссинговера, при котором создаются новые хромосомы. Следует заметить, что технически пересчет энергии при мутации хромосомы проще, чем при кроссинговере. Следовательно, для решения поставленной задачи генетическим алгоритмом необходимо разработать оператор кроссинговера, способный за несколько поколений привести популяцию к глобальному минимуму, оставив при этом Больцмановскую формулу в операторе селекции (вынос температуры в оператор мутации сделает алгоритм аналогичным алгоритму Метрополиса).

**4. Сравнение алгоритма Метрополиса и генетических алгоритмов. Алгоритм Метрополиса как “бесплодный” генетический алгоритм.** При сравнении этапов работы генетического алгоритма и алгоритма Метрополиса можно заметить, что последний является бесплодным (не использующим операторы рекомбинации и кроссинговера) генетическим алгоритмом со 100 % мутациями на основе формулы Больцмана. Алгоритм Метрополиса, в отличие генетического алгоритма, работает только с одной особью в популяции. На каждой итерации алгоритма Метрополиса единственная особь копируется в особь-потомок. Так же как и в генетическом алгоритме, потомок подвергается мутации. Однако в отличие от генетического алгоритма в алгоритме Метрополиса мутации происходят постоянно, причем у потомка мутирует только один ген, после чего вычисляется значение приспособленности (энергии). Принятие потомка в новую популяцию (замена им родительской особи) в алгоритме Метрополиса происходит так же, как и в генетическом алгоритме — по методу элитизма. То, что алгоритм Метрополиса является частным случаем генетического алгоритма, следует из таблицы сравнения их итераций (табл. 1).

**5. Сравнительный анализ результатов применения алгоритма Метрополиса и генетических алгоритмов на примере двумерной модели Изинга.** Мы тестировали генетический алгоритм и алгоритм Метрополиса на двумерной квадратной решетке с линейным размером 32, т.е. всего рассматривалось  $32^2$  атомов. Использовалась модель Изинга. Абсолютное значение спинов атомов, как и константа обменного взаимодействия, принималось равным единице. В качестве целевой функции рассматривалась функция (6). В обоих алгоритмах проводился отжиг с начальной температурой 500 К и конечной температурой 20 К, шаг изменения температуры — 20 К. Магнитное поле в формуле (6) принималось равным

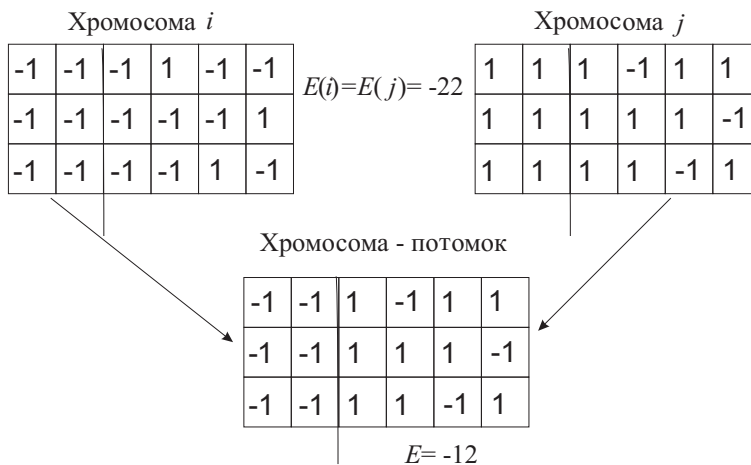


Рис. 2. Разрушение приспособленных симметричных шаблонов в результате кроссинговера



Таблица 1

Сравнительный анализ этапов алгоритма Метрополиса и генетического алгоритма, применяемых к построению канонического ансамбля для заданной температуры

Алгоритм Метрополиса	Генетический алгоритм
Создание начальной конфигурации атомов. Расположение спинов задается случайным образом. Вычисляется энергия для созданной системы атомов.	Создание начальной популяции из $N$ систем атомов (хромосом), где $N$ — размер популяции. На этом этапе случайным образом создается $N$ массивов, каждый из которых соответствует некоторому состоянию кристаллической решетки. Для всех хромосом вычисляется значение функции приспособленности (энергии).
Итерация алгоритма Метрополиса	Итерация генетического алгоритма
	Рекомбинация, кроссинговер, получение потомков.
Изменение значения случайно выбранного спина, т.е. меняется значение случайно выбранного элемента массива. Для полученной новой конфигурации вычисляется изменение энергии $\Delta E$ .	Мутации потомков с вероятностью $p_m$ . В каждом двумерном массиве, соответствующем какому-либо потомку, согласно вероятности $p_m$ меняется группа элементов. Количество элементов в группе задается некоторым образом, например, оно может быть определено как величина $p_m \times \text{Size}$ , где $\text{Size}$ — количество элементов в массиве. После применения оператора мутации вычисляется приспособленность (энергия) всех потомков. Следует заметить, что при низкой вероятности какой-либо потомок может не мутировать.
Новая конфигурация принимается в том случае, если ее энергия понизилась по сравнению с энергией предыдущего состояния или если вероятность ее принятия пропорциональна вероятности Больцмана.	Отбор в новое поколение методом элитизма. В новую популяцию попадают конфигурации или наиболее приспособленные, или с вероятностью выживания, пропорциональной вероятности Больцмана.
Если канонический ансамбль не построен, то переходим к новой итерации.	Если в популяции, имитирующей канонический ансамбль, не достигнуто состояние равновесия (сходимость к некоторому состоянию), то переходим к новой итерации.
Завершение процесса. Получение итоговой решетки спинов, являющейся результатом усреднения по каноническому ансамблю.	Завершение процесса. Получение итоговой популяции — совокупности решеток спинов, по которым путем усреднения получается искомое состояние системы.

единице. Для построения канонического ансамбля в алгоритме Метрополиса выполнялось от 500 до 1500 итераций. Алгоритм Метрополиса прекращал работу, когда была достигнута 1% точность вычислений энергии (значения энергии создаваемых конфигураций решетки отличались друг от друга не более, чем на 0.01 К) или были выполнены все 1500 итераций. Генетический алгоритм выполнял 150 шагов формирования популяций по следующей схеме:

- согласно заданной вероятности кроссинговера создавалось определенное количество хромосом-потомков; новые хромосомы заменяли старые, которые не участвовали в скрещивании; вероятность кроссинговера составляла 80%;
- мутирование хромосом-потомков заключалось в том, что заданное вероятностью мутации количество генов принимало положительное значение; вероятность мутации 1%;
- новая популяция хромосом создается из случайной выборки между старыми и новыми хромосомами с вероятностью Больцмана.

Популяция в генетическом алгоритме состояла из 10 хромосом, каждая из которых соответствовала одной конфигурации алгоритма Метрополиса. Начальные конфигурации атомов генерировались случай-

ным образом. Результаты работы алгоритмов представлены на рис. 3–5.

Таблица 2

Сравнительный анализ сложности алгоритмов на основе их применения к двумерной модели Изинга

Параметр	Алгоритм Метрополиса	Генетический алгоритм
Полное время работы при изменении температуры от 500 К до 20 К с шагом $\Delta T = -20$ К	1675 секунд	3280 секунд
Время, затраченное на вычисление энергии	1059 секунд	2823 секунд

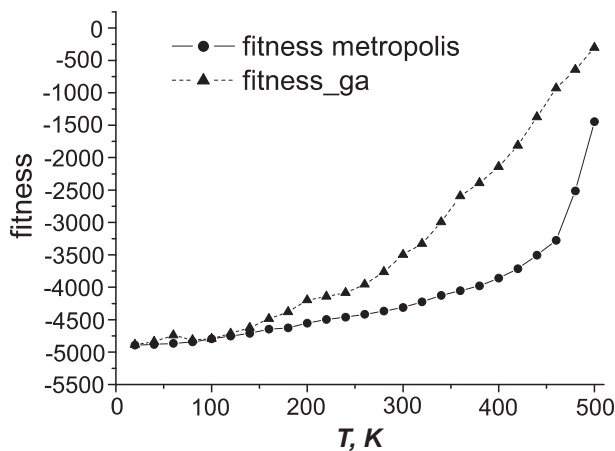


Рис. 3. Зависимость целевой функции от температуры

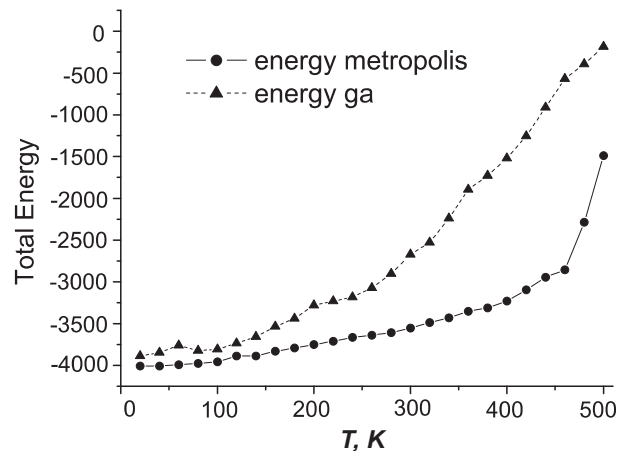


Рис. 4. Зависимость полной энергии от температуры

В табл. 2 представлено распределение времени выполнения алгоритмов. Из нее видно, что генетический алгоритм находит решение медленнее, чем алгоритм Метрополиса. В данном генетическом алгоритме применялись направленные мутации, которые переворачивали спины в одинаковом положительном направлении. Такие мутации увеличивают приспособленность хромосомы. Случайные мутации могут быть деструктивными (т.е. в результате мутаций возможны разрушения пригодных шаблонов). В связи с последним фактом применение мутаций с произвольным изменением гена ведут к очень длительным вычислениям. Применение значительного количества направленных мутаций, в которых значение гена заменяется определенной величиной, ведет к преждевременной сходимости.

За одну итерацию в генетическом алгоритме происходит гораздо больше вызовов функции, вычисляющей значение энергии, чем в алгоритме Метрополиса. Генетические алгоритмы работают эффективнее алгоритма Метрополиса на небольших решетках за счет вставок строительных блоков и оператора кроссинговера. Строительные блоки — это высокоприспособленные фрагменты хромосом, разрушение которых не рекомендуется. В рассматриваемом варианте генетического алгоритма они представлены направленными мутациями. При увеличении размера решетки длина хромосом в генетическом алгоритме возрастает;

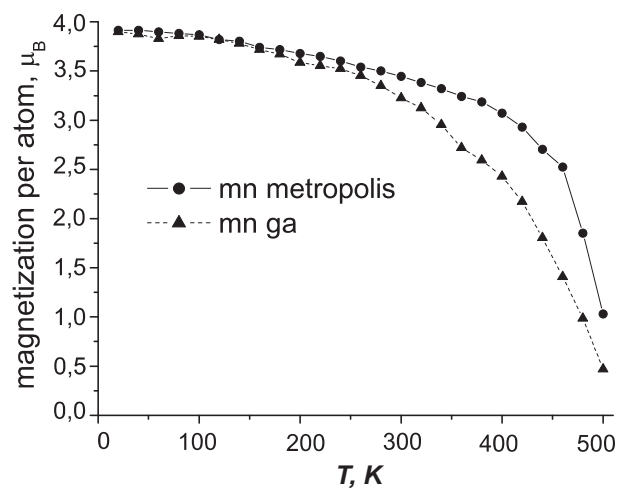


Рис. 5. Зависимость намагниченности (на атом) от температуры

следовательно, становится необходимым увеличение числа итераций алгоритма, в каждой из которых ведется пересчет энергии для всех хромосом популяции. Таким образом, количество пересчетов энергии возрастает в несколько раз, в то время как в алгоритме Метрополиса количество пересчетов энергии в одной итерации остается прежним, поскольку рассматривается только одна хромосома. Таким образом, время, затраченное на итерации алгоритма Метрополиса для больших решеток, намного меньше, чем время, затраченное на итерации генетического алгоритма.

**6. Выводы.** Рассмотренные оптимизационные однокритериальные задачи физики твердого тела основаны на нахождении спиновой конфигурации атомов с наименьшей энергией при заданной температуре и являются NP-сложными. Для их решения возможно применение таких недетерминированных алгоритмов, как алгоритм Метрополиса и генетические алгоритмы. Алгоритм Метрополиса по своей сути является генетическим алгоритмом со 100 % мутациями и с отсутствием кроссинговера, т.е. таким генетическим алгоритмом, в котором:

- популяция состоит из одной хромосомы;
- хромосома-потомок создается путем мутации случайно выбранного одного гена;
- из двух получившихся хромосом происходит отбор в новую популяцию: хромосома выбирается, если ее энергия ниже или если вероятность ее выживания пропорциональна вероятности Больцмана.

Алгоритм Метрополиса не используется в многоэкстремальной задаче об основных состояниях, так как его возможности ограничиваются определением лишь одного глобального минимума. В этих целях применяются генетические алгоритмы. Алгоритм Метрополиса применим в задаче определения температуры фазовых переходов. Применение других модификаций генетического алгоритма для этой задачи менее эффективно в связи с существованием ряда проблем, вызванных структурой хромосомы.

#### СПИСОК ЛИТЕРАТУРЫ

1. *Holland J.H.* Adaptation in natural and artificial systems. Ann Arbor: University of Michigan Press, 1975.
2. *Michalewicz Z.* Genetic algorithms + data structures = evolution programs. New York: Springer-Verlag, 1996.
3. *Mitchell M.* An introduction to genetic algorithms. Cambridge: MIT Press, 1996.
4. *Fogel D.B.* Evolutionary computation: towards a new philosophy of machine intelligence. Piscataway: IEEE Press, 1995.
5. *Koza J.R.* Genetic programming. Cambridge: MIT Press, 1992.
6. *Hartmann A.K., Rieger H.* Optimization algorithms in physics. Berlin: Wiley-VCH, 2002.
7. *Макконнелл Дж.* Основы современных алгоритмов. М.: Техносфера, 2004.
8. *Батищев Д.И.* Генетические алгоритмы решения экстремальных задач. Воронеж: Воронеж. гос. техн. ун-т, 1995.
9. *Поттер Д.* Вычислительные методы в физике. М.: Мир, 1975.
10. *Metropolis N., Rosenbluth A.W., Rosenbluth M.N., Teller A.H., Teller E.* Equation of state calculations by fast computing machines // J. Chem. Phys. 1953. **21**, N 6. 1087–1092.
11. *Goldberg D.* Genetic algorithms in search, optimization, and machine learning. Boston: Addison-Wesley, 1989.
12. *Курейчик В.А., Курейчик В.В., Гладков В.М.* Генетические алгоритмы. М.: Физматлит, 2006.
13. *Şafak H., Şahin M., Gülveren B., Tomak M.* Efficiency of genetic algorithm and determination of ground state energy of impurity in a spherical quantum dot // Int. J. of Modern Physics C. 2003. **14**, N 6. 775–784.
14. *Şahin M., Tomak M.* Self-consistent calculation of semiconductor heterojunctions by using quantum genetic algorithm // Int. J. of Modern Physics B. 2002. **16**, N 26. 3883–3893.

Поступила в редакцию  
19.01.2007