

На правах рукописи

Мальсагов Магомед Юсупович

ПРИМЕНЕНИЕ ДИСКРЕТИЗАЦИИ ДЛЯ РЕШЕНИЯ ЗАДАЧ  
БИНАРНОЙ ОПТИМИЗАЦИИ С ПОМОЩЬЮ НЕЙРОННОЙ  
СЕТИ ХОПФИЛДА

05.13.01 – «Системный анализ, управление и обработка информации»  
(по математическим отраслям и информатике)

Диссертация на соискание ученой степени  
кандидата физико-математических наук

Научный руководитель:  
кандидат физико-математических наук  
Крыжановский Михаил Владимирович

# СОДЕРЖАНИЕ

<b>СОДЕРЖАНИЕ</b>	<b>2</b>
<b>ГЛАВА 1. ВВЕДЕНИЕ</b>	<b>3</b>
§1.1. АКТУАЛЬНОСТЬ ТЕМЫ.....	3
§1.2. НЕЙРОННЫЕ СЕТИ И МОДЕЛЬ ХОПФИЛДА .....	7
§1.3. ИССЛЕДОВАНИЯ И ПРИМЕНЕНИЯ НЕЙРОННОЙ СЕТИ ХОПФИЛДА .....	13
§1.4. ЦЕЛИ И ЗАДАЧИ ДИССЕРТАЦИОННОЙ РАБОТЫ.....	26
§1.5. ОСНОВНЫЕ ПОЛОЖЕНИЯ, ВЫНОСИМЫЕ НА ЗАЩИТУ .....	27
§1.6. МЕТОДЫ ИССЛЕДОВАНИЯ .....	28
§1.7. НАУЧНАЯ НОВИЗНА .....	29
§1.8. ПРАКТИЧЕСКАЯ ЦЕННОСТЬ.....	30
§1.9. АПРОБАЦИЯ РАБОТЫ И ПУБЛИКАЦИИ .....	31
§1.10. СТРУКТУРА И ОБЪЕМ ДИССЕРТАЦИИ.....	33
<b>ГЛАВА 2. ПРОЦЕДУРА ДИСКРЕТИЗАЦИИ</b>	<b>34</b>
§2.1. ПРОЦЕДУРА ДИСКРЕТИЗАЦИИ.....	35
§2.2. МАТРИЦЫ С РАВНОМЕРНЫМ РАСПРЕДЕЛЕНИЕМ ЭЛЕМЕНТОВ.....	43
§2.3. МАТРИЦЫ С НОРМАЛЬНЫМ РАСПРЕДЕЛЕНИЕМ ЭЛЕМЕНТОВ.....	48
§2.4. РАССТОЯНИЕ МЕЖДУ МИНИМУМАМИ .....	53
§2.5. ВЫВОДЫ.....	59
<b>ГЛАВА 3. АЛГОРИТМЫ ПОИСКА МИНИМУМОВ</b>	<b>62</b>
§3.1. ОДНОЭТАПНЫЙ АЛГОРИТМ.....	62
§3.2. ДВУХЭТАПНЫЙ АЛГОРИТМ.....	72
§3.3. БЫСТРОДЕЙСТВИЕ АЛГОРИТМОВ.....	77
§3.4. ВЫВОДЫ.....	83
<b>ГЛАВА 4. СХЕМЫ АЛГОРИТМОВ ПОИСКА</b>	<b>86</b>
§4.1. СХЕМА ОДНОЭТАПНОГО АЛГОРИТМА .....	86
§4.2. СХЕМА ДВУХЭТАПНОГО АЛГОРИТМА.....	88
§4.3. НЕЙРОННАЯ СЕТЬ ХОПФИЛДА С ДИСКРЕТИЗИРОВАННОЙ МАТРИЦЕЙ.....	90
§4.4. ОПИСАНИЕ ПРОГРАММНОЙ РЕАЛИЗАЦИИ.....	97
§4.5. ВЫВОДЫ.....	103
<b>ЗАКЛЮЧЕНИЕ</b>	<b>104</b>
<b>ПРИЛОЖЕНИЕ</b>	<b>106</b>
<b>СПИСОК ЛИТЕРАТУРЫ</b>	<b>123</b>

# ГЛАВА 1. ВВЕДЕНИЕ

## *§1.1. Актуальность темы*

Задачи дискретной оптимизации [1-8] широко распространены и встречаются практически во всех сферах человеческой деятельности. Многочисленные проблемы в математике, статистике, технике, науке, медицине и экономике могут рассматриваться как проблемы оптимизации. Задачей оптимизации является нахождение решения, которое удовлетворяет системе ограничений и максимизирует или минимизирует целевую функцию. Наиболее известными задачами дискретной оптимизации являются задача о рюкзаке, задача о коммивояжере, задача планирования вычислений для многопроцессорной системы, выбор оптимальной структуры автоматизированной системы управления и т.д. Решение этих задач связано с рядом трудностей. Например, полный перебор возможных решений, как правило, невозможен из-за большого объема вычислений. Из-за дискретности пространства решений неприменимы многие приемы, разработанные в математическом программировании, например, движение по направлению градиента, переход из одной точки допустимых решений в другую и т.д. Однако для большинства прикладных задач достаточно получить хорошее приближенное решение. За счет отказа от поиска точного решения часто удается построить простые алгоритмы для сложных задач. Поэтому на практике часто используются приближенные методы как альтернатива полному перебору [9-14].

Большое значение имеют вопросы эффективности алгоритмов решения задач дискретной оптимизации [1, 15]. Для дискретных задач не всегда пригодны критерии оценки эффективности алгоритмов оптимизации, используемые для непрерывных задач. Состояние теории в настоящее время не позволяет указать универсальные количественные, и даже качественные оценки эффективности вычислительных алгоритмов. Такие оценки дает лишь вычислительная практика.

При решении задач дискретной оптимизации характерным приемом является разбиение задачи на подзадачи или ветвление. Полученные подзадачи либо решают, либо, в свою очередь, применяют к ним процедуру ветвления. Решение всей задачи получается в результате анализа всех подзадач наиболее низкого уровня ветвления.

Основными методами дискретной оптимизации являются метод ветвей и границ, динамическое программирование, метод отсечений, генетические алгоритмы, нейросетевые алгоритмы.

В *методе ветвей и границ* [16] из рассматриваемой задачи получаются две подзадачи путем специального разбиения пространства решений и отбрасывания областей, заведомо не содержащих решения исходной задачи. В случае, когда целочисленные переменные являются булевыми, обычно используется аддитивный алгоритм (метод Балаша [17]). Метод ветвей и границ позволяет получать хорошие результаты, но только при малых размерностях задачи ( $N=50-60$ ) [18-21].

*Метод динамического программирования* [22-25] применяется для решения динамических, т.е. меняющихся во времени, или многошаговых оптимизационных задачах. Метод строится на основе принципа оптимальности: оптимальное поведение в многошаговом процессе обладает тем свойством, что, какими бы ни были решение, принятое на последнем шаге, и состояние процесса перед последним шагом, предыдущие решения должны составлять оптимальное относительно этого состояния поведение. Таким образом, метод динамического программирования можно представить в виде трех этапов:

1. Исходная задача разбивается на подзадачи меньшего размера.
2. Находится оптимальное решение подзадач, рекурсивно применяя к каждой подзадаче тот же трехэтапный алгоритм, пока решение подзадачи не будет очевидно.
3. Используются решения подзадач для формирования решения исходной задачи.

Важным достоинством метода динамического программирования является его предсказуемость, т.е. возможность теоретически оценить объем вычислительной работы и объем занимаемой памяти. К недостаткам метода относится потребность в большом объеме памяти, а также сложность его катастрофически возрастает с увеличением размерности задачи.

Исходной задачей *методов отсечений* [26-29], используемых для решения линейных целочисленных задач, является задача с ослабленными ограничениями, которая возникает в результате исключения требования целочисленности переменных. По мере введения специальных дополнительных ограничений, учитывающих требование целочисленности, область допустимых решений ослабленной задачи постепенно изменяется до тех пор, пока координаты оптимального решения не станут целочисленными. Вводимые дополнительные ограничения отсекают некоторые области допустимых решений, в которых отсутствуют точки с целочисленными координатами. Примером метода отсечений является метод Гомори [17], основным недостатком которого является его достаточно медленная сходимость.

В основе *генетических алгоритмов* [30] лежит принцип естественного отбора (выживания сильнейшего или наиболее приспособленного). Вкратце, стандартный генетический алгоритм можно описать следующим образом:

1. Формируется начальная популяция решений задачи (некоторый набор векторов).
2. После проверки всех векторов, выбираются два, стартуя из которых, система сошлась в состояния соответствующие двум лучшим минимумам (операция селекции).
3. К векторам применяется операция скрещивания, в результате которой, получается новый вектор.
4. К полученному вектору применяется операция мутации, т.е., над вектором производятся небольшие случайные модификации.

5. Решение добавляется в популяцию, при этом из нее удаляется решение, соответствующее худшему минимуму.
6. Шаги с 1-го по 5-й выполняются до тех пор, пока популяция не будет состоять из векторов доставляющих самые оптимальные решения, согласно определенным заранее критериям.

Существует большое число [31-38], всевозможных модификаций описанного выше генетического алгоритма. Все они в той или иной мере справляются с задачей комбинаторной оптимизации. Однако у этих алгоритмов есть существенный недостаток – время работы.

*Искусственные нейронные сети* [39-43] — математические модели, а также их программные или аппаратные реализации, построенные по принципу организации и функционирования биологических нейронных сетей — сетей нервных клеток живого организма. Это понятие возникло при изучении процессов, протекающих в мозге, и при попытке смоделировать эти процессы. ИНС позволяют быстро находить приближенные решения задач дискретной оптимизации больших размерностей ( $N > 1000$ ). Однако нейросети также критичны к требованиям оперативной памяти и вычислительной сложности.

Как следует из выше сказанного требования к оперативной памяти и вычислительной сложности остаются слабыми местами всех известных оптимизационных алгоритмов. Поэтому по сей день остается актуальной задача построения и модификации алгоритмов с целью увеличения их производительности и снижения требований на вычислительные ресурсы.

## ***§1.2. Нейронные сети и модель Хопфилда***

Теория искусственных нейронных сетей – одна из областей математики, существующая уже более полувека и активно развивающаяся по настоящее время. Создание искусственных нейронных сетей было вызвано попытками понять принципы работы человеческого мозга. Однако, в сравнении с человеческим мозгом, искусственные нейронные сети сегодня представляют собой весьма упрощенные абстракции. Основные результаты в этой области связаны с именами У. Маккалоха (W. McCulloch) [44], У. Питтса (W. Pitts) [44], Д. Хебба (D. Hebb) [45], Ф. Розенблатта (F. Rosenblatt) [46], М. Минского (M. Minsky) [47] и Дж. Хопфилда (J. Hopfield) [48]. Область применения нейронных сетей довольно обширна. С их помощью можно успешно решать такие задачи как распознавание образов, классификация, идентификация, прогнозирование, аппроксимация функций, оптимизация, управление сложными объектами и др.

Нейронные сети не программируются в привычном смысле этого слова, они обучаются. Возможность обучения — одно из главных преимуществ нейронных сетей перед традиционными алгоритмами. Технически обучение заключается в нахождении коэффициентов связей между нейронами. В процессе обучения нейронная сеть способна выявлять сложные зависимости между входными данными и выходными, а также выполнять обобщение. Это значит, что в случае успешного обучения сеть сможет вернуть верный результат на основании данных, которые отсутствовали в обучающей выборке, а также неполных и/или «зашумленных», частично искаженных данных.

Одной из известнейших моделей искусственной нейронной сети является модель Хопфилда, предложенная в 1982 году американским ученым – специалистом в области физики твердого тела, Джоном Хопфилдом.

На создание данной модели Хопфилда подтолкнули его исследования в области спиновых стекол [49-52]. В кристаллической решетке атомы, обладающие магнитными моментами – спинами, могут различным образом

взаимодействовать друг с другом. Если взаимодействие между спинами таково, что оно стремится сориентировать их параллельно, то в основном состоянии (когда температура вещества  $T = 0^{\circ} K$ ) все атомы в решетке ориентируют свои спины параллельно. Такие вещества называются ферромагнетиками. Для других веществ – антиферромагнетиков, каждому положительному направлению спина атома соответствует атом с противоположным направлением спина. Однако существуют такие вещества, в которых характер взаимодействия между магнитными моментами приводит к тому, что в основном состоянии направления спинов хаотичны. Такие вещества носят название спиновых стекол. Для спиновых стекол такое поведение магнитных моментов означает вырождение основного состояния. Спиновое стекло может «замерзнуть» в любом из возможных состояний системы.

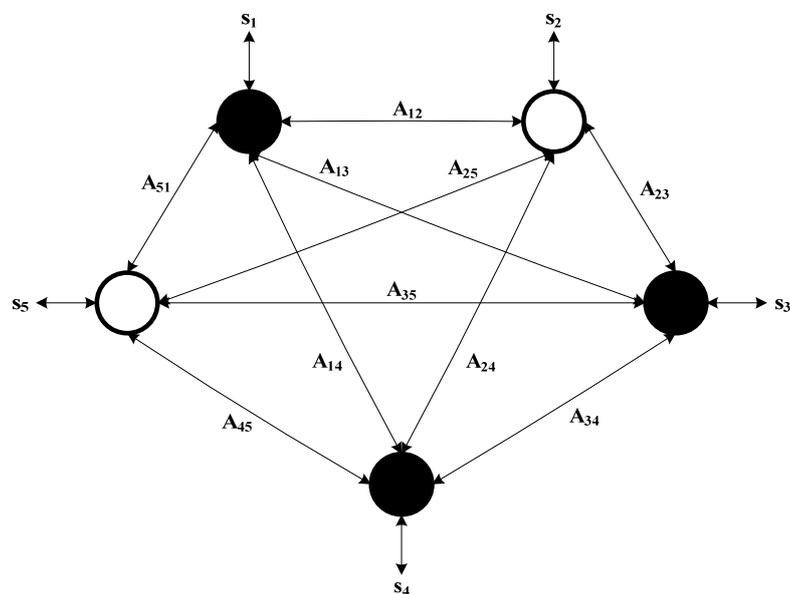
При исследовании механизмов распознавания Хопфилд предположил, что аналогичное явление может лежать в основе существования огромного числа состояний памяти, характерного для мозга. Например, можно рассмотреть модель полносвязной нейронной сети с рекуррентными симметричными связями между нейронами. В такой модели, возбуждающим связям будут соответствовать ферромагнитные связи в спиновом стекле, а тормозным – антиферромагнитные связи. Подобно спиновым стеклам, такие сети будут иметь множество стационарных конфигураций активностей нейронов, т.е. таких состояний, к которым сходится динамика сети. Именно введенная Хопфилдом динамика изменения состояния нейронов наряду с симметричностью связей между ними обусловила новизну описываемой модели.

Для модели Хопфилда была измерена емкость памяти, величина которой составила менее 15% от размерности сети  $N$ . Впервые было введено понятие энергии нейронной сети, которая в процессе работы нейронной сети монотонно понижается. Хопфилд показал, что эта энергия ограничена и в силу монотонности ее понижения процесс распознавания конечен, а

эталонные векторы, записываемые в нейронную сеть по правилу Хебба, являются локальными минимумами введенного им энергетического функционала.

В период 1985-1987 гг. Амит, Гудфренд и Шомполинский [53] на основе статфизических методов развивают подход, позволяющий получить аналитические оценки емкости памяти ( $M = N / \ln N$ ) и вероятности ошибки распознавания модели Хопфилда.

Исследования, начатые Гарднер и проводимые с 1965 по 1993 годы, показали, что *максимальная* емкость памяти модели Хопфилда [54-58] при оптимальном правиле обучения (не хеббовском) не превышает  $2N$ .



**Рис. 1.1.** Схема нейронной сети Хопфилда размерностью  $N = 5$ .

Модель Хопфилда принято описывать как систему взаимосвязанных нейронов (рис. 1.1). Состояние сети описывается  $N$ -мерным вектором  $\mathbf{S}$ , компоненты которого, бинарные переменные, принимают значения  $\pm 1$ . Связи между нейронами задаются с помощью симметричной матрицы связей  $\hat{\mathbf{A}}$  с нулевыми диагональными элементами. Каждый нейрон связан со всеми остальными.

Пусть в начальный момент времени сеть находится в состоянии  $\mathbf{S}_0$ . На каждый нейрон со стороны остальных нейронов действует локальное поле  $\mathbf{H}$

$$\mathbf{H} = -\mathbf{V} + \hat{\mathbf{A}} \cdot \mathbf{S}. \quad (1.1)$$

Под влиянием этого поля компоненты состояния  $\mathbf{S}(t)$  меняются по правилу:

$$s_i(t+1) = \begin{cases} s_i(t), & s_i(t)H_i(t) \geq 0 \\ -s_i(t), & s_i(t)H_i(t) < 0 \end{cases}, i=1, \dots, N. \quad (1.2)$$

Состояние нейрона не меняется, если ориентация этого нейрона и локального поля одинаковы, и меняется на противоположное в противном случае. Вектор  $\mathbf{B}$  задает пороги нейронов, т.е. границу, при превышении которой нейрон меняет свое состояние.

Состояние сети характеризуется функцией энергии  $E$

$$E = -\frac{1}{2}(\mathbf{S}, \hat{\mathbf{A}}\mathbf{S}) + (\mathbf{B}, \mathbf{S}), \quad (1.3)$$

В процессе эволюции сети энергия состояния неуклонно понижается. Постепенно система приходит в состояние покоя.

Правило обучения для сети Хопфилда опирается на исследования Дональда Хебба [45], который предположил, что синаптическая связь, соединяющая два нейрона, будет усиливаться, если в процессе обучения оба нейрона согласованно испытывают возбуждение либо торможение. Простой алгоритм, реализующий такой механизм обучения, получил название *правила Хебба*. Итак, требуется построить процесс получения матрицы связей  $\hat{\mathbf{A}}$ , такой, что соответствующая нейронная сеть будет иметь в качестве стационарных состояний образы обучающей выборки (значения порогов нейронов  $\mathbf{B}$  обычно полагаются равными нулю). Математически правило Хебба можно записать следующим образом

$$\hat{\mathbf{A}} = \sum_{\mu=1}^M \mathbf{S}_{\mu} \mathbf{S}_{\mu}^+. \quad (1.4)$$

Здесь  $M$  - число эталонных образов, записываемых в нейронную сеть;  $\mathbf{S}_{\mu}$  -  $\mu$ -ый эталон. Ряд исследований показывает, что нейронная сеть, обученная по правилу Хебба, может в среднем, при больших размерах сети  $N$ , хранить не более чем  $M = 0.138N$  различных образов.

Довольно часто используется обобщенное правило Хебба, т.е. каждый эталон записывается в сеть с некоторым весовым коэффициентом

$$\hat{\mathbf{A}} = \sum_{\mu=1}^M r_{\mu} \mathbf{S}_{\mu} \mathbf{S}_{\mu}^+, \text{ где } \sum_{\mu=1}^M r_{\mu}^2 = 1. \quad (1.5)$$

При использовании обобщенного правила становится возможным сделать некоторые эталоны глубже других (чем больше  $r_{\mu}$ , тем глубже минимум). В работах [59, 60] исследуются различные способы выбора весовых коэффициентов  $r_{\mu}$ , например, как числовой ряд или геометрическую прогрессию. При этом как коэффициенты не выбирай больше  $M = 0.138N$  не получить. Такой способ формирования матрицы межсвязей позволяет легко добавлять (или вычитать) в сеть новые эталоны. В работе [61] было показано, что любую матрицу можно разложить как сумму внешних произведений векторов (минимумов сети), т.е. представить в виде обобщенного правила Хебба.

К сожалению, у нейросети Хопфилда есть ряд существенных недостатков. У нее относительно небольшой объем памяти. Это означает, что попытка записать большее число образов приводит к тому, что сеть перестает их распознавать. Так же возникает множество ложных состояний. Причины их появления хорошо исследованы методами статистической физики в работе [62], где показано, что ложные состояния образуются как следствие интерференции минимумов. Число их экспоненциально увеличивается с ростом размерности сети. Из-за большого числа ложных минимумов с маленькой энергией сложнее находить более глубокие минимумы, так как в процессе работы нейросеть застревает в мелких состояниях. Таким образом, с ростом размерности задачи вероятность найти самые глубокие минимумы резко падает. Поэтому приходится делать большое число случайных стартов, что значительно замедляет алгоритм и при больших размерностях его время работы становится неприемлемо большим. Например, сеть Хопфилда размерности 100 имеет более 50 000 локальных минимумов, и нахождение субоптимального решения требует  $\sim 1$  часа машинного времени. В случае размерности сети 1000 вероятность нахождения наиболее глубокого минимума составляет  $\sim 10^{-6}$  и требует  $\sim 500$  часов. Кроме того, нейронные

сети Хопфилда не могут решить задачу распознавания, если изображение смещено или повернуто относительно его исходного запомненного состояния.

### §1.3. Исследования и применения нейронной сети Хопфилда

В работе [63] была определена зависимость вероятности нахождения минимума от его глубины

$$W = W_0 \exp \left[ -NE_0^2 \left( \frac{1}{E_m^2} - \frac{1}{E_\infty^2} \right) \right]. \quad (1.6)$$

Здесь  $E_m$  - энергия локального минимума,  $E_0$ ,  $E_\infty$  - верхняя и нижняя границы спектра локальных минимумов,  $W_0$  - не существенная медленная функция от  $E_m$ . На основе этой зависимости даны рекомендации по проведению процедуры оптимизации. Состоят они в том, чтобы использовать выражение (1.6) с неизвестными параметрами  $W_0$ ,  $E_0$  и  $E_\infty$ . Для этого запускается процедура случайного поиска и находится какое-то число минимумов. По полученным данным определяются характерные для данной матрицы значения  $E_0$ ,  $E_\infty$  и величина подгоночного параметра  $W_0$ . Подстановка этих величин в (1.6) позволит оценить вероятность нахождения неизвестного более глубокого минимума  $E_m$  (если он существует) и принять решение на остановку программы поиска (если оценка пессимистична) или ее продолжение.

В 1991-1999 г. Фролов и Муравьев различными методами исследуют нейронные сети хопфилдова типа с *редким кодированием* [64-70]. Анализируют информационную емкость, качество воспроизведения и размер области притяжения. Исследуют асимптотические свойства ассоциативной памяти. Показывают, что оптимальная разреженность близка к уровню нервной активности. Разработали алгоритм, позволяющий моделировать нейронные сети экстремально большой размерности (до  $1,5 \cdot 10^5$  нейронов).

Большинство исследований нейронных сетей направлены на увеличение их количественных характеристик таких, как емкость памяти, вероятность распознавания эталонов, помехоустойчивость и т.д. Тем не менее, распознавание образов не единственная сфера применения сети

Хопфилда. Динамическая процедура, описанная выше, на каждом шаге понижает значение энергии нейронной сети. Это позволяет решать комбинаторные задачи оптимизации, если они могут быть сформулированы как задачи минимизации квадратичного функционала (1.1).

Впервые о возможности использования нейросети для поиска оптимумов, заявил сам Хопфилд в его совместных работах с Танком [71, 72], в которых был разработан нейросетевой подход к решению задачи путешествующего коммивояжера (Traveling salesman problem - TSP) [73, 74]. Подход заключался в дополнении функции энергии нейросети набором дополнительных параметров, при правильном подборе которых, случайно инициализированная нейронная сеть, в ходе своей динамики всегда переходит в устойчивое состояние, которое может быть одним из решений. Смоделированная ими задача включала в себя 10 городов, которые коммивояжер должен был посетить за минимальное время. Результаты экспериментов показали высокий процент нахождения сетью качественных (коротких) маршрутов. Эта публикация вызвала большой интерес к сети Хопфилда как к инструменту оптимизации, поскольку она имеет ряд свойств необходимых для успешной оптимизации. Основными из них являются: параллелизм, легкая реализация в виде аппаратного устройства и общий подход к решению различных оптимизационных задач, таких, как многопараметрические задачи, задачи с большим уровнем помех и возможностью сбоя, неустойчивые, с большой размерностью (особенно NP-полные задачи), с множеством локальных экстремумов и негладкостью оптимизируемой функции. Однако при всех достоинствах существуют и недостатки. В основном это недостаточная скорость работы алгоритма и тенденция к сходимости в неглубокий локальный минимум (а таких минимумов в реальных задачах очень много). Кроме того, результаты данной работы оказались спорными, поскольку многие исследователи не смогли повторить их. Первыми из них были Уилсон и Паулей (Wilson and Pawley) [75], которые указали, что набор параметров, которыми дополнялась

функция энергии, заранее неизвестен, и подобрать их оптимальное соотношение очень сложно. Они предложили заменить эти параметры одним дополнительным выражением. В дальнейшем Бранд (Brandt) с соавторами [76] и Айер (Aiyer) с соавторами [77] показали, что свойства сходимости сети в устойчивые состояния могут быть улучшены путем дополнительной модификации функции энергии. Протцель (Protzel) и др. [78] подробно изучили свойства модели предложенной Брандом с отличным от нее набором параметров с целью нахождения глобального оптимума. Однако к существенным улучшениям эти модификации не привели. Вероятность остановки системы в неоптимальном локальном минимуме была все еще высока.

В работах [79-81] Крыжановским и Магомедовым была предложена модификация сети Хопфилда – доменная нейронная сеть, спины которой группируются в домены – группы по  $k$  связанных спинов в каждой. Показано что, доменная модель обладает в  $k$  раз большей емкостью памяти и в  $k^2$  раз большей скоростью работы по сравнению со стандартной моделью Хопфилда. Исследована возможность применения доменной сети в задачах комбинаторной оптимизации. Показано, что использование доменного алгоритма позволяет значительно ускорить решение таких задач и получать при существенно меньших вычислительных затратах более оптимальное решение, чем при использовании стандартных нейросетевых подходов.

В работе [82] Костенко В.А. и др. предложили решение задачи планирования вычислений для многопроцессорной системы с помощью сети Хопфилда. Взяв за основу подход Хопфилда и Танка, они построили функционал с дополнительными ограничениями на допустимые решения и свели его к энергии сети

$$E = -\frac{1}{2} \sum_{x=1}^N \sum_{y=1}^N \sum_{i=1}^M \sum_{\substack{j=1 \\ i \neq j}}^M w_{xij} s_{xi} s_{yj} + \sum_{x=1}^N \sum_{i=1}^M B_{xi} s_{xi} . \quad (1.7)$$

Здесь  $N$  - количество процессов не связанных между собой,  $M$  - количество процессоров,  $t_x$  - время выполнения одного процесса,  $\alpha, \beta, \gamma$  - некоторые

константы,  $\delta_{ij}$  - символ Кронекера. Состояние нейронной сети задается матрицей  $\mathbf{S}$  размерности  $N \times M$ , а ее элементы принимают значения -1 или +1. Весовые коэффициенты и пороги нейронов равны, соответственно,

$$w_{xij} = -\frac{1}{4}(\alpha\delta_{ij}t_x t_y + \gamma + \beta\delta_{xy}),$$

$$B_{xi} = \alpha/2 T_{cp} (M/2 - 1)t_x + \beta/4 (M - 1) + \gamma/2 N(M/2 - 1). \quad (1.8)$$

Коэффициенты были выбраны следующие

$$\alpha = 1, \beta = \left( \frac{1}{N} \sum_{x=1}^N t_x t_x \right)^2, \gamma = 0. \quad (1.9)$$

В своей работе авторы исследуют влияние числа процессов ( $N = 30 \div 90$ ), числа процессоров ( $M = 6; 8; 10$ ) и начального приближения (случайное и эвристическое) на качество решения сетью поставленной задачи. Так было определено, что если начальное приближение выбрано некоторой эвристикой, а не случайным образом, то процент корректных решений возрастает с 33.8% до 55.6%; процент загрузки снижается в среднем с 98.8% до 95.5%; а количество итераций – с  $2.8NM$  до  $1.2NM$ . Число корректных решений с ростом числа процессоров  $M$  в среднем возрастает, а с ростом числа процессов  $N$  - уменьшается.

В работе [83] Тарковым решалась задача планирования вычислений для многопроцессорной системы в несколько ином виде. В данной работе рассматривалась задача вложения графов параллельных программ (т.е. имеются связи между процессами) в графы распределенных вычислительных систем, когда совпадает число вершин графа программы с числом машин в системе. Качество вложения оценивалось числом совпадений ребер графа программы с ребрами графа вычислительной системы. В своей работе автор предлагает рассмотреть матрицу нейронов  $\mathbf{S}$  размером  $N \times N$ , каждая строка матрицы соответствует ветви параллельной программы, каждый столбец матрицы соответствует элементарной машине. Каждая строка и каждый столбец матрицы  $\mathbf{S}$  должны содержать один и только один ненулевой

элемент, равный единице, остальные элементы должны быть равны нулю. Энергия соответствующая нейронной сети Хопфилда описывается следующей функцией Ляпунова

$$E = \frac{C}{2} \left[ \sum_x \left( \sum_j s_{xj} - 1 \right)^2 + \sum_i \left( \sum_y s_{yi} - 1 \right)^2 \right] + \frac{D}{2} \sum_x \sum_i \sum_{y \in Nb_p(x)} \sum_{j \neq i} s_{xi} s_{yj} t_{ij}. \quad (1.10)$$

Здесь  $s_{xi}$  - состояние нейрона строки  $x$  и столбца  $i$  матрицы  $S$ ,  $C$  и  $D$  - параметры функции Ляпунова, взаимосвязь которых исследована в [84],  $t_{ij}$  - расстояние между вершинами графа системы, содержащими соседние ветви параллельной программы,  $Nb_p(x)$  - множество вершин, смежных с вершиной  $x$  на графе программы. В квадратных скобках минимум первого слагаемого соответствует наличию одного единичного элемента в каждой строке матрицы  $S$ , а минимум второго означает наличие одного единичного элемента в каждом столбце этой матрицы. Минимум суммы первых двух слагаемых обеспечивается в случае, когда каждая строка и каждый столбец матрицы содержат один и только один элемент, равный 1, остальные элементы – нули (корректное решение). Минимум третьего слагаемого обеспечивает минимум суммы расстояний между соседними ветвями параллельной программы при их размещении на графе системы.

Динамика сети Хопфилда, минимизирующей функцию  $E$ , описывается системой уравнений

$$\frac{\partial u_{xi}}{\partial t} = - \frac{\partial E}{\partial s_{xi}}, \quad (1.11)$$

где  $u_{xi}$  - активация нейрона с индексами  $x, i$ , а состояние нейрона в процессе минимизации задается следующим образом

$$s_{xi} = \frac{1}{1 + \exp(-\beta u_{xi})}. \quad (1.12)$$

После того как сеть приходит в устойчивое состояние, элементы матрицы  $S$  обрезаются до значений 0 и 1.

Эксперименты проводились для числа вершин 9 и 16. Подбирая свободные параметры и применяя небольшие модификации функционала, автор добивается уменьшения числа некорректных решений и увеличения вероятности нахождения оптимального решения до  $80 \div 90 \%$ .

Как и в прочих аналогичных работах, получить приемлемое решение удается только при малых размерностях, так как алгоритмы являются критичными к объему оперативной памяти и времени работы.

Однако у этих подходов есть несколько недостатков. Во-первых, при составлении минимизационного функционала используются штрафные функции, что в свою очередь приводит к появлению коэффициентов, которые сильно влияют на качество работы сети. Подбор этих коэффициентов является сам по себе трудоемкой задачей, до сих пор не решенной. Во-вторых, у сети есть так называемые «некорректные» решения (например, сеть может сойтись к состоянию, в котором одна и та же задача подается на два процессора одновременно, что противоречит постановке задачи). Количество «некорректных» решений зависит от корректирующих коэффициентов, упомянутых выше и размерности сети.

В работах [85-91] решалась та же задача многопроцессорного планирования. Но предложенные там нейросетевые алгоритмы лишены описанных выше недостатков.

Отнесем к  $T_j$  как к случайной величине. Очевидно, что для любого распределения задач по процессорам среднее значение  $\{T_j\}_1^M$ , принимает одно и то же значение:  $\bar{T} = \sum_{j=1}^M T_j / M$ . Понятно, что чем ближе распределение

задач к оптимальному, тем дисперсия распределения  $\{T_j\}_1^M$  меньше:

$$D \sim \sum_{j=1}^M (T_j - \bar{T})^2.$$

Рассмотрим для начала систему из двух процессоров ( $M = 2$ ). В этом случае бинарный вектор  $\mathbf{S} = (s_1, s_2, \dots, s_N)$  содержит информацию о том,

какому процессору назначена задача. Если  $s_i = 1$ , то  $i$ -ая задача назначена на 1-ый процессор, если  $s_i = -1$ , то задача назначена на 2-ой процессор.

Минимальное время работы системы достигается, когда процессоры загружены, по возможности, одинаково. В этом случае дисперсия времен работы процессоров минимальна. Поэтому будем минимизировать дисперсию, которую можно записать в виде

$$D = (T_1 - T_2)^2 = (\mathbf{V} \cdot \mathbf{S})^2, \quad (1.13)$$

где  $\mathbf{V} = (V_1, V_2, \dots, V_N)$  - вектор вычислительных сложностей задач.

Локальные поля определяются выражением

$$H_i = W - V_i s_i, \quad (1.14)$$

где  $W = \mathbf{V} \cdot \mathbf{S}$ . При каждом изменении состояния нейрона величина  $W$  обновляется по схеме

$$W(t+1) = W(t) + 2V_i s_i(t+1). \quad (1.15)$$

Среднее число операций алгоритма не превышает  $Op \approx 7N$ .

Анализ результатов показывает, что с увеличением нагрузки процессора погрешность решения уменьшается и при  $N/M > 20$  становится пренебрежимо малой. При уменьшении нагрузки погрешность решения возрастает. Тем не менее, даже в этой области она меньше 1%. Вероятность отыскания оптимального решения увеличивается с ростом числа задач в наборе. Вероятность отыскания глобального минимума становится больше 0.5 при загрузке больше 50. В процессе минимизации нейроны изменяют свое состояние не более  $0.2N$  раз. Число операций алгоритма пропорционально числу задач в наборе и составляет менее  $Op \approx 4.6N$ . Интересен факт, что с увеличением нагрузки алгоритм дает лучший результат. Это связано с тем, что при большем числе задач в потоке появляется больше вариантов оптимальных или близких к оптимальному распределений, которые сеть успешно находит.

Обратимся теперь к распределению задач в многопроцессорной системе:  $M > 2$ . В основе алгоритма балансировки многопроцессорной

системы лежит описанный выше алгоритм оптимизации двухпроцессорной системы. Алгоритм балансировки заключается в следующем.

1. Задачи случайным образом распределяются по всем процессорам.
2. Выбирается два процессора: один с максимальным временем работы, другой с минимальным временем.
3. Двухпроцессорной моделью оптимизируется распределение на этой паре процессоров.
4. Шаги 2 и 3 повторяются до тех пор, пока происходят изменения в общем распределении.

Средняя величина погрешности решения увеличивается. Вероятность отыскания оптимального решения при этом уменьшается. Тем не менее, с увеличением нагрузки на процессор погрешность решения быстро уменьшается и уже, например, для  $M = 16$  и  $N/M > 10$  составляет менее 1%. В среднем количество выбираемых пар процессоров для балансировки составляет  $\approx 2.5M$ , а число операций алгоритма составляет  $Op \approx 23N$ .

В работах [92] Крыжановским и Карандашевым предлагается возводить матрицу в некоторую степень с целью уменьшения числа мелких локальных минимумов и увеличения тем самым вероятности нахождения глобального минимума. Авторами предлагается следующий алгоритм. Подготовительная фаза: исходная матрица  $T$  симметризуется (если она изначально не симметрична) и диагональные матричные элементы обнуляются; матрица возводится в  $k$ -ю степень и в полученной матрице  $M = T^k$  обнуляются диагональные элементы; на основе матрицы  $M$  в соответствии с (8) строится функционал  $E_k(\mathbf{S})$ . После подготовки можно приступить к процедуре случайного поиска на основе алгоритма двухэтапного спуска, состоящего в следующем: на первом этапе из случайной начальной конфигурации осуществляется спуск по поверхности  $E_k(\mathbf{S})$  в ближайший локальный минимум  $\mathbf{S}_m^k$  функционала  $E_k(\mathbf{S})$ ; на втором этапе проводится коррекция – из точки  $\mathbf{S}_m^k$  по поверхности  $E_1(\mathbf{S})$  спускаемся

в ближайший локальный минимум  $S_m$  функционала  $E_1(\mathbf{S})$ , который, как правило, находится недалеко от точки  $S_m^k$ . Исследования показали, что трансформация поверхности приводит к существенному повышению эффективности оптимизационного алгоритма. В частности, при использовании спуска по поверхности  $E_3(\mathbf{S})$  имеем: вероятность попадания в глобальный минимум увеличилась более чем в  $1.5 \times 10^4$  раз и достигла вполне разумной величины  $\sim 0.04$ ; разница между средней энергией и энергией глобального минимума сократилась на порядок; вероятность попадания в близкий к глобальному минимуму интервал энергий  $E_m \in [-1; -0.99]$  возросла более чем в 5000 раз.

Вопрос выбора стартовых состояний исследовался отдельно. В работах [93-98] рассматривалась идея инициализации нейронной сети конфигурационными векторами, близкими к реальным собственным векторам матрицы связей. В этих работах предпринята попытка найти оптимальное начальное состояние оптимизируемой системы. Т.е. вместо случайного поиска, для матрицы связей находились ее собственные значения и собственные вектора, затем по полученным собственным векторам строились конфигурационные вектора (элементы которых принимают значения  $\pm 1$ ), с которых и запускалась работа сети. В результате выяснилось, что нейронная сеть чаще конвергирует в глубокий минимум, если инициализировать ее собственными векторами соответствующим первым 10 – 30 максимальным собственным значениям матрицы связей. Большая серия экспериментов, доказала, что использование этого алгоритма несколько чаще доставляет более глубокий минимум по сравнению со стандартным слепым поиском, однако при этом затрачивается дополнительное время на определение собственных векторов и значений матрицы связей. Так же подбора стартовых состояний касаются в работах [82, 91].

Для нахождения более глубоких минимумов при решении задач оптимизации был разработан целый ряд алгоритмов, которые позволяют системе выходить из локальных минимумов.

*Simulated annealing.* Одним из таких алгоритмов является алгоритм *имитации отжига*. Данный алгоритм был впервые предложен Киркпатриком [99] в 1983 году и хорошо описан в [100]. В основу алгоритма легли процессы, происходящие в металле при нагревании и последующем медленном охлаждении. При высокой температуре молекулы в металле, имея большую степень свободы, хаотически перемещаются. Однако, при медленном снижении температуры они выстраиваются в устойчивую кристаллическую решетку. Такое состояние металла оптимально, поскольку силы взаимодействия элементарных частиц максимально уравновешены и система стабильна.

Для имитации такого процесса в модель нейронной сети Хопфилда вводится понятие убывающей со временем «температуры»

$$T(\tau) = \frac{T(0)}{\ln(1 + \tau)},$$

где  $\tau$  - время работы алгоритма.

Алгоритм имитации отжига состоит в следующем:

1. Нейронная сеть инициализируется в некоторое случайное начальное состояние -  $\mathbf{s}_0$
2. В каждый момент времени  $\tau$  выбирается один спин и анализируется его состояние в локальном поле. Если спин неустойчив, т.е. его переворот приведет систему к состоянию с меньшим значением энергии, то такой спин переворачивается с вероятностью

$$P(\mathbf{s}_\tau \rightarrow \mathbf{s}_{\tau+1}) = \frac{1}{1 + e^{-\Delta E/T}}.$$

Если же спин устойчив и его переворот ухудшит состояние системы и повысит значение энергии, то такой спин будет перевернут с вероятностью

$$P(\mathbf{s}_\tau \rightarrow \mathbf{s}_{\tau+1}) = \frac{e^{-\Delta E/T}}{1 + e^{-\Delta E/T}},$$

где  $\Delta E = E(\tau) - E(\tau + 1)$ ,  $T = T(\tau)$ .

3. Шаг №2 повторяется до тех пор, пока система не остановится в некоем минимуме.

Температура в начале работы алгоритма полагается большой, однако со временем ее значение уменьшается и стремится к нулю, по аналогии с температурой остывающего металла. Таким образом, в начале своей работы алгоритм с большей вероятностью будет принимать неоптимальные решения, но когда температура спадет, вероятность принятия таких решений уменьшится. Шаги, повышающие значение энергии, иногда необходимы, когда нужно вывести систему из локального минимума. Но принятие многих негативных шагов может увести в сторону от глобального оптимума. В настоящее время данный метод активно исследуется [101-103], и успешно применяется во множестве областей, например при проектировании СБИС [100].

Tabu search. Еще один хорошо зарекомендовавший себя алгоритм называется алгоритмом *поиска с запретами*. Основоположником алгоритма является Ф. Гловер [104, 105], который предложил принципиально новую схему локального поиска. После получения решения случайным поиском, произвольным образом выбирается один нейрон, из числа тех чье энергетическое состояние стабильно, и изменяется его состояние. Это выводит сеть из стабильного состояния и немного повышает значение энергии, однако позволяет сети выйти из локального минимума. Помимо этого, чтобы не позволить сети сойтись в прежний минимум номер выбранного нейрона заносится в специальный список «табу» (tabu queue), и пока он находится в этом списке нейронной сети запрещается изменять его состояние. Существует много вариантов реализации основной идеи поиска с запретами [106], которые наряду с базовым вариантом активно применяются к задачам комбинаторной оптимизации [107, 108].

Local minima escape. На практике также активно применяется предложенный в работе [109], алгоритм проведения бегства из локальных минимумов. LME является модификацией случайного поиска, с той разницей, что каждое начальное состояние сети, задается вектором который не приведет систему в прежний минимум, т.е. лежит вне области его притяжения. Этот, алгоритм развит в [110], и нацелен на поиск глобального минимума. Однако при этом, для реализации алгоритма, в памяти ЭВМ должно храниться две копии нейронной сети, и затрачиваться значительное время на вычисление нового состояния.

При работе с сетью Хопфилда приходится хранить матрицу весовых коэффициентов, что налагает большие требования на оперативную память. А необходимость проводить множество случайных пусков для нахождения удовлетворительных решений занимает много вычислительного времени.

Больших успехов в экономии памяти и увеличении быстродействия алгоритмов удалось добиться с помощью клиппирования (или бинаризации) элементов матрицы связей. Суть клиппирования заключается в том, что элементы матрицы связей «обрезаются» по знаку, т.е. положительные\отрицательные числа заменяются  $+1\backslash-1$ , а нули остаются без изменений. Изначально клиппирование применялось в задаче распознавания образов.

Впервые нейронную модель с бинарными коэффициентами в 1961 году предложил Стейнбах [111]. Сейчас эта модель больше известна как модель Вилшоу [112]. Исследования, проведенные Пальмом, показали [113], что при *редком кодировании* эта модель обладает рекордной информационной емкостью.

Бинаризацию синапсов для нейросетевой модели ADALINE в 1960 году также исследовали Уидроу и Хофф [114], которые показали, что клиппирование связей привело к увеличению *емкости памяти*.

В 80-е годы, в работах [115-117] вновь была описано клиппирование связей, только уже для модели Хопфилда. Методами статистической физики,

были получены аналитические оценки *емкости памяти* и восстанавливающей способности сети. Основной недостаток модели Хопфилда - малая *емкость памяти*. Поскольку на тот момент еще не существовало способов повышения емкости памяти, дальнейшие исследования в области бинаризации синапсов модели Хопфилда казались нецелесообразными.

Эти исследования были продолжены в [118, 119] для сети Хопфилда, и было показано, что уменьшение клиппированного функционала при переходе из одного состояния в другое сопровождается уменьшением исходного функционала. Удалось на порядок повысить быстродействие алгоритма минимизации и снизить требования к оперативной памяти. На основе этого в [120] было предложено использование процедуры клиппирования при решении задач оптимизации.

Тем не менее, в среднем эффективность минимизации на 80% хуже, чем при использовании обычной модели Хопфилда. Минимумы функционала, построенного на клиппированной матрице, не являются минимумами исходного функционала. Поэтому приходится проводить коррекционные спуски на исходной сети, что сильно снижает преимущества даваемые клиппированием.

В настоящей работе исследуется обобщенная процедура бинаризации, а также возможности ее улучшения.

#### ***§1.4. Цели и задачи диссертационной работы***

Основная цель диссертационной работы состояла в разработке быстрого метода решения задач комбинаторной оптимизации на базе нейронных сетей.

В диссертационной работе были решены следующие задачи:

1. Предложена и исследована процедура дискретизации матричных элементов, позволяющая ускорить процесс минимизации многоэкстремального квадратичного функционала, построенного в пространстве состояний с бинарными переменными.
2. Теоретически и экспериментально получены зависимости вероятности несовпадения градиентов и эффективности минимизации.
3. На основе процедуры дискретизации разработаны и исследованы алгоритмы минимизации квадратичного функционала. Исследовано увеличение быстродействия новых алгоритмов и его эффективность по сравнению со стандартной моделью Хопфилда.

### ***§1.5. Основные положения, выносимые на защиту***

1. Процедура дискретизации матричных элементов, позволяющая ускорить процесс минимизации многоэкстремального квадратичного функционала, построенного в пространстве состояний с бинарными переменными.
2. Алгоритмы поиска минимумов квадратичного функционала в пространстве бинарных переменных с использованием процедуры дискретизации.

### ***§1.6. Методы исследования***

Для решения поставленных задач в работе были использованы методы вычислительной математики, теории вероятностей и математической статистики, а также методы прикладного программирования.

### ***§1.7. Научная новизна***

В диссертационной работе

1. Предложена и исследована процедура дискретизации
2. Создан алгоритм, позволяющий увеличить скорость решения оптимизационных задач и получать при существенно меньших вычислительных затратах более лучшее решение, чем при использовании стандартного модели Хопфилда.
3. Применение дискретизации снижает требования к оперативной памяти, что позволяет решать задачи недоступные стандартной модели Хопфилда.
4. Получены выражения, позволяющие аналитически оценить качество решения при использовании минимизационного алгоритма.

### ***§1.8. Практическая ценность***

Практическая ценность результатов работы состоит в следующем:

1. На основе процедуры дискретизации создан алгоритм, позволяющий уменьшить время решения оптимизационных задач и получать при существенно меньших вычислительных затратах более лучшее решение, чем при использовании стандартного подхода, основанного на модели Хопфилда;
2. Получены выражения, позволяющие априори аналитически оценить качество решения, которое может быть получено в результате минимизации.

Разработанные алгоритмы и методы дискретизации найдут свое применение в прикладных системах на основе нейронных сетей Хопфилда.

### ***§1.9. Апробация работы и публикации***

По материалам диссертации опубликовано 11 работ, из них 7 - в российских и зарубежных журналах, 4 - в трудах конференций.

1. Kryzhanovsky M.V., Malsagov M.U. **Clipping procedure in optimization problems and its generalization** // Optical Memory and Neural Networks (Information Optics).– 2009.– Vol. 18, №3.– pp. 181-187.
2. Крыжановский Б.В., Крыжановский М.В., Мальсагов М.Ю. **Дискретизация матрицы в задаче бинарной минимизации квадратичного функционала** // Доклады Академии Наук.– 2011.– Т.438, №3.– сс. 312-317.
3. Мальсагов М.Ю. **Понижение разрядности элементов матрицы для ускорения алгоритма дискретной оптимизации** // Нейрокомпьютеры: разработка и применение.– 2011.– №4.– сс. 22-31.
4. Kryzhanovsky M.V., Malsagov M.Yu. **Modification of Binary Optimization Algorithm and Use Small Digit Capacity Numbers** // Optical Memory and Neural Networks (Information Optics).– 2011.– Vol. 20, №3.– pp. 194–200.
5. Крыжановский М.В., Мальсагов М.Ю. **Применение чисел малой разрядности в задаче бинарной оптимизации** // Программные продукты и системы.– 2011.– №4.– сс. 40-44.
6. Крыжановский М.В., Мальсагов М.Ю. **Обобщение процедуры клиппирования в задачах оптимизации в дискретном пространстве** // Искусственный интеллект.– 2009.– №3.– сс. 496-503.
7. Крыжановский М.В., Мальсагов М.Ю. **Особенности применения дискретизации в задачах поиска глобального минимума** // Искусственный интеллект.– 2011.– №3.– сс. 497-505.
8. Крыжановский М.В., Мальсагов М.Ю. **Применение процедуры клиппирования и ее обобщение в задачах поиска глобального минимума** // Сборник трудов XI Всероссийской научно-технической

конференции «Нейроинформатика-2009».– М.: МИФИ, 2009.– ч.2.– сс. 61-68.

9. Крыжановский М.В., Мальсагов М.Ю. **Ускорение модифицированной процедуры клиппирования** // Сборник трудов XII Всероссийской научно-технической конференции «Нейроинформатика-2010».– М.: МИФИ, 2010.– ч.2.– сс. 45-54.
10. Kryzhanovsky M.V., Malsagov M.U. **Small digit capacity arithmetic for problems of discrete optimization** // Proc. of the IV International Conference on Neural Networks and Artificial Intelligence «ICNNAI-2010».– Brest.– Belarus.– pp. 101-106.
11. Крыжановский М.В., Мальсагов М.Ю. **Дискретизация матрицы в задаче бинарной оптимизации** // Сборник трудов XIII Всероссийской научно-технической конференции «Нейроинформатика-2011».– М.: МИФИ, 2011.– ч.3.– сс. 152-160.

Основные положения работы докладывались на следующих конференциях:

1. XI Всероссийская научно-техническая конференция «Нейроинформатика-2009», Москва, 2009.
2. Международная Научно-Техническая конференция «Искусственный Интеллект. Интеллектуальные Системы – 2009», ИИ – 2009, Украина, пос. Кацевели, 2009.
3. XII Всероссийская научно-техническая конференция «Нейроинформатика-2010», Москва, 2010.
4. The fourth International Conference on Neural Networks and Artificial Intelligence, ICNNAI – 2010, респ. Белоруссия, Брест, 2010.
5. XIII Всероссийская научно-техническая конференция «Нейроинформатика-2011», Москва, 2011.
6. Международная Научно-Техническая конференция «Искусственный Интеллект. Интеллектуальные Системы – 2011», ИИ – 2011, Украина, пос. Кацевели, 2011.

### ***§1.10. Структура и объем диссертации***

Работа состоит из четырех глав, заключения, приложения и списка литературы.

Первая глава представляет собой введение, в котором дается описание проблемы и приводится обзор существующих методов ее решения. Сформулированы цели диссертационной работы, обоснована ее актуальность и научная новизна.

Во второй главе дается описание предлагаемого алгоритма дискретизации. Описывается процедура дискретизации, определяются оптимальные параметры алгоритма. Приводится аналитический анализ предложенного алгоритма: вероятность совпадения градиентов, эффективность минимизации. Результаты теоретических исследований подтверждаются вычислительными экспериментами.

Третья глава посвящена проблеме применения процедуры дискретизации. Построение минимизационного алгоритма (одно- или двухэтапного). Рассматриваются распределения полученных минимумов, скорость алгоритмов, вероятность нахождения самых глубоких минимумов. Показаны преимущества использования алгоритма на основе дискретизации.

В четвертой главе приводятся схемы алгоритмов, листинги основных процедур программ реализующих алгоритмы минимизации. Проводится анализ вычислительной сложности алгоритмов.

В заключении излагаются основные результаты диссертационной работы.

Общий объем диссертации составляет 133 страницы. Список литературы насчитывает 121 наименование.

## ГЛАВА 2. ПРОЦЕДУРА ДИСКРЕТИЗАЦИИ

Модель Хопфилда принято описывать как систему взаимосвязанных нейронов. Состояние сети описывается  $N$  – мерным вектором  $\mathbf{S}$ , компоненты которого, бинарные переменные, принимают значения  $\pm 1$ . Связи между нейронами задаются с помощью симметричной матрицы связей  $\hat{\mathbf{A}}$  с нулевыми диагональными элементами. Каждый нейрон связан со всеми остальными.

Пусть в начальный момент времени сеть находится в состоянии  $\mathbf{S}_0$ . На каждый нейрон со стороны остальных нейронов действует локальное поле  $\mathbf{H}$

$$\mathbf{H} = -\mathbf{V} + \hat{\mathbf{A}} \cdot \mathbf{S}. \quad (2.1)$$

Под влиянием этого поля компоненты состояния  $\mathbf{S}(t)$  меняются по правилу:

$$s_i(t+1) = \begin{cases} s_i(t), & s_i(t)H_i(t) \geq 0 \\ -s_i(t), & s_i(t)H_i(t) < 0 \end{cases}, \quad i = 1, \dots, N. \quad (2.2)$$

Состояние нейрона не меняется, если ориентация этого нейрона и локального поля одинаковы, и меняется на противоположное в противном случае. Вектор  $\mathbf{V}$  задает пороги нейронов, т.е. границу, при превышении которой нейрон меняет свое состояние.

Состояние сети характеризуется функцией энергии  $E$

$$E = -\frac{1}{2}(\mathbf{S}, \hat{\mathbf{A}}\mathbf{S}) + (\mathbf{V}, \mathbf{S}). \quad (2.3)$$

В процессе эволюции сети энергия состояния неуклонно понижается. Постепенно система приходит в состояние покоя.

Свойство нейронной сети находить минимумы квадратичного функционала часто используется для решения многих задач дискретной оптимизации. В большинстве таких задач элементами матрицы связей  $\hat{\mathbf{A}}$  являются вещественные числа. Это обстоятельство накладывает ограничения на размерность решаемых задач, т.к. для хранения большой матрицы вещественных чисел требуется большой объем памяти. Так же операции с такими числами являются довольно медленными, что обуславливает большое

время работы алгоритмов. Чтобы снизить требования к оперативной памяти и скорости алгоритмов для задач распознавания образов, еще в 60-х годах было предложено заменять матричные элементы их знаками, т.е.  $\pm 1$ . Нули остаются без изменений. Обзор работ по бинаризации приведен во введении. Сравнение приведенных в данной главе результатов исследований будет проводиться с результатами работы [121], в которой подробно исследовалась процедура бинаризации.

В данной главе предлагается подход, основанный на бинаризации. Его отличие заключается в том, что элементы исходно матрицы связей будут заменяться целыми числами. Еще одно отличие в том, что элементы близкие к нулю заменяются нулем. Будет показано, что введение такой области обнуления позволит существенно улучшить минимизационные качества алгоритма.

### ***§2.1. Процедура дискретизации***

Как можно понять из описания модели Хопфилда, основные требования на оперативную память налагает матрица связей. Вычислительная сложность алгоритма в основном связана с вычислением локального поля, что в свою очередь опять же упирается в действия с матрицей связей. Таким образом, очевидно, что именно модификация матрицы каким-либо способом позволит, хотя бы частично, решить проблемы с памятью и ускорением. Как упоминалось ранее, бинаризация позволила решить эти проблемы за счет эффективности алгоритма. В данной работе предлагается «расширение» процедуры клиппирования.

Рассмотрим случайную матрицу  $\hat{A}$ , элементы которой подчинены некоторому симметричному распределению  $f(A_{ij})$ . Дискретизация заключается в замене матрицы  $\hat{A}$  некоторой матрицей  $\hat{C}$ . Элементами матрицы  $\hat{C}$  являются целые числа в диапазоне  $[-q; q]$ , где  $q$  - число

градаций, свободный параметр, задаваемый пользователем. Матрица  $\hat{C}$  формируется следующим образом:

1. Вычитаем из каждого элемента матрицы  $\hat{A}$  среднее значение ее элементов  $A_0$  и получаем новую матрицу

$$A'_{ij} = A_{ij} - A_0. \quad (2.4)$$

2. Разбиваем область распределения элементов центрированного остатка  $\hat{A}'$  на  $(2q + 1)$  отрезка.

3. Формируем матрицу  $\hat{C}$  по правилу

$$C_{ij} = k \cdot \text{sign}(A'_{ij}), \text{ когда } x_{k-1} < |A'_{ij}| \leq x_k, \quad k = 0, 1, \dots, q. \quad (2.5)$$

Здесь  $x_k$  - правая граница  $k$ -ого отрезка.

Длины отрезков выбираются так, чтобы средние значения на отрезках были кратны величине  $m$ , где  $m > 0$  - наименьшее положительное среднее значение (среднее на  $k$ -ом отрезке равно  $km$ ,  $k = 0, \pm 1, \dots, \pm q$ ). Т.е. чтобы определить координаты отрезков необходимо решить систему уравнений относительно  $x_k$

$$\int_{x_{k-1}}^{x_k} y \cdot f(y) dy = km. \quad (2.6)$$

Заметим, что длины всех отрезков однозначно определяются длиной  $l_0$  отрезка с нулевым средним, которая является свободным параметром для последующей оптимизации. Более того, введение отрезка с нулевым средним является ключевым моментом, позволяющим наилучшим образом представить матрицу  $\hat{A}$  матрицей  $\hat{C}$  и существенно повысить эффективность алгоритма минимизации.

*Адаптация алгоритма минимизации.* Адаптируем алгоритм минимизации к работе с матрицей  $\hat{C}$ . Для этого сделаем в (2.2) замены  $A_{ij} \rightarrow C_0 + C_{ij}$  и  $B_i \rightarrow b_i$ . Тогда, расчет локального поля и обновление компонент конфигурационного вектора будет производиться в соответствии с выражениями:

$$h_i = -b_i + \sum_{j \neq i}^N (C_0 + C_{ij}) s_j \quad (2.7)$$

и

$$s_i(t+1) = \begin{cases} s_i(t), & s_i(t)h_i(t) \geq 0 \\ -s_i(t), & s_i(t)h_i(t) < 0 \end{cases}, \quad (2.8)$$

где величина  $C_0$  и компоненты вектора  $\mathbf{b} = (b_1, b_2, \dots, b_N)$  будут определены ниже из условия оптимизации минимизационного процесса. Заметим, что решающее правило (2.8) соответствует спуску по поверхности, описываемой «дискретизированным» функционалом

$$\varepsilon = -\frac{1}{2} (\mathbf{s}, (C_0 + \hat{\mathbf{C}})\mathbf{s}) + (\mathbf{b}, \mathbf{s}), \quad (2.9)$$

являющимся функцией Ляпунова для алгоритма минимизации.

Заменить правило обновления  $s_i = \text{sign}(H_i)$  его дискретизированным аналогом  $s_i = \text{sign}(h_i)$  можно только в том случае, когда знаки локальных полей  $H_i$  и  $h_i$  совпадают в любой точке  $N$ -мерного пространства с достаточно большей вероятностью (амплитуды полей  $H_i$  и  $h_i$  не играют роли). Иными словами, использовать модифицированное правило градиентного спуска (2.7) и (2.8) для минимизации функционала (2.3) можно, если свести до минимума величину ошибки – вероятность несовпадения направлений локальных полей в случайной точке пространства:

$$P = \text{Pr}\{H_i > 0 \cap h_i < 0\} + \text{Pr}\{H_i < 0 \cap h_i > 0\}, \quad (2.10)$$

Для минимизации ошибки надо найти наилучшее представление матрицы  $\hat{\mathbf{A}}$  матрицей  $\hat{\mathbf{C}}$  и найти оптимальные значения величин  $C_0$  и  $b_i$ ,  $i = \overline{1, N}$ .

*Оптимизация процедуры дискретизации.* Чтобы оптимизировать процедуру дискретизации необходимо минимизировать вероятность несовпадения направлений локальных полей. Сделать это можно, подобрав оптимальные значения величин  $C_0$  и  $b_i$ ,  $i = \overline{1, N}$ , а так же оптимизировать по длине отрезка с нулевым средним, найдя  $l_0$ .

Рассмотрим величину ошибки  $P$  в случайной точке пространства. В соответствии с центральной предельной теоремой, в случае  $N \gg 1$  величины  $H_i$  и  $h_i$  можно рассматривать как случайные (не независимые) переменные, подчиняющиеся двумерному нормальному распределению. Распределение задается средними значениями  $\langle H \rangle$  и  $\langle h \rangle$  величин  $H_i$  и  $h_i$ , соответственно, их стандартными отклонениями  $\sigma_H$  и  $\sigma_h$ , коэффициентом корреляции  $\rho$ :

$$\begin{aligned}\langle H \rangle &= -B_i, \quad \sigma_H^2 = N(A_0^2 + \sigma_A^2), \\ \langle h \rangle &= -b_i, \quad \sigma_h^2 = N(C_0^2 + \sigma_C^2), \\ \rho &= \frac{N[C_0 A_0 + \langle C_{ij} A'_{ij} \rangle]}{\sigma_H \sigma_h},\end{aligned}\tag{2.11}$$

где  $\sigma_A$  и  $\sigma_C$  - стандарты матричных элементов  $A_{ij}$  и  $C_{ij}$ , соответственно, а уголки скобки означают усреднение по ансамблю.

С учетом (2.11) выражение (2.10) преобразуется к виду:

$$P = 1 - \frac{1}{2\pi\sigma_H\sigma_h\sqrt{1-\rho^2}} \int_0^\infty \int_0^\infty (f_+ + f_-) dH dh,\tag{2.12}$$

где

$$f_\pm = \exp\left\{-\frac{1}{2(1-\rho^2)}\left[\frac{(H \pm \langle H \rangle)^2}{\sigma_H^2} - 2\rho\frac{(H \pm \langle H \rangle)(h \pm \langle h \rangle)}{\sigma_H\sigma_h} + \frac{(h \pm \langle h \rangle)^2}{\sigma_h^2}\right]\right\}.\tag{2.13}$$

Теперь определимся с выбором величин  $C_0$  и  $b_i$  ( $i = \overline{1..N}$ ). Зададим их минимизируя ошибку  $P$ . Из условий  $\partial P / \partial b_i = 0$  и  $\partial P / \partial C_0 = 0$  найдем оптимальные значения  $C_0$  и  $b_i$ , а также соответствующее им значение коэффициента корреляции:

$$C_0 = A_0 \frac{\sigma_C^2}{\langle C_{ij} A'_{ij} \rangle}, \quad b_i = B_i \frac{\sigma_C^2}{\langle C_{ij} A'_{ij} \rangle}, \quad \rho = \sqrt{\frac{A_0^2 + \langle C_{ij} A'_{ij} \rangle^2 \sigma_C^{-2}}{A_0^2 + \sigma_A^2}}.\tag{2.14}$$

С учетом (2.11) и (2.14) выражение можно переписать в виде:

$$P = 1 - \int_{-\alpha}^{\infty} Q(\gamma x + \beta) Dx - \int_{\alpha}^{\infty} Q(\gamma x - \beta) Dx, \quad (2.15)$$

где

$$Q(z) = \int_z^{\infty} Dx, \quad Dx = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}. \quad (2.16)$$

Здесь введены обозначения

$$\alpha = \frac{|B_i|}{\sqrt{N(A_0^2 + \sigma_A^2)}}, \quad \beta = \frac{\alpha}{\rho\sqrt{1-\rho^2}}, \quad \gamma = \frac{\rho}{\sqrt{1-\rho^2}}. \quad (2.17)$$

Анализ (2.14)-(2.17) показывает, что ошибка  $P$  не зависит от знаков величин  $A_0$  и  $B_i$ . С ростом величин  $|A_0|$  и  $|B_i|$  величина ошибки  $P$  быстро уменьшается. Наихудшие условия для дискретизации имеют место в случае, когда  $A_0 = 0$  и  $B_i = 0$ . В этом случае величина ошибки максимальна и описывается вытекающим из (2.15) выражением:

$$P = \frac{1}{2} - \frac{1}{\pi} \arcsin \rho_{\min}, \quad (2.18)$$

где  $\rho_{\min}$  - минимальное значение коэффициента корреляции, имеющее место при  $A_0 = 0$ :

$$\rho_{\min} = \frac{\langle C_{ij} A'_{ij} \rangle}{\sigma_C \sigma_A}. \quad (2.19)$$

Отметим, что все выражения получены в самом общем виде, без привязки к виду распределения матричных элементов  $A_{ij}$  и типу выбранного метода дискретизации. Поэтому можно сделать общий вывод: наилучшее представление матрицы  $\hat{A}'$  матрицей  $\hat{C}$  сводится к такому подбору метода дискретизации, при котором величина  $\rho_{\min}$  максимальна. В этом смысле выбранный нами метод дискретизации, направленный на максимальное ускорение алгоритма минимизации, как правило, не является наилучшим.

Выражениями (2.18) и (2.19) задается минимальное значение эффективности алгоритма минимизации. Эти выражения следует оптимизировать по длине отрезка с нулевым средним, находя  $l_0$  из условия

$$\frac{\partial P}{\partial l_0} = 0. \quad (2.20)$$

*Минимум функционала.* Процесс минимизации функционала (2.3) начинается с некоторой случайной точки пространства  $\mathbf{S}$ . Подчиняясь решающему правилу (2.8) нейронная сеть приходит в некоторое устойчивое состояние  $\mathbf{S}_0^*$ , являющееся минимумом дискретизированного функционала (2.9). Если из этой точки продолжить спуск с решающим правилом (2.2), то сеть придет в состояние  $\mathbf{S}_0$ , соответствующее минимуму функционала (2.3). На всем пути  $\mathbf{S} \rightarrow \mathbf{S}_0^* \rightarrow \mathbf{S}_0$  значение ошибки только уменьшается. Значение ошибки в исходной (случайной) точке анализировалось выше, см. (2.10)-(2.13). Проанализируем теперь вероятность ошибки в точке  $\mathbf{S}_0$  - минимуме функционала (2.3).

В отличие от случайной точки в минимуме направление локальных полей совпадает с состояниями нейронов

$$H_i s_{0i} > 0, \quad \forall i = \overline{1, N}. \quad (2.21)$$

Поэтому вероятность несовпадения направлений локальных полей  $\mathbf{H}$  и  $\mathbf{h}$  в минимуме  $\mathbf{S}_0 = (s_{01}, s_{02}, \dots, s_{0N})$  функционала нужно рассматривать при условии (2.21), т.е.

$$P_m = \Pr\{H_i h_i < 0 \mid H_i s_{0i} > 0\} = \frac{\Pr\{H_i s_{0i} > 0 \cap h_i s_{0i} < 0\}}{\Pr\{H_i s_{0i} > 0\}}. \quad (2.22)$$

Ограничимся анализом ситуации  $A_0 = 0$  и  $B_i = 0$  - наихудшей с точки зрения минимизации.

Прежде всего, вычленим из матрицы  $\hat{\mathbf{A}}$  слагаемое  $\hat{\mathbf{A}}_0$ , ответственное за образование минимума  $\mathbf{S}_0$ . Для этого представим матрицу  $\hat{\mathbf{A}}$  в виде

$$\hat{\mathbf{A}} = \hat{\mathbf{A}}_0 + \hat{\mathbf{A}}', \quad \hat{\mathbf{A}}_0 = r_0 \sigma_A \mathbf{S}_0 \mathbf{S}_0^+. \quad (2.23)$$

Коэффициент  $r_0$  найдем из условия отсутствия корреляции между элементами матриц  $\hat{\mathbf{A}}_0$  и  $\hat{\mathbf{A}}'$ . Рассчитав ковариацию матричных элементов и, положив ее равной нулю, для  $r_0$  получим

$$r_0 = -\frac{E_0 + \langle \mathbf{A} \rangle \delta}{1 - \delta^2}, \quad \delta = \frac{1}{N^2} \left[ \left( \sum_{i=1}^N s_{0i} \right)^2 - N \right]. \quad (2.24)$$

Здесь  $\langle \mathbf{A} \rangle$  - среднее значение элементов матрицы  $\hat{\mathbf{A}}$ , а  $\delta$  - величина с нулевым средним и малым стандартным отклонением  $\sigma_\delta = \sqrt{2}/N$ ,  $E_0$  - энергия минимума. В нашем случае можно положить  $\langle \mathbf{A} \rangle = 0$  и  $\delta = 0$ , тогда получаем

$$E_0 = -r_0. \quad (2.25)$$

Продолжая разложение (2.23) можно представить произвольную матрицу  $\hat{\mathbf{A}}$  в виде взвешенной суммы внешних произведений минимумов

$$\hat{\mathbf{A}} = \sigma_A \sum_{\mu=0}^{\infty} r_\mu \mathbf{S}_\mu \mathbf{S}_\mu^+, \quad \sum_{\mu=1}^{\infty} r_\mu = 1. \quad (2.26)$$

Для такого типа матриц в [120] показано, что любой из векторов  $\mathbf{S}_\mu$ , присутствующий в разложении матрицы  $\hat{\mathbf{A}}$ , будет минимумом функционала (2.3) тогда и только тогда, когда его статистический вес  $r_\mu$  больше или равен критическому значению

$$r_c = \frac{1}{2\sqrt{0.138N}}. \quad (2.27)$$

В дальнейшем, для простоты мы будем считать все коэффициенты  $r_\mu$  равными критическому значению (2.27).

Далее, зная распределение элементов матрицы  $\hat{\mathbf{A}}$ , мы сможем определить среднее, дисперсию и корреляцию величин  $\xi_i = H_i s_{0i}$  и  $\eta_i = h_i s_{0i}$ . Этих значений достаточно, чтобы определить вероятность несовпадения направлений локальных полей в минимуме исходного функционала (2.3)

$$P_m = \frac{1}{\Phi(\alpha)\sqrt{2\pi}} \int_0^\infty \Phi\left(-\frac{\beta - \rho x}{\sqrt{1 - \rho^2}}\right) e^{-\frac{1}{2}(x-\alpha)^2} dx. \quad (2.28)$$

Здесь введены следующие обозначения

$$x = \frac{\xi_i}{\sigma_\xi}, \quad \alpha = \frac{\langle \xi_i \rangle}{\sigma_\xi}, \quad \beta = \frac{\langle \eta_i \rangle}{\sigma_\eta}, \quad \Phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^z e^{-\frac{1}{2}x^2} dx. \quad (2.29)$$

В следующих параграфах приводятся примеры применения дискретизации к матрицам с равномерным и нормальным распределением элементов.

## §2.2. Матрицы с равномерным распределением элементов

Рассмотрим матрицу вида  $A_{ij} = A_0 + A'_{ij}$ , где  $A_0$  - некоторая константа, определяющая среднее значение элементов матрицы  $\hat{A}$ ,  $\hat{A}'$  - матрица с равномерным распределением элементов в диапазоне  $[-1; +1]$ . Решив систему (2.6), мы получим координаты отрезков, на которые надо разбить область распределения  $\hat{A}'$ :

$$x_k = \sqrt{\frac{k(k+1)}{q(q+1)} + \left[1 - \frac{k(k+1)}{q(q+1)}\right] l_0^2}, \quad k = 0, 1, \dots, q. \quad (2.30)$$

Отрезки отрицательных элементов  $A'_{ij}$  будут зеркальным отражением относительно центра распределения. Используя (2.18)-(2.20) находим оптимальное значение  $l_0$ :

$$l_0 = \frac{1}{2q+1}. \quad (2.31)$$

Тогда, из (2.30) и (2.31) следует, что дискретизация матрицы сводится к разбиению отрезка  $[-1; +1]$  на  $(2q+1)$  отрезков равной длины. В этом случае параметры дискретизированной матрицы

$$\sigma_C^2 = \frac{1}{4l_0^2} \sigma_A^2 (1 - l_0^2), \quad \langle C_{ij} A'_{ij} \rangle = \frac{1}{4l_0} \sigma_A^2 (1 - l_0^2). \quad (2.32)$$

Оптимальные параметры дискретизации будут равны

$$C_0 = \frac{(2q+1)}{2} A_0, \quad b_i = \frac{(2q+1)}{2} B_i, \quad (2.33)$$

$$\rho = \sqrt{1 - \frac{\sigma_A^2}{(2q+1)^2 (A_0^2 + \sigma_A^2)}}.$$

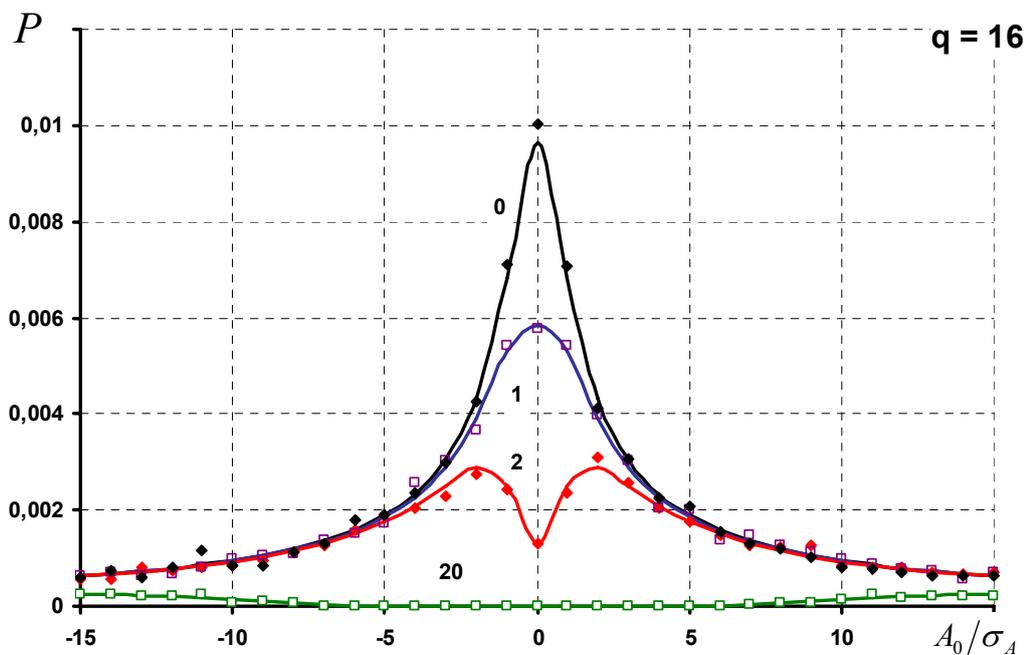
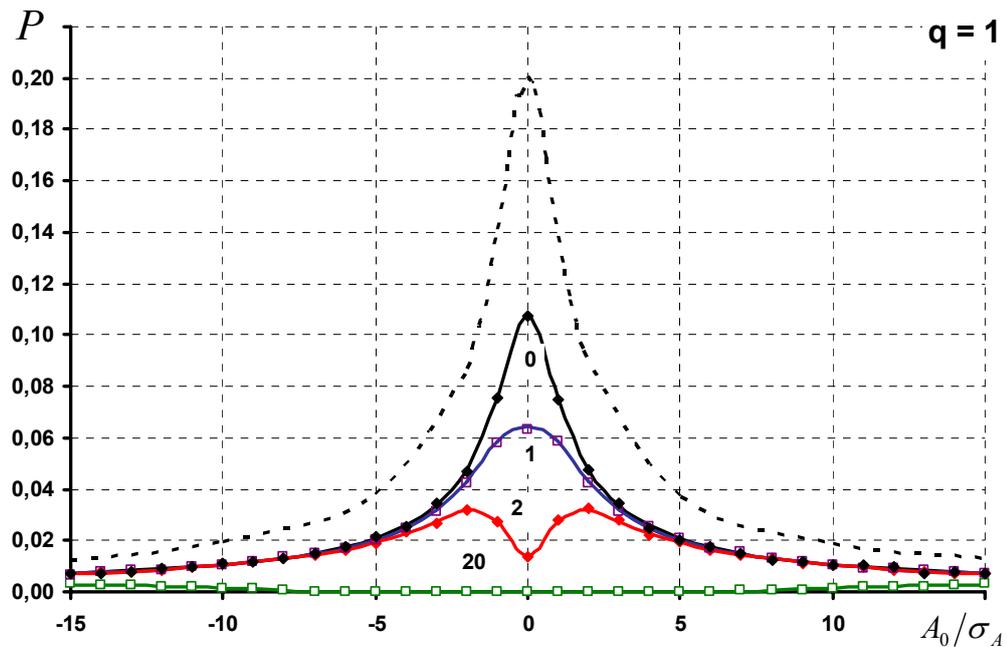
Как видим, величина коэффициента корреляции не зависит от знака  $A_0$ , с ростом  $|A_0|$  стремится к единице. Соответственно близка к нулю величина ошибки. Сказанное справедливо даже в наихудшей ситуации ( $A_0 = 0, B_i = 0$ ), когда коэффициент корреляции минимален, а величина ошибки

максимальна. В этом случае из (2.18)-(2.19) и (2.32)-(2.33) следуют простые оценочные выражения:

$$\rho_{\min} \approx 1 - \frac{1}{2(2q+1)}, \quad P_{\max} \approx \frac{1}{\pi(2q+1)}. \quad (2.34)$$

Как видим, даже в простейшем случае  $q=1$ , величина ошибки достаточно мала ( $P_{\max} \approx 0.11$ ). С ростом числа градаций величина  $P_{\max}$  быстро стремится к нулю.

Сказанное выше подкрепляется результатами численного эксперимента. На рисунке 2.1 приведена зависимость величины ошибки от величины  $|A_0|$  при различных значениях  $|B_i|$ . Графики приведены для двух значений числа градаций ( $q=1$  и  $q=16$ ). На первом графике пунктирной линией показаны результаты эксперимента с бинаризованной матрицей [121].



**Рис. 2.1.** Вероятность несовпадения направлений локальных полей в случайной точке конфигурационного пространства. Матрица  $\hat{A}$  имеет равномерное распределение матричных элементов. По горизонтальной оси отложено среднее значение элементов исходной матрицы  $\hat{A}$ . Маркерами показаны экспериментальные результаты, сплошные кривые построены по теоретическим выражениям (2.12). Величина порога выбиралась как  $\frac{B_i}{\sigma_A \sqrt{N}} = 0, 1, 2, 20$ . Для сравнения пунктирной линией на верхнем графике приведена кривая, соответствующая данным эксперимента с бинаризованной матрицей [121]. На нижнем рисунке для  $q=16$  стоит обратить внимание на масштаб. Ошибка при увеличении числа градаций снизилась в 10 раз.

Найдем теперь вероятность ошибки в точке  $S_0$ , соответствующей минимуму функционала. Зная распределение элементов матрицы  $\hat{A}$ , определим среднее и дисперсию величин  $\xi_i = H_i s_{0i}$  и  $\eta_i = h_i s_{0i}$ :

$$\langle \xi_i \rangle = r_0 \sigma_A N, \quad \sigma_{\xi}^2 = N \sigma_A^2 (1 - r_0^2),$$

$$\langle \eta_i \rangle = \frac{1}{2l_0} N r_0 \sigma_A (1 - l_0^2), \quad \sigma_{\eta}^2 = \frac{1}{4l_0^2} N \sigma_A^2 (1 - l_0^2) [1 - r_0^2 (1 - l_0^2)], \quad (2.35)$$

$$\rho = \sqrt{1 - \frac{l_0^2}{1 - r_0^2 (1 - l_0^2)}}.$$

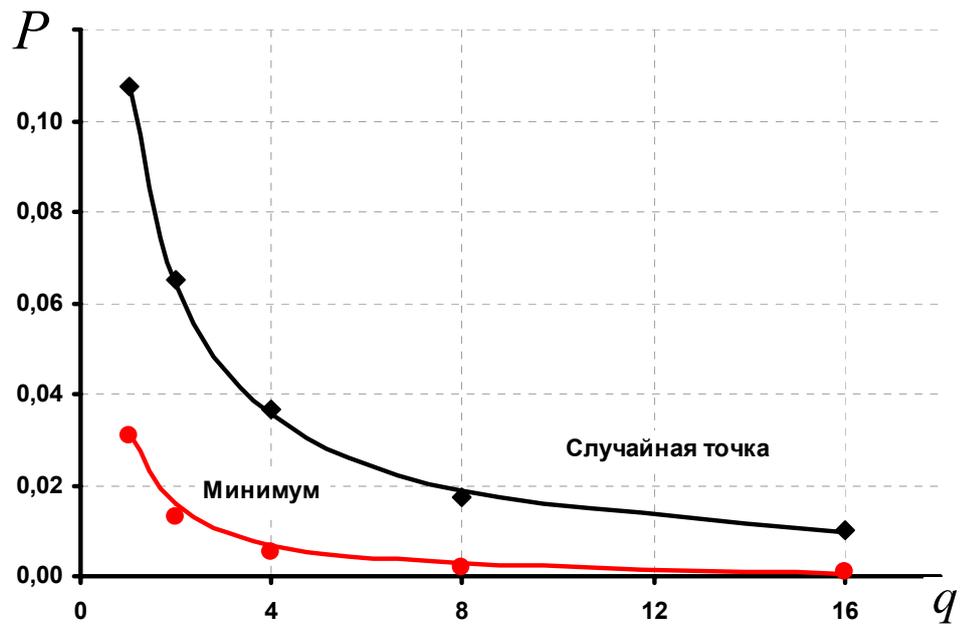
С учетом (2.35) вероятность ошибки определяется по формуле (2.28).

На рисунке 2.2 представлены зависимости вероятности ошибки от числа градаций. Данные получены для случая  $A_0 = 0$  и  $B_i = 0$ . Видно, что выражение (2.28) хорошо согласуется с экспериментом. Максимальное значение ошибки  $P_m \approx 0.03$  имеет место при  $q = 1$ . С ростом числа градаций вероятность ошибки убывает. Скорость убывания определяется вытекающим из (2.28) и (2.35) оценочным выражением

$$P_m \approx \frac{1}{\pi^2 \sqrt{\pi + 8q(q+1)}}, \quad (2.36)$$

справедливым при  $q \gg 1$  (при  $q \sim 1$  это выражение дает заниженную на 30% величину).

Так же на графике представлена зависимость ошибки от числа градаций в случайной точке пространства. Очевидно, что по мере спуска вероятность несовпадения направлений полей уменьшается, хотя и не обращается в нуль.



**Рис. 2.2.** Вероятность несовпадения направлений локальных полей от числа градаций в случайной точке пространства и в минимуме. Маркерами показаны экспериментальные результаты, сплошные кривые построены по теоретическим выражениям (2.34) и (2.18).

### §2.3. Матрицы с нормальным распределением элементов

Рассмотрим матрицу вида  $A_{ij} = A_0 + A'_{ij}$ , где  $A_0$  - некоторая константа, определяющая среднее значение элементов матрицы  $\hat{A}$ ,  $\hat{A}'$  - матрица с нормальным распределением элементов (нулевое среднее, единичная дисперсия). Решив систему (2.6), мы получим координаты отрезков, на которые надо разбить область распределения  $\hat{A}'$ :

$$x_k = \sqrt{l_0^2 - 2\sigma_A^2 \ln \left[ 1 - \frac{k(k+1)}{q(q+1)} \right]}, \quad k = 0, 1, \dots, q. \quad (2.37)$$

Отрезки отрицательных элементов  $A'_{ij}$  будут зеркальным отражением относительно центра распределения. Используя (2.18)-(2.20) находим оптимальное значение  $l_0$

q = 1	q = 2	q = 4	q = 8	q = 16
$l_0 = 0.61 \cdot \sigma_A$	$l_0 = 0.42 \cdot \sigma_A$	$l_0 = 0.28 \cdot \sigma_A$	$l_0 = 0.23 \cdot \sigma_A$	$l_0 = 0.21 \cdot \sigma_A$

**Таблица 2.1.** Размер нулевого отрезка для различного числа градаций и дисперсии элементов исходной матрицы.

В этом случае параметры дискретизированной матрицы

$$\langle C_{ij} A'_{ij} \rangle = \frac{2q+1}{3} \sigma_A \sqrt{\frac{2}{\pi}} e^{-\frac{l_0^2}{2\sigma_A^2}}, \quad \sigma_C^2 = 2 \sum_{k=1}^q k^2 P_k. \quad (2.38)$$

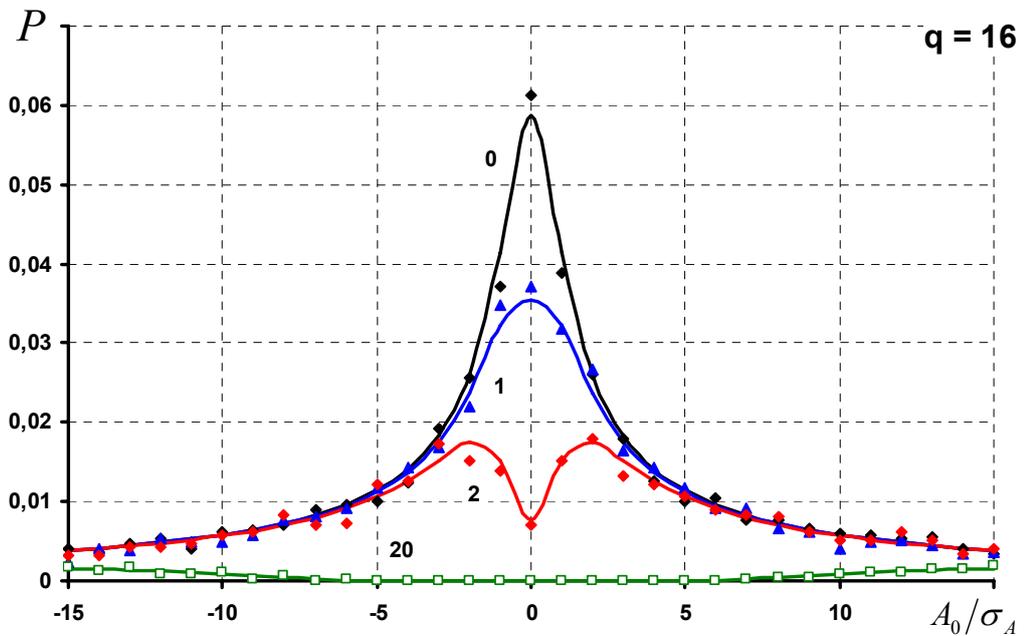
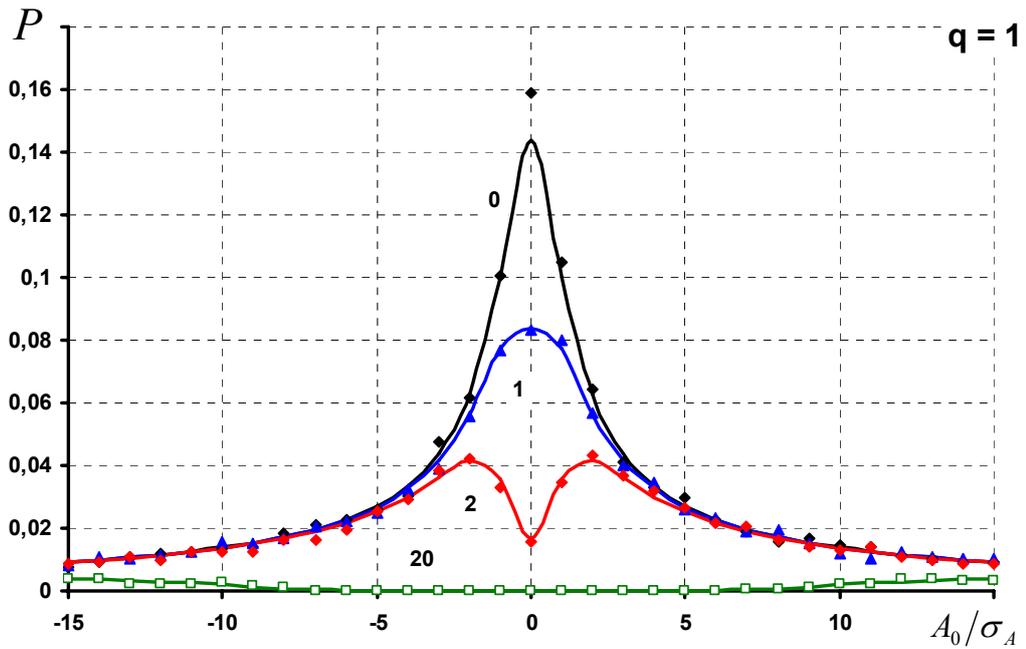
Здесь  $P_k$  - вероятность появления  $k$ -ого элемента,  $\sigma_C^2$  - дисперсия матрицы  $\hat{C}$ . Оптимальные параметры дискретизации определяются выражениями (2.11) и (2.14), корреляция локальных полей равна

$$\rho = \frac{(2q+1)}{3 \sqrt{\pi \sum_{k=1}^q k^2 P_k}} \exp\left(-\frac{l_0^2}{2\sigma_A^2}\right). \quad (2.39)$$

Как видим, величина коэффициента корреляции не зависит от знака  $A_0$ , с ростом  $|A_0|$  стремится к единице. Соответственно близка к нулю величина ошибки. Сказанное справедливо даже в наихудшей ситуации ( $A_0 = 0, B_i = 0$ ),

когда коэффициент корреляции минимален, а величина ошибки максимальна. Максимальная вероятность ошибки, оцененная выражениями (2.18) и (2.28), составляет  $P_{\max} \approx 0.16$  для  $q = 1$ .

Сказанное выше подкрепляется результатами численного эксперимента. На рисунке 2.3 приведена зависимость величины ошибки от величины  $|A_0|$  при различных значениях  $|B_i|$ . Графики приведены для двух значений числа градаций ( $q = 1$  и  $q = 16$ ). На первом графике пунктирной линией показаны результаты эксперимента с бинаризованной матрицей [121].



**Рис. 2.3.** Вероятность несовпадения направлений локальных полей в случайной точке конфигурационного пространства. Вероятность несовпадения направлений локальных полей в случайной точке конфигурационного пространства. Матрица  $\hat{A}$  имеет нормальное распределение матричных элементов. По горизонтальной оси отложено среднее значение элементов исходной матрицы  $\hat{A}$ . Маркерами показаны экспериментальные результаты, сплошные кривые построены по теоретическим выражениям (2.12). Величина порога выбиралась как  $\frac{B_i}{\sigma_A \sqrt{N}} = 0, 1, 2, 20$ .

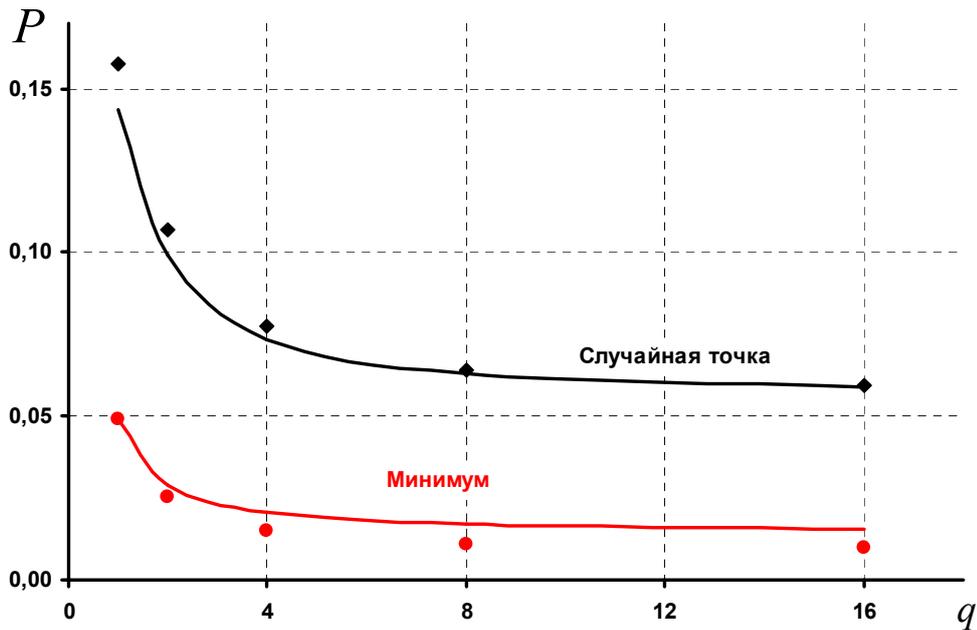
Найдем теперь вероятность ошибки в точке  $S_0$ , соответствующей минимуму функционала. Зная распределение элементов матрицы  $\hat{A}$ , определим среднее и дисперсию величин  $\xi_i = H_i s_{0i}$  и  $\eta_i = h_i s_{0i}$ :

$$\begin{aligned} \langle \xi_i \rangle &= r_0 \sigma_A N, \quad \sigma_{\xi}^2 = N \sigma_A^2 (1 - r_0^2), \\ \langle \eta_i \rangle &= N \sum_{k=-q}^q k P_k, \quad \sigma_{\eta}^2 = N \sum_{k=-q}^q k^2 P_k - \frac{\langle \eta_i \rangle^2}{N}, \\ \rho &= \frac{N^2 \langle C_{ij} A'_{ij} \rangle - \langle \xi_i \rangle \langle \eta_i \rangle}{N \sigma_{\xi} \sigma_{\eta}}, \end{aligned} \quad (2.40)$$

где

$$P_k = \Pr\{x_{k-1} < A_{ij} s_{0i} s_{0j} < x_k\} = \frac{1}{\sigma_A \sqrt{2\pi(1-r_0^2)}} \int_{x_{k-1}}^{x_k} \exp\left\{-\frac{(x - r_0 \sigma_A)^2}{2\sigma_A^2(1-r_0^2)}\right\} dx. \quad (2.41)$$

С учетом (3.37) вероятность ошибки определяется по формуле (2.28).



**Рис. 2.4.** Вероятность несовпадения направлений локальных полей от числа градаций в случайной точке пространства и в минимуме. Маркерами показаны экспериментальные результаты, сплошные кривые построены по теоретическим выражениям (2.18) и (2.28).

На рисунке 2.4 представлены зависимости вероятности ошибки от числа градаций. Данные получены для случая  $A_0 = 0$  и  $B_i = 0$ . Видно, что выражение (2.28) хорошо согласуется с экспериментом. Максимальное

значение ошибки  $P_m \approx 0.05$  имеет место при  $q = 1$ . С ростом числа градаций вероятность ошибки убывает.

Так же на графике представлена зависимость ошибки от числа градаций в случайной точке пространства. Очевидно, что по мере спуска вероятность несовпадения направлений полей уменьшается, хотя и не обращается в нуль.

## §2.4. Расстояние между минимумами

В предыдущих параграфах была определена вероятность несовпадения локальных полей дискретизированного и исходного функционалов. Тем самым мы показали, что минимизируя дискретизированный функционал (2.9) мы минимизируем также и исходный функционал (2.3). Однако, этого недостаточно пока, чтобы сказать, насколько такая минимизация эффективна. Интуитивно понятно, что чем больше будут расходиться направления локальных полей в стартовой точке, тем больше будут отличаться минимумы. Т.е. должна существовать некоторая зависимость между вероятностью ошибки в случайной точке и расстоянием между минимумами дискретизированного функционала и исходного. Эту зависимость мы и постараемся определить ниже.

Рассмотрим расстояние по Хеммингу между минимумом  $S_0^*$  дискретизированного функционала и минимумом  $S_0$  исходного функционала, в который пришла бы нейронная сеть, используя решающее правило (2.2) и точку  $S_0^*$  в качестве начальной. Расстоянием  $d$  будем называть долю несовпадающих компонент  $n$  векторов  $S_0$  и  $S_0^*$ , которое будет определяться как

$$d = \frac{n}{N}, \quad (2.42)$$

где  $N$  - размерность нейронной сети.

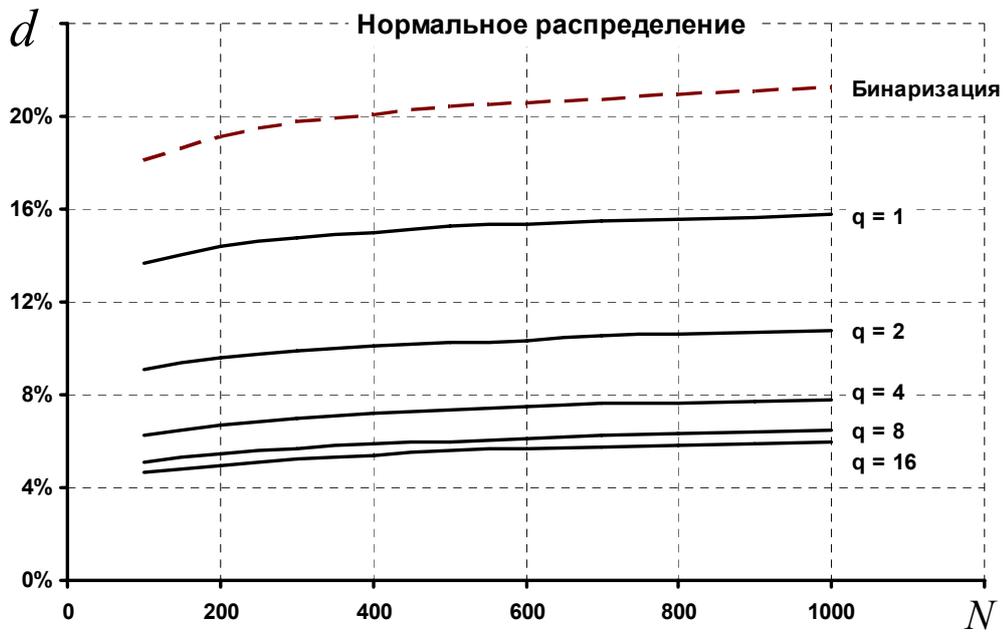
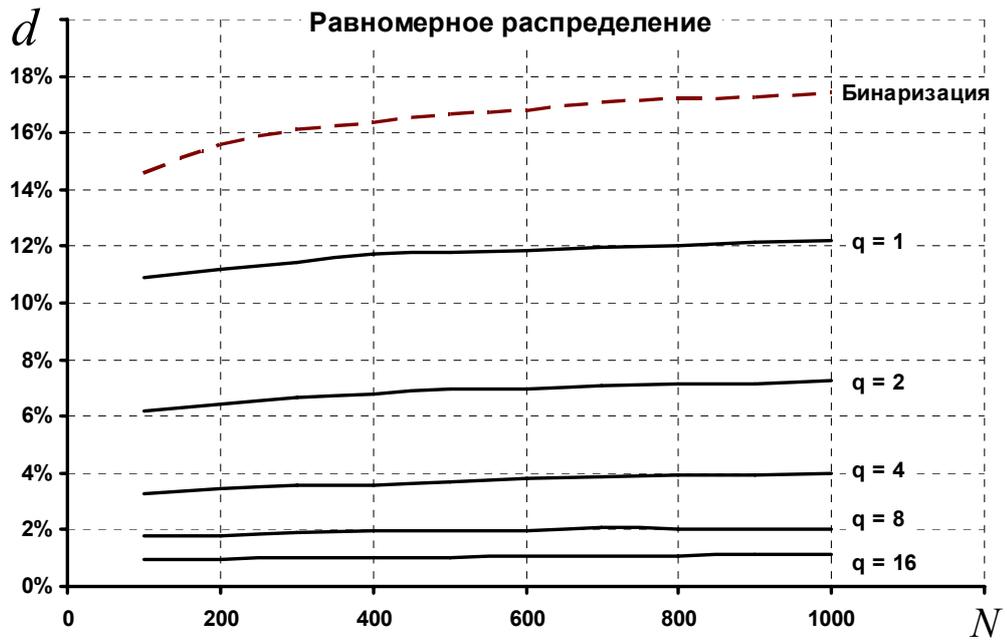
Так как точка  $S_0^*$  для дискретизированного функционала (2.9) может быть как минимумом, так и случайной точкой, то нет никакой возможности (на данный момент) аналитически определить расстояние  $d$ . Поэтому были проведены вычислительные эксперименты, в которых определялось среднее расстояние между минимума для различных размерностей и числа градаций. Из графиков видно, что расстояние практически не зависит от размерности сети, а также уменьшается с ростом числа градаций. Подобное поведение наблюдается у вероятности ошибки в случайной точке пространства.

Поэтому логично предположить, что расстояние определяется исключительно вероятностью несовпадения полей  $P$ , т.е.

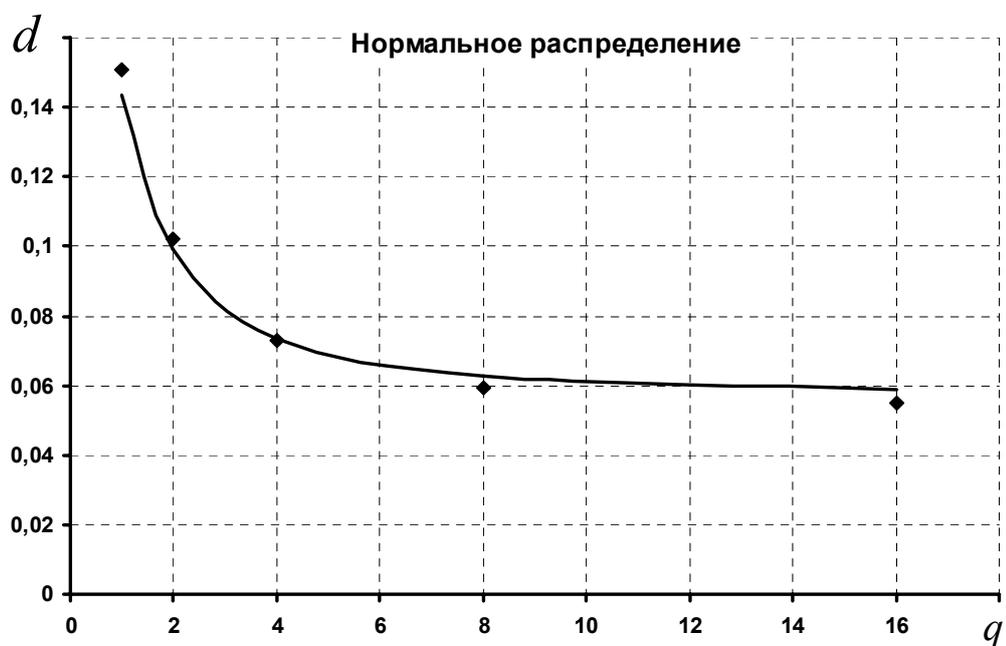
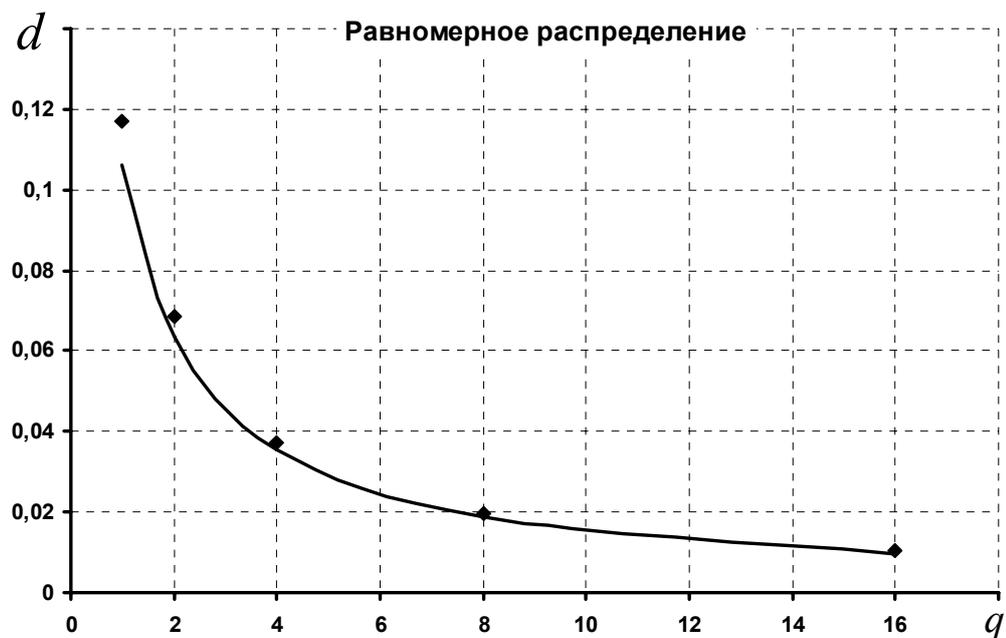
$$d = P \text{ или } n = PN. \quad (2.43)$$

На рисунке 2.6 показана экспериментальная зависимость расстояния от числа градаций, а также зависимость (2.43). Видно, что зависимость (2.43) хорошо описывает необходимую нам зависимость. Тем самым, определив вероятность ошибки с помощью выражения (2.18), можно сразу же оценить и Хеммингово расстояние между найденным минимумом дискретизированного функционала и истинным минимумом, до которого сеть не добралась. Зная такую информацию можно принять решение продолжать ли спуск от точки  $S_0^*$  до точки  $S_0$  или же остановиться (в зависимости от требований поставленной задачи).

Для матриц с равномерным распределением элементов это расстояние меняется от 12% для  $q = 1$  до 1% для  $q = 16$ . При бинаризации хеммингово расстояние между минимумами составляло 17%. Для матриц с нормальным распределением элементов хеммингово расстояние меняется от 15% для  $q = 1$  до 5% для  $q = 16$ . При бинаризации тех же матриц расстояние составляет 20%.



**Рис. 2.5.** Зависимость хеммингова расстояния между минимумами исходного и дискретизированного функционалов при различном числе градаций.



**Рис. 2.6.** Зависимость хеммингова расстояния между минимумами исходного и дискретизированного функционалов от числа градаций. Маркеры – экспериментальные результаты, сплошная линия – оценка по (2.43).

Хотя возможность оценить Хеммингово расстояние между минимумами еще до того, как сеть начала свою работу и является полезной, тем не менее, интереснее знать, сколько по энергии мы не дойдем до истинного минимума, ведь нейронная сеть минимизирует функционал (2.3) и

именно он является мерой эффективности минимизации. Оценим теперь расстояние между точками  $S_0$  и  $S_0^*$  по энергии.

Минимумы  $S_0$  и  $S_0^*$  различаются  $n$  координатами, т.е. достаточно изменить  $n < 0.2N$  нейронов в векторе  $S_0^*$  и мы получим минимум исходного функционала. На деле же, нейронная сеть Хопфилда при спуске из точки  $S_0^*$  в минимум  $S_0$  изменит порядка  $0.9N$  нейронов, т.е. некоторые нейроны изменят свое состояние четное число раз, и их вклад в изменение энергии в итоге окажется нулевым. Поэтому при оценке расстояния эти лишние действия сети не учитывать и полагать, что свое состояние изменило только  $n$  нейронов. Из динамики нейронной сети Хопфилда ясно, что при изменении состояния нейрона величина энергии изменяется на величину локального поля, действующего на этот нейрон. Таким образом, при каждом изменении состояния нейрона энергия меняется в среднем на величину  $de$  равную среднему значению модуля локального поля. Так как локальные поля в случайной точке распределены по нормальному закону с нулевым средним и дисперсией равной дисперсии матрицы, то получим

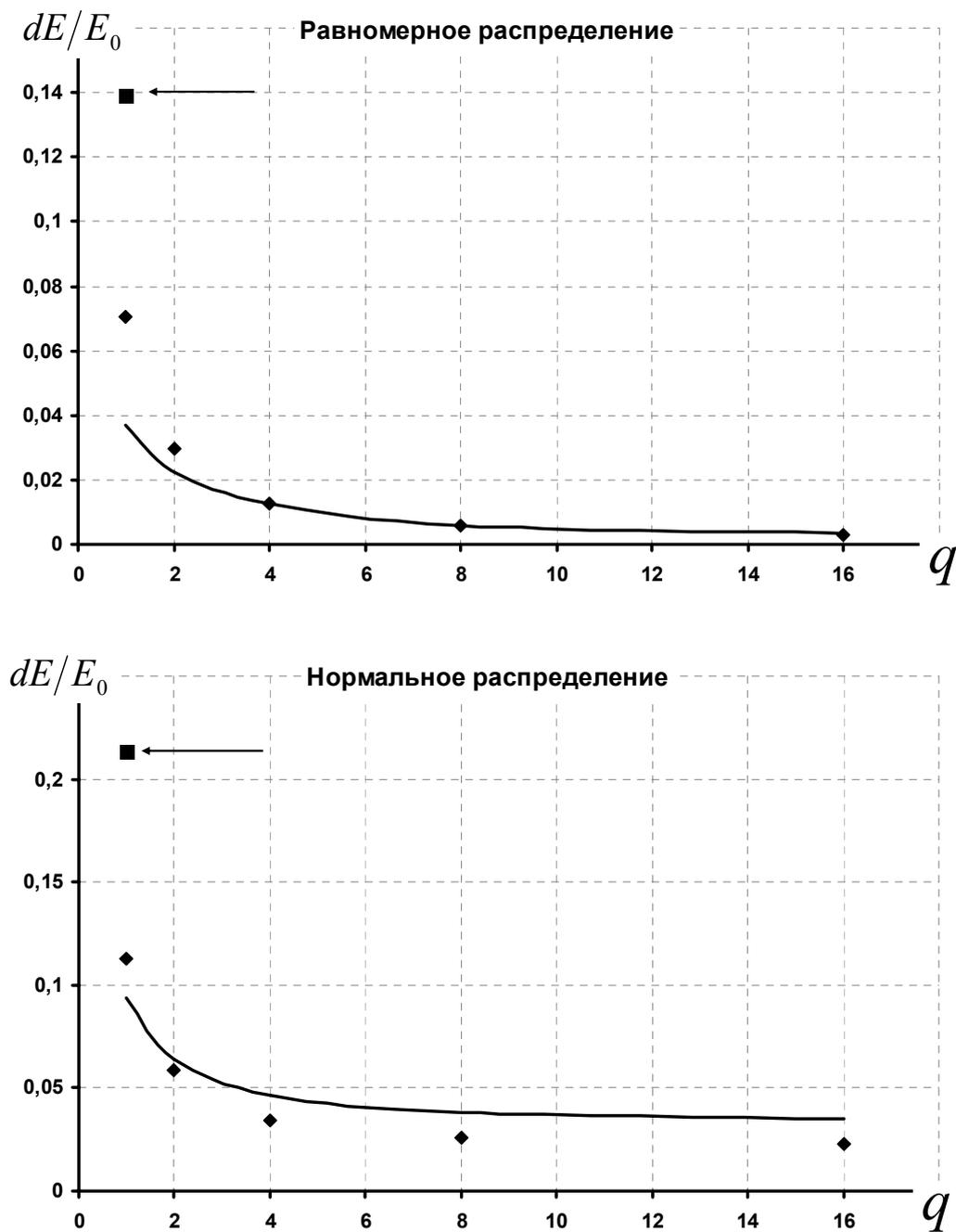
$$de = \sigma_A \sqrt{\frac{2N}{\pi}}. \quad (2.44)$$

Теперь, зная число неудовлетворенных нейронов (2.43) и шаг, с которым меняется энергия, можно определить расстояние по энергии между точками  $S_0$  и  $S_0^*$ :

$$dE = \sigma_A PN \sqrt{\frac{2N}{\pi}}. \quad (2.45)$$

На рисунке 2.7 показаны зависимости расстояния между минимумами исходного и дискретизированного функционалов по энергии. Видно, что оценочные выражения описывают характер зависимости расстояния по энергии между минимума. Разница в энергиях для матриц с равномерным распределением элементов меняется от 7% для  $q = 1$  до 0.2% для  $q = 16$ . При бинаризации – 14%. Для матриц с нормальным распределением расстояние

по энергии между минимума дискретизированного минимума и минимума, полученного после коррекции меняется от 11% для  $q=1$  до 2% для  $q=16$ . При бинаризации – 21%.



**Рис. 2.7.** Расстояние между минимумами исходного и дискретизированного функционалов по энергии. Маркеры – экспериментальные данные, сплошная линия – теоретическая оценка (2.45). Стрелкой обозначено значение, получаемое при бинаризации.

## §2.5. Выводы

В настоящей главе были получены следующие результаты:

1. *Описана процедура дискретизации.* Суть дискретизации заключается в замене вещественных элементов исходной матрицы связей  $\hat{A}$  в квадратичном функционале  $E$  (см. (2.3)) целыми числами в некотором диапазоне  $[-q; +q]$ , где  $q$  - число градаций, задаваемый пользователем параметр.
2. *Введены два основных параметра, позволяющих управлять характеристиками дискретизации по желанию пользователя:*  $q$  - число градаций и  $l_0$  - размер нулевого отрезка. В работе [121] рассматривалась процедура бинаризации, в которой элементы матрицы связей заменялись их знаками (т.е.  $\pm 1$ ), а ноль оставался без изменений. Дискретизация является общим случаем бинаризации с двумя основными отличиями. Во-первых, при дискретизации элементы матрицы связей заменяются целыми числами из диапазона, задаваемого числом градаций, а не просто знаками элементов. Во-вторых, нулем заменяются ноль и элементы, близкие к нулю в некоторой области, которая задается параметром  $l_0$ . Оптимальный подбор числа градаций и размера нулевого отрезка позволит наилучшим образом провести дискретизацию матрицы и получить наилучшие результаты.
3. *Получены аналитические оценки вероятности несовпадения направлений локальных полей дискретизированного и исходного функционалов. Ее наихудшее значение определяется выражением*

$$P = \frac{1}{2} - \frac{1}{\pi} \arcsin \rho_{\min}, \quad \rho_{\min} = \frac{\langle C_{ij} A'_{ij} \rangle}{\sigma_C \sigma_A}.$$

Замена исходной матрицы связей  $\hat{A}$  ее дискретизированным аналогом  $\hat{C}$  приводит к образованию нового функционала - дискретизированного  $\varepsilon$  (см. (2.9)). Минимумы дискретизированного функционала, в основном, отличаются от минимумов исходного

функционала. Поэтому, чтобы с уверенностью сказать, что минимизируя функционал  $\varepsilon$  минимизируется и функционал  $E$  необходимо, чтобы вероятность  $P$  несовпадения локальных полей исходного и дискретизированного функционалов была как можно меньше.

4. Процедура дискретизации продемонстрирована на примерах матриц с нормальным и равномерным распределениями матричных элементов.

Для матриц с равномерным распределением элементов при  $q=1$  -  $P \approx 0.11$  и с ростом числа градаций снижается до  $P \approx 0.01$  для  $q=16$ . Для тех же матриц при бинаризации ошибка составляет  $P \approx 0.2$ , практически в 2 раз больше худшего значения при дискретизации. Для матриц с нормальным распределением ошибка меняется от значения  $P \approx 0.16$  до  $P \approx 0.06$ . Несмотря на то, что тип распределения влияет на эффективность дискретизации, ошибка все равно остается достаточно малой величиной.

5. Получена оценка хеммингова расстояния между минимумом дискретизированного функционала и минимумом исходного функционала, в который пришла бы исходная сеть из дискретизированного минимума. Хеммингово расстояние между минимумами оказалось пропорционально вероятности несовпадения направлений локальных полей

$$n = PN .$$

Минимумы дискретизированного функционала не являются минимумами исходного функционала. Однако можно провести коррекцию дискретизированного минимума, используя его как стартовое состояние исходной сети, и найти таким образом ближайший минимум функционала  $E$ . Экспериментальные наблюдения показали число различных компонент дискретизированного минимума и минимума исходно функционала (т.е. Хеммингово расстояние)

пропорционально вероятности несовпадения направлений локальных полей в случайной точке. Для матриц с равномерным распределением элементов это расстояние меняется от 12% для  $q = 1$  до 1% для  $q = 16$ . При бинаризации хеммингово расстояние между минимумами составляло 17%. Для матриц с нормальным распределением элементов хеммингово расстояние меняется от 15% для  $q = 1$  до 5% для  $q = 16$ . При бинаризации тех же матриц расстояние составляет 20%.

6. *Получена оценка расстояния по энергии между минимумом дискретизированного функционала и минимумом исходного функционала, в который пришла бы исходная сеть из дискретизированного минимума:*

$$dE = \sigma_A PN \sqrt{\frac{2N}{\pi}}.$$

Исследования показали, что разница в энергиях для матриц с равномерным распределением элементов меняется от 7% для  $q = 1$  до 0.2% для  $q = 16$ . При бинаризации – 14%. Для матриц с нормальным распределением расстояние по энергии между минимума дискретизированного минимума и минимума, полученного после коррекции меняется от 11% для  $q = 1$  до 2% для  $q = 16$ . При бинаризации – 21%.

7. *Полученные аналитические оценки позволяют оценить эффективность применения дискретизации еще до ее непосредственного применения на практике.*
8. *Все аналитические оценки подтверждены множеством численных экспериментов, в главе приводятся рисунки, демонстрирующие согласованность теоретических и экспериментальных наблюдений.*

## ГЛАВА 3. АЛГОРИТМЫ ПОИСКА МИНИМУМОВ

Основной целью дискретизации является максимальное увеличение скорости минимизационного алгоритма при минимальной потере в качестве минимизации. В настоящей главе будут рассмотрены алгоритмы на основе дискретизации, проанализирована скорость алгоритмов и глубина находимых минимумов в зависимости от числа градаций.

### *§3.1. Одноэтапный алгоритм*

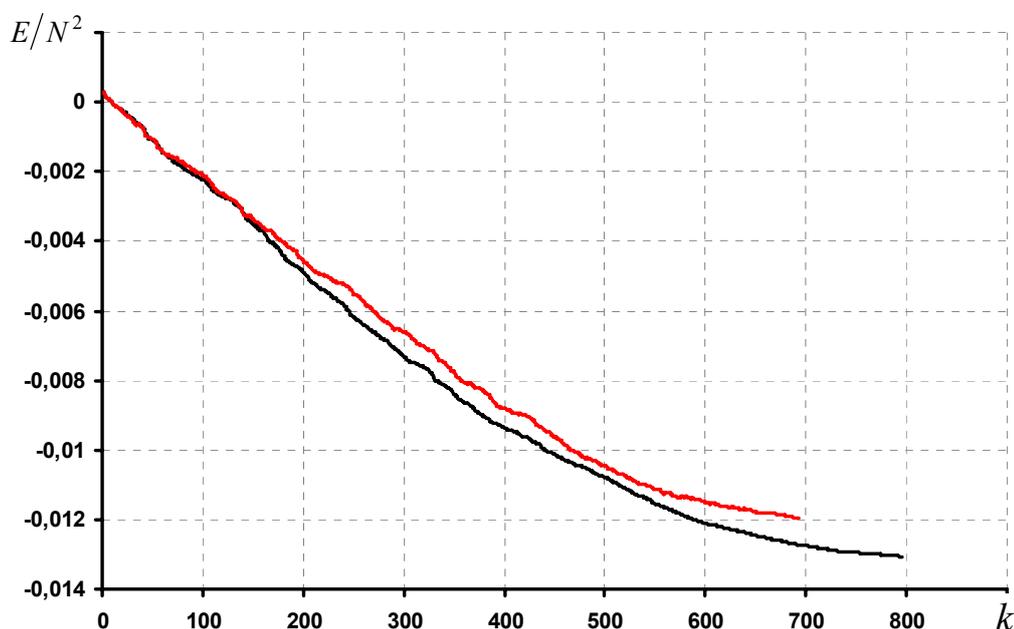
Во многих прикладных задачах дискретной оптимизации достаточно минимизировать функционал, а не найти его минимум. В этом случае удобно использовать дискретизацию. Действительно, за счет перехода к дискретизированной матрице экономится объем оперативной памяти, поэтому можно решать задачи большей размерности. Конечно, эффективность минимизации будет хуже в сравнении с обычной моделью Хопфилда. Тем не менее, дискретизация позволяет работать с размерностями недоступными обычной нейронной сети. Однако, рассмотрев зависимость эффективности минимизации от числа градаций, можно найти, что при достаточно большом числе градаций достигается удовлетворительная эффективность минимизации, а так же скорость алгоритма.

Одноэтапным алгоритмом в дальнейшем будем называть просто минимизацию дискретизированного функционала без дальнейшей коррекции минимумов. Алгоритм состоит из следующих шагов.

1. Дискретизируем исходную матрицу  $\hat{A}$ .
2. Находим минимумы дискретизированного функционала, используя динамику из главы 2. Проводится серия стартов из случайных точек.
3. Вычисляем значение дискретизированного функционала в минимуме и выбираем самый глубокий минимум.

Что же дает такой подход? Так как исходная матрица не участвует в минимизации, то нет никакой необходимости ее хранить или загружать в оперативную память. Это позволит значительно снизить требования к

оперативной памяти, а так же решать минимизационные задачи, которые нельзя было решить обычной нейронной сетью Хопфилда. Действительно, если каждый элемент матрицы  $\hat{A}$  занимает 8 байт памяти, то переход к дискретизированной матрице снизит объем занимаемой памяти, например, от 10 ( $q=16$ ) до 32 ( $q=1$ ) раз. Конечно, одноэтапный алгоритм предназначен для быстрой минимизации функционала (2.3), а не для нахождения его минимумов. Далее будет показано, какой вид имеют распределения минимумов функционала  $\varepsilon$  и насколько эффективно одноэтапный алгоритм позволяет решать минимизационную задачу.



**Рис. 3.1.** Изменение энергии при изменении нейронном своего состоянии в процессе спуска. Верхняя кривая – одноэтапный алгоритм, нижняя кривая – стандартная модель Хопфилда.

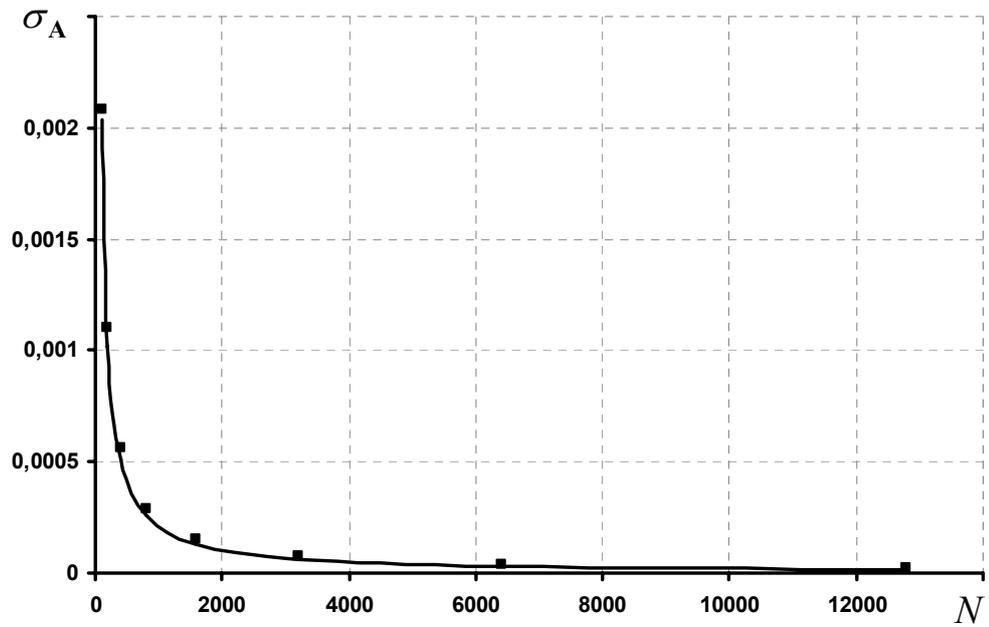
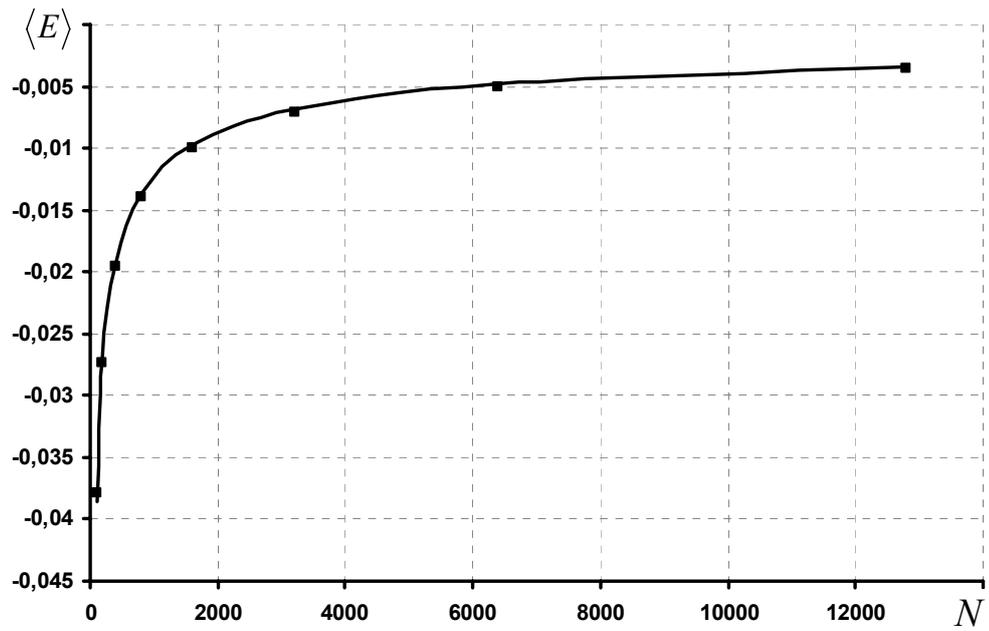
В главе 2 было показано, что направления локальных полей дискретизированного и исходного функционалов в случайной точке пространства не совпадают с некоторой вероятностью. Поэтому, не смотря на одинаковую стартовую точку, минимизационные пути сети на дискретизированной матрице и на исходной матрице будут различными. Сказанное продемонстрировано на рисунке 3.1. На рисунке изображено как меняется величина функционала (2.3) при изменении состояния нейрона.

Верхняя кривая соответствует минимизации с дискретизированной матрицей ( $q = 1$ ). Как видно, минимизация дискретизированного функционала приводит к минимизации исходного функционала. Однако минимумы дискретизированного функционала обычно не являются минимумами исходного функционала. Поэтому нельзя использовать только динамику (2.8) для нахождения устойчивых состояний функционала  $E$ , но этого достаточно для быстрой минимизации  $E$ .

Рассмотрим свойства алгоритма. Распределения энергий минимумов исходного функционала стандартной сети Хопфилда описываются некоторым распределением, среднее и стандартное отклонение которого можно оценить по формулам

$$\langle E \rangle = -\frac{\sigma_A N^2}{4\sqrt{0.14N}}, \quad \sigma_E = \frac{\sigma_A N \sqrt{2(1 - \langle E \rangle^2 / N^4)}}{4}. \quad (3.1)$$

Выражения (3.1) являются оценочными, так как на данный момент не существует полученных строго аналитическим путем выражений. На рисунке 3.2 представлены экспериментальные зависимости средней энергии минимумов и их стандартное отклонение (маркеры на графиках). Сплошные кривые построены по формулам (3.1). Видно, что представленные выражения хорошо описывают эксперимент.

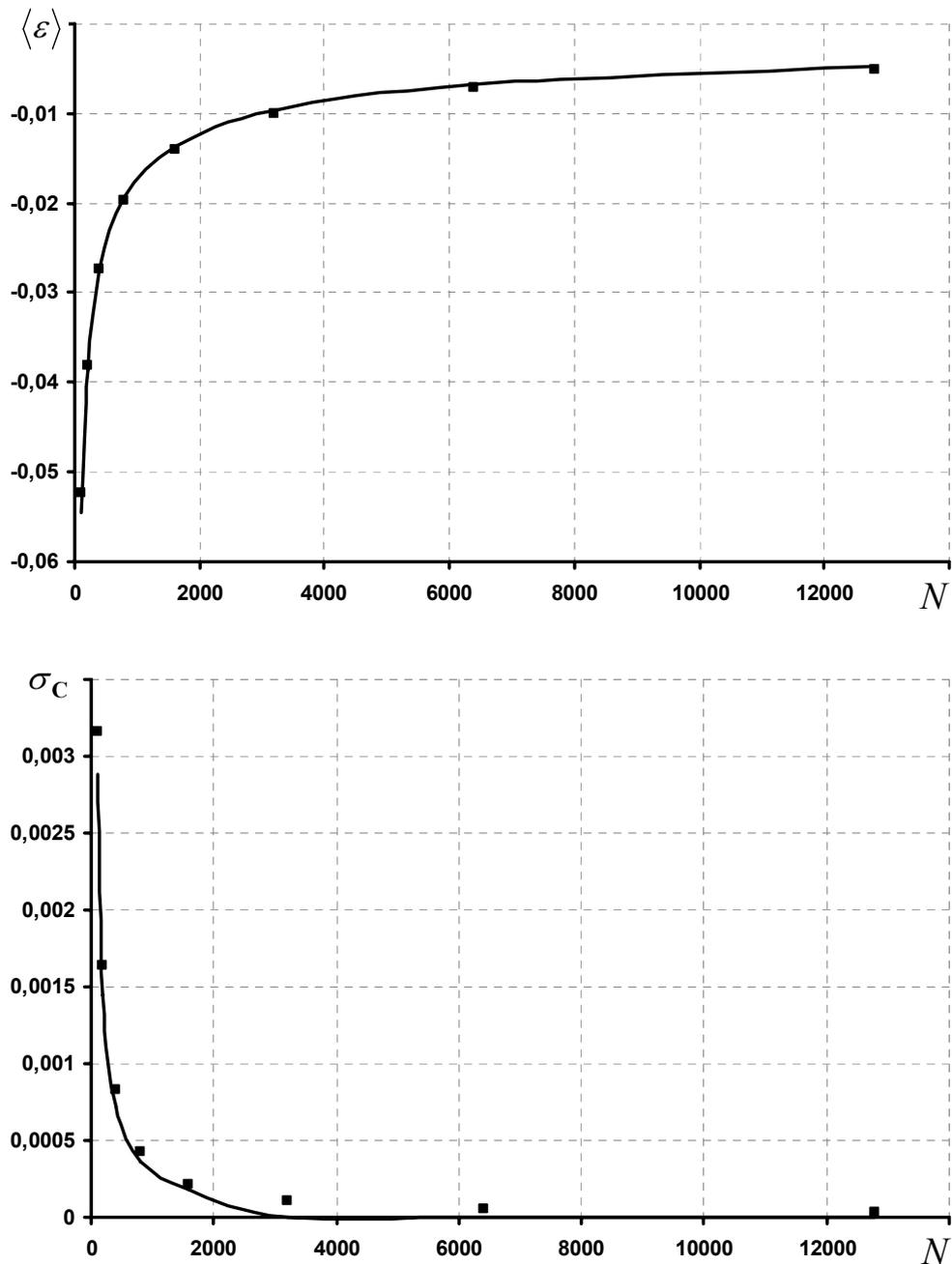


**Рис. 3.2.** Зависимость среднего значения (верхний график) и стандартного отклонения (нижний график) минимумов исходного функционала. Маркеры – экспериментальные данные, сплошные кривые – теоретические оценки.

Заметим, что выражения (3.1) зависят только от дисперсии матрицы и ее размерности. Поэтому можно предположить, что при замене исходной матрицы ее дискретизированным аналогом, выражения для средней энергии останутся тем же, только с соответствующими параметрами новой матрицы. Т.е. выражения (3.1) преобразуются для дискретизированного функционала в следующий вид

$$\langle \varepsilon \rangle = -\frac{\sigma_c N^2}{4\sqrt{0.14N}}, \quad \sigma_\varepsilon = \frac{\sigma_c N \sqrt{2(1 - \langle \varepsilon \rangle^2 / N^4)}}{4}. \quad (3.2)$$

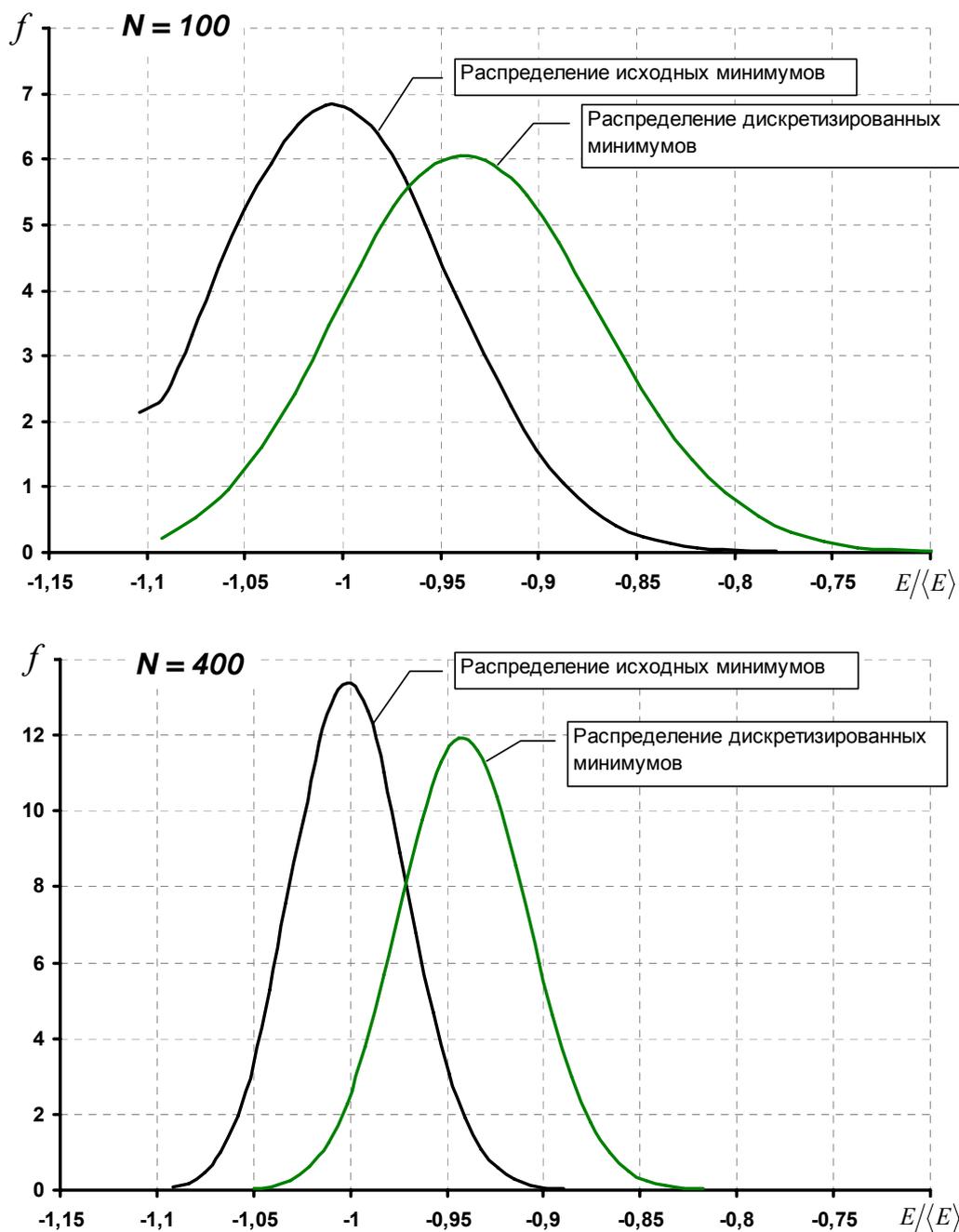
На рисунке 3.3 представлены экспериментальные и теоретические зависимости среднего и стандартного отклонений энергий минимумов дискретизированного функционала. Как видно, (3.2) так же неплохо описывают поведение  $\varepsilon$ . В дальнейшем эти выражения помогут нам в построении алгоритмов.



**Рис. 3.3.** Зависимость среднего значения (верхний график) и стандартного отклонения (нижний график) минимумов дискретизированного функционала. Маркеры – экспериментальные данные, сплошные кривые – теоретические оценки.

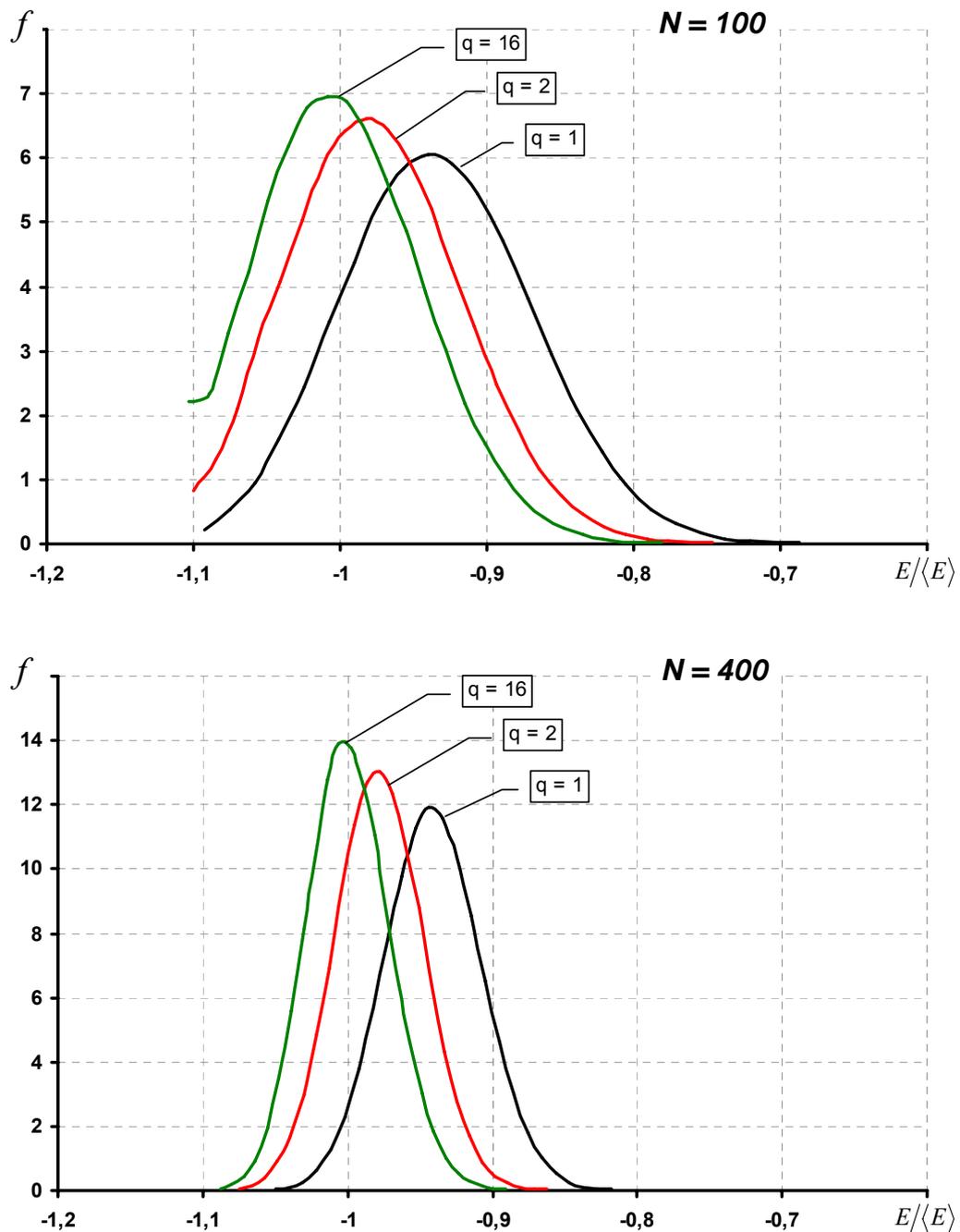
Рассмотрим теперь распределения минимумов полученных стандартной сетью Хопфилда и одноэтапным алгоритмом. На рисунке 3.4 показано взаимное расположение распределений энергий минимумов исходного функционала и энергий минимумов дискретизированного функционала, вычисленных по формуле (2.3). Все значения нормированы на среднее значение энергии минимумов исходного функционала. Видно, что

распределения имеют гауссовый вид и довольно близко расположены друг к другу. Т.е. переход к одноэтапному алгоритму всего лишь смещает диапазон найденных минимумов. Расстояние сдвига распределений было оценено в главе 2. На рисунке 3.4 представлен случай  $q=1$ . Но согласно результатам, полученным в главе 2, с ростом числа градаций и минимумы находятся глубже.



**Рис. 3.4.** Взаимное расположение минимумов одноэтапного алгоритма и минимумов обычной нейронной сети Хопфилда.

На рисунке 3.5 представлена зависимость распределений энергий минимумов дискретизированного функционала от числа градаций. Видим, что при  $q=16$  распределение практически совпадает с результатами стандартной модели Хопфилда (пики распределений расположены на -1).



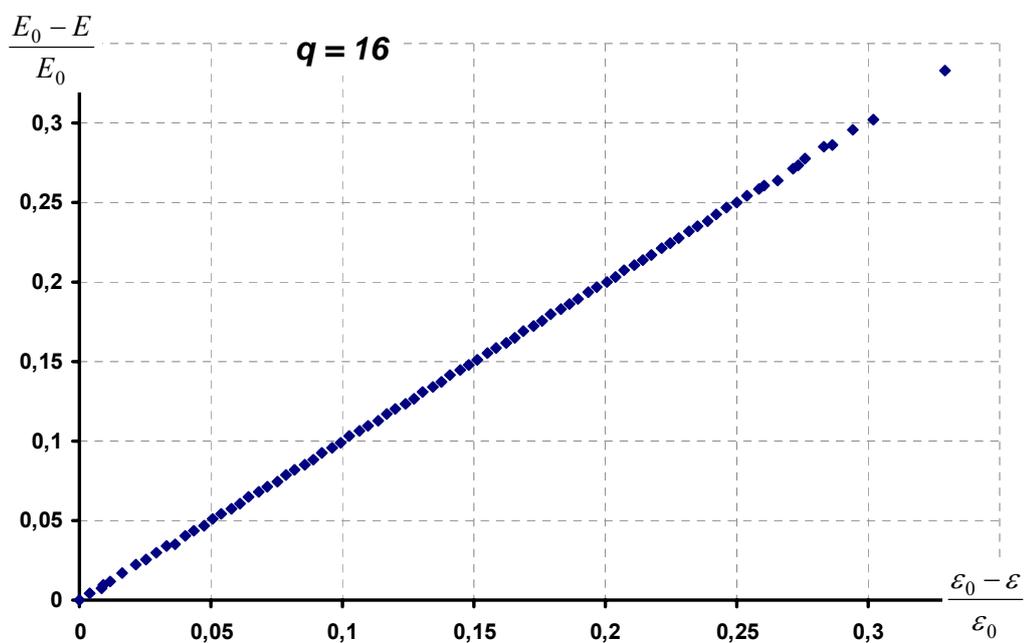
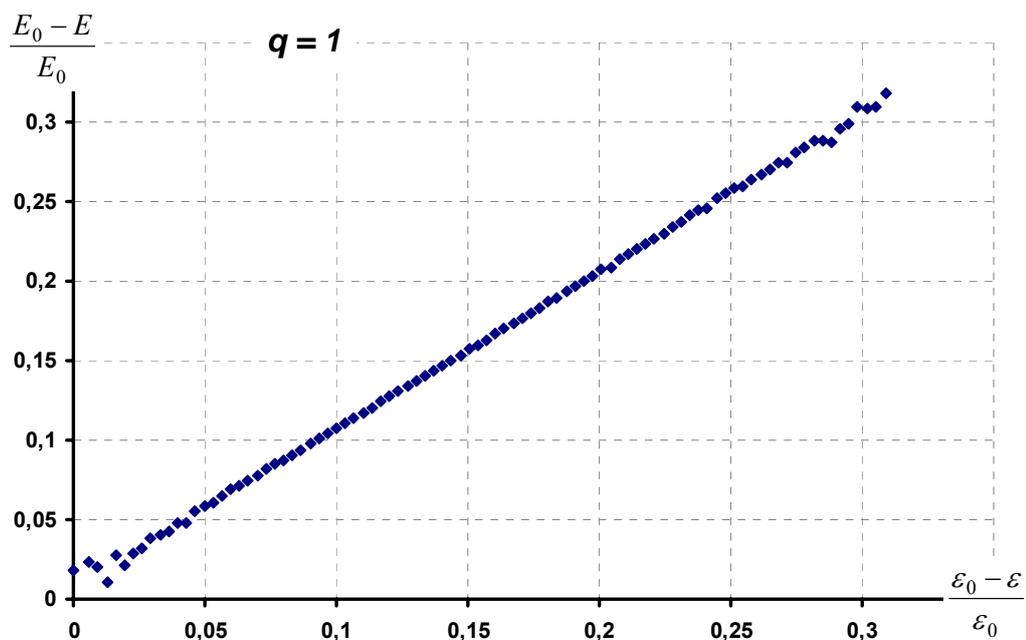
**Рис. 3.5.** Распределение плотности вероятности нахождения минимумов одноэтапного алгоритма по энергии исходного функционала для разного числа градаций.

Хотя при одноэтапном алгоритме минимизируется также исходный функционал, тем не менее, самый глубокий минимум дискретизированного

функционала необязательно будет самым глубоким исходного функционала. Вычислять энергию функционала (2.3) для каждого минимума довольно накладно и сильно затормаживает алгоритм. Поэтому удобно было бы определять глубину минимума по энергии дискретизированного функционала.

На рисунке 3.6 представлены зависимости энергий одних и тех же состояний на исходном функционале и дискретизированном функционале. Как видно, на большей части диапазона энергий минимумов есть соответствие между глубинами минимумов. Тем не менее, при малом числе градаций в области самых глубоких минимумов наблюдается рассогласование энергий. Т.е. самый глубокий минимум  $\varepsilon$  не является самым глубоким минимумом  $E$ . И все же при минимизации достаточно определить 10-15 самых глубоких минимумов дискретизированного функционала, а затем вычислить для них энергию  $E$  и выбрать самый глубокий. При больших значениях числа градаций ( $q > 8$ ) подобный отбор делать необязательно. Это показывает, что энергетические поверхности дискретизированного и исходного функционалов совпадают с большой корреляцией.

С учетом сказанного, одноэтапный алгоритм можно сформулировать следующим образом: после дискретизации матрицы определяются минимумы дискретизированного функционала, лежащие глубже  $1.5\sigma_\varepsilon$  от среднего значения  $\bar{\varepsilon}$ . Затем, в зависимости от цели задачи, можно либо ограничиться самым глубоким минимумом дискретизированного функционала, либо вычислить соответствующие им энергии исходного функционала и выбрать самое глубокое состояние из них. Второй способ более медленный, так как вычисление энергии исходного функционала требует  $\sim 2N^2$  операций. Вычисление энергии дискретизированного функционала может быть выполнено быстро суммированием локальных полей (уже известных на последнем этапе минимизации) с соответствующими знаками.



**Рис. 3.6.** Соотношение энергий минимумов одноэтапного алгоритма на исходном функционале и дискретизированном функционале.

### §3.2. Двухэтапный алгоритм

Конечно, найдется множество задач, в которых достаточно просто минимизировать функционал и на этом остановиться. Однако часто все же необходимо найти устойчивую точку (минимум)  $E$ . В этом случае достаточно добавить к одноэтапному алгоритму еще один этап: используя стандартную модель Хопфилда, спуститься из минимума одноэтапного алгоритма.

Двухэтапный алгоритм можно записать следующим образом.

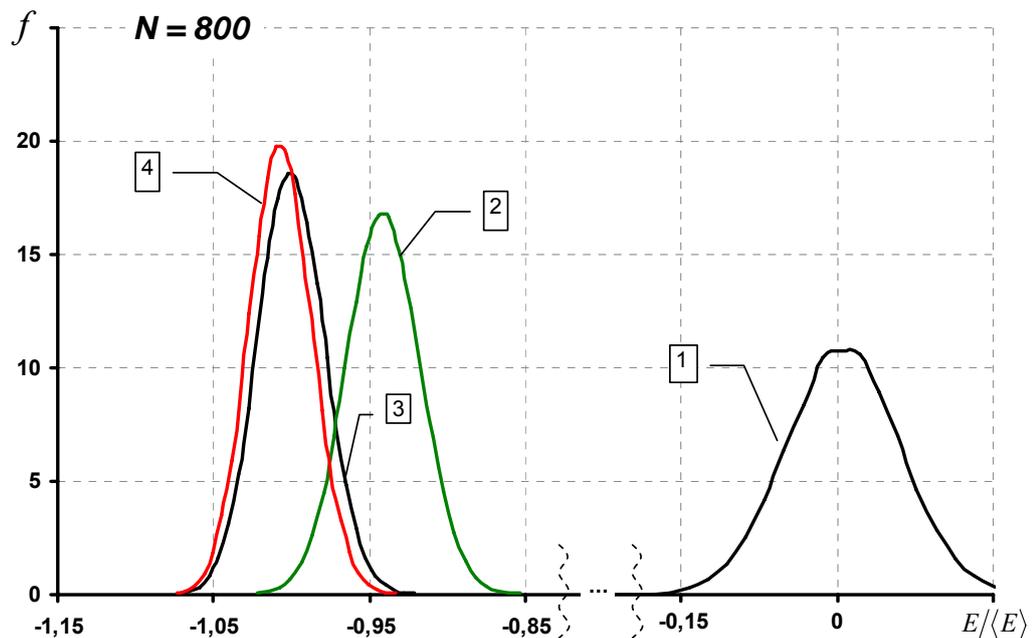
1. Из случайной точки пространства  $S$ , используя одноэтапный алгоритм, приходим в некоторую точку  $S_0^*$ , являющуюся минимумом дискретизированного функционала  $\varepsilon$ .
2. Из состояния  $S_0^*$  спускаемся, используя обычную модель Хопфилда на исходной матрице, в минимум  $S_0$  функционала  $E$ .

Второй этап – коррекции – начинается из состояний  $S_0^*$ , которые расположены глубже случайных точек  $S$ . Поэтому логично предположить, что и минимумы, получаемые двухэтапным алгоритмом, будут лучше, чем минимумы стандартной модели. Однако надо помнить, что у нейронной сети довольно много локальных минимумов, число которых растет экспоненциально с ростом размерности. Поэтому существенное смещение стартовых состояний по большому счету не оказывает ощутимого влияния на эффективность минимизации.

Сказанное выше хорошо иллюстрирует рисунок 3.7, на котором представлено взаимное расположение распределений энергий минимумов в различных точках пространства. Все значения нормированы на среднее значение энергии минимумов исходного функционала. Случайные точки  $S$  распределены по нормальному закону около нуля (кривая 1). Минимумы стандартной модели Хопфилда с исходной матрицей распределены около -1 (кривая 3), что соответствует среднему значению энергии минимумов, которое можно оценить с помощью выражения (3.1). Результаты

одноэтапного алгоритма ( $q = 1$ ) отображены кривой 3. А кривая 4 показывает распределение минимумов двухэтапного алгоритма.

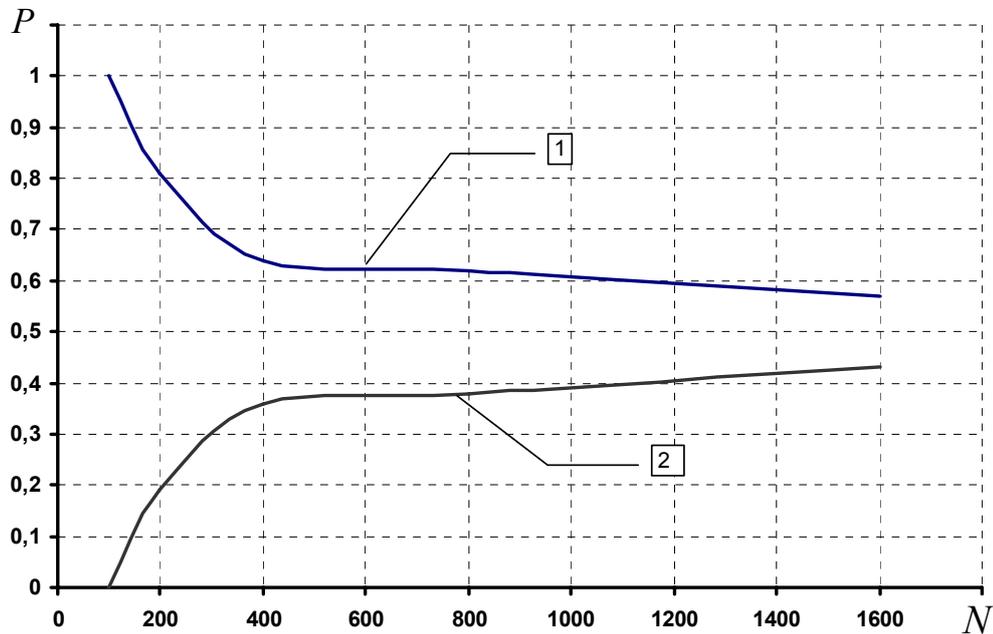
Большая часть минимизационного пути проходится одноэтапным алгоритмом, которая даже в случае малого числа градаций ( $q = 1$ ) составляет более 90% от пути обычной модели Хопфилда. При большом числе градаций ( $q = 16$ ) кривые 2 и 3 практически полностью совпадут, т.е. подавляющее число найденных минимумов  $S_0^*$  окажутся минимумами исходного функционала  $E$ , и второй этап окажется фактически ненужным. Тем не менее, второй этап позволяет с большей вероятностью находить более глубокие минимумы (кривая 4), чем обычная нейронная сеть Хопфилда. В среднем минимумы, получаемые двухэтапным алгоритмом, глубже. Вероятность того, что двухэтапный алгоритм найдет более глубокий (или равный) минимум, чем сеть Хопфилда составляет более 50% (см. рис. 3.8).



**Рис. 3.7.** Функции распределения состояний: начального  $s$  (кривая 1), дискретизированного  $s_0^*$  (кривая 2) и конечного  $s_0$  (кривая 4). Кривая 3 – распределение состояний исходного функционала.

К сожалению, использование двухэтапного алгоритма влечет за собой отказ от некоторых преимуществ одноэтапного алгоритма. Например, из-за необходимости использовать исходную матрицу на втором этапе теряется

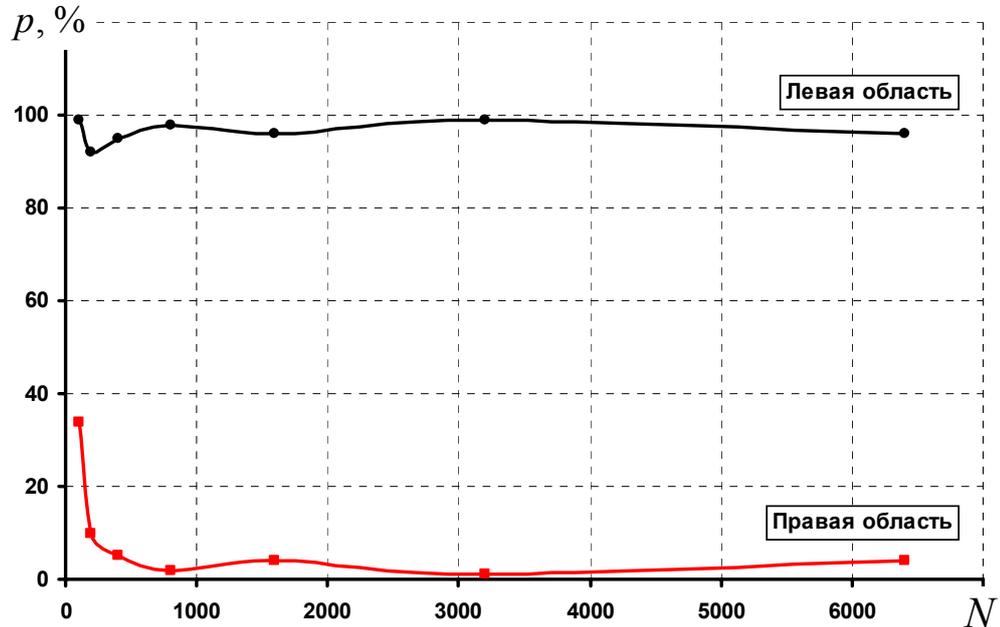
преимущество в экономии оперативной памяти. Так же снижается скорость алгоритма, поскольку второй этап может оказаться по затратам сравним с обычной моделью Хопфилда.



**Рис. 3.8.** Вероятность найти более (кривая 1) глубокий минимум или менее (кривая 2) глубокий с помощью 2-этапного алгоритма.

Однако влияние указанных выше недостатков можно снизить. К сожалению, нельзя утверждать, что старт из более глубокого состояния приведет к нахождению более глубокого минимума. Тем не менее, необязательно проводить старты из всех точек  $S_0^*$ , найденных на первом этапе. Например, если производить старты из состояний, лежащих левее среднего значения энергии минимумов  $S_0^*$ , то вероятность найти самый глубокий минимум окажется достаточно большой (рис. 3.9). Уже в случае  $q=1$  вероятность найти самый глубокий минимум превышает 60%, и, соответственно, увеличивается с ростом числа градаций (при числе градаций  $q=8$  вероятность превышает 90%). В основном число минимумов с обеих сторон от среднего значения приблизительно одинаково. Однако для размерности  $N=100$  распределение является асимметричным, т.е. слева минимумов на 8-10% больше, чем справа. Поэтому на графике наблюдается

выброс для размерности  $N=100$  (вероятность близка к 100%). Итак, при отбрасывании половины состояний можно уменьшить число операций на втором этапе вдвое.



**Рис. 3.9.** Вероятность попадания в самый глубокий минимум из левой и правой областей распределения  $s_0^*$ .

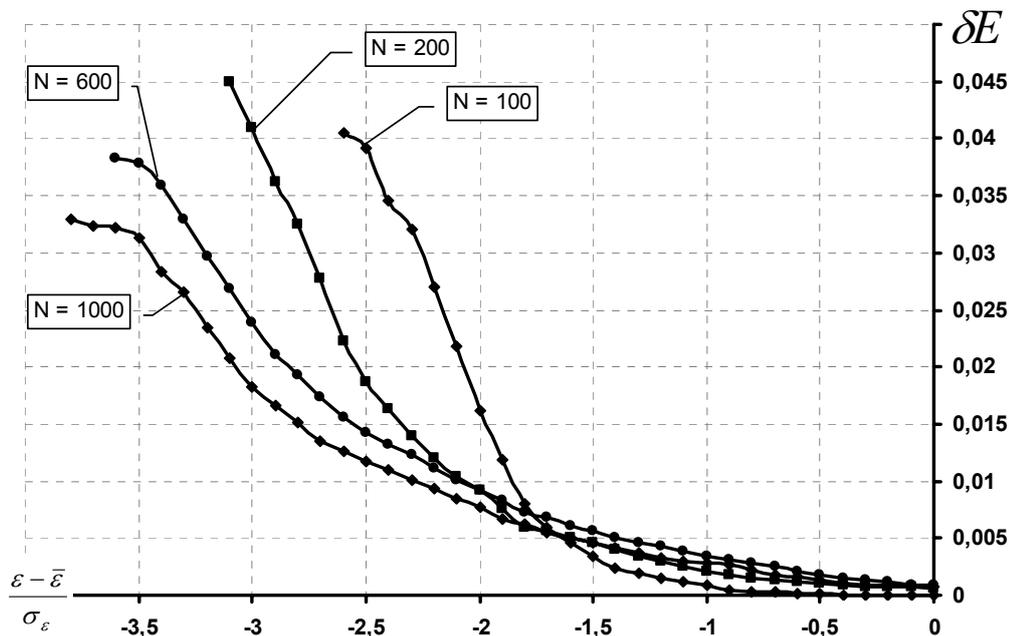
В качестве границы раздела стартовых областей необязательно выбирать среднее значение энергий  $\varepsilon$  в состояниях  $S_0^*$ , ее можно смещать влево, значительно уменьшая число стартов на втором этапе. Конечно, при таком подходе будут потери в эффективности минимизации, т.е. чем меньше стартовая область, тем хуже получаемое решение (рис. 3.10). Эффективность минимизации оценивается выражением

$$\delta E = \frac{E_0 - E^*}{E_0}, \quad (3.3)$$

где  $E_0$  - самый глубокий минимум, найденный со всех состояний  $S_0^*$ ,  $E^*$  - самый глубокий минимум, найденный при стартах из левой области.

Если уменьшить стартовую область на  $\approx 2\sigma_\varepsilon$  от среднего значения, то эффективность алгоритма упадет приблизительно на 1%. При этом число

стартов на втором этапе составит порядка 0.05 от общего числа состояний  $S_0^*$ , найденных на первом этапе.



**Рис. 3.10.** Зависимость эффективного расстояния от размера стартовой области. Размер стартовой области задается в единицах  $\sigma_\varepsilon$  функции распределения энергии  $\varepsilon(s_0^*)$ .

Двухэтапный алгоритм, несмотря на некоторые недостатки, все же позволяет находить более глубокие минимумы, чем обычная сеть Хопфилда. Правильный подбор области стартовых точек на втором этапе позволит достаточно ускорить сам алгоритм.

### §3.3. Быстродействие алгоритмов

Программная реализация стандартной модели Хопфилда требует порядка  $Op \approx 2TN^2$  операций, где  $T$  - число итераций.

Из предыдущих параграфов следует, что при минимизации дискретизированного функционала проходит более 95% минимизационного пути. При оптимизации количество итераций для дискретизированной матрицы примерно равно числу итераций исходной матрицы (рис. 3.11).

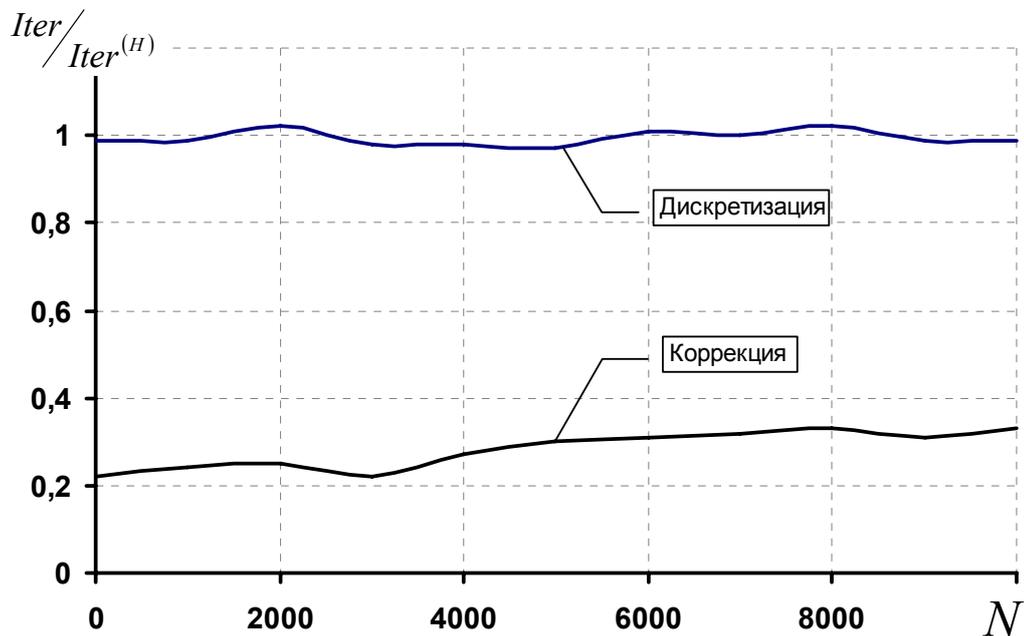
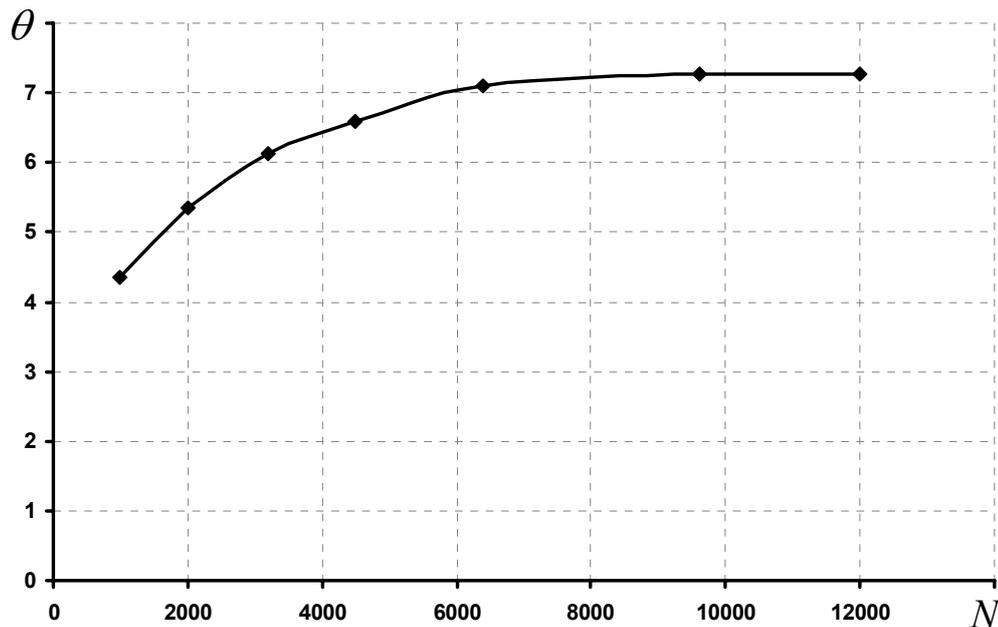


Рис. 3.11. Соотношение числа итераций при замене вещественной матрицы целочисленной ( $q = 1$ ).

Таким образом, путь, проходимый при оптимизации по энергии примерно равен пути проходимому на исходной матрице, а объем вычислений при такой замене тот же. Тем не менее, применение целых чисел малой разрядности в процедуре дискретизации дает возможность для более экономного хранения этих чисел и увеличивает скорость работы алгоритма. В случае  $q = 15$  в 4 байта можно записать одновременно 4 целых числа. При этом время работы сократится в 4 раза. А при  $q = 1$  в 4 байта можно записать сразу 8 чисел, и время работы сократится до 8 раз. Экономия происходит за

счет операций процессор-память, т.к. при использовании чисел малой разрядности можно оперировать сразу несколькими числами.

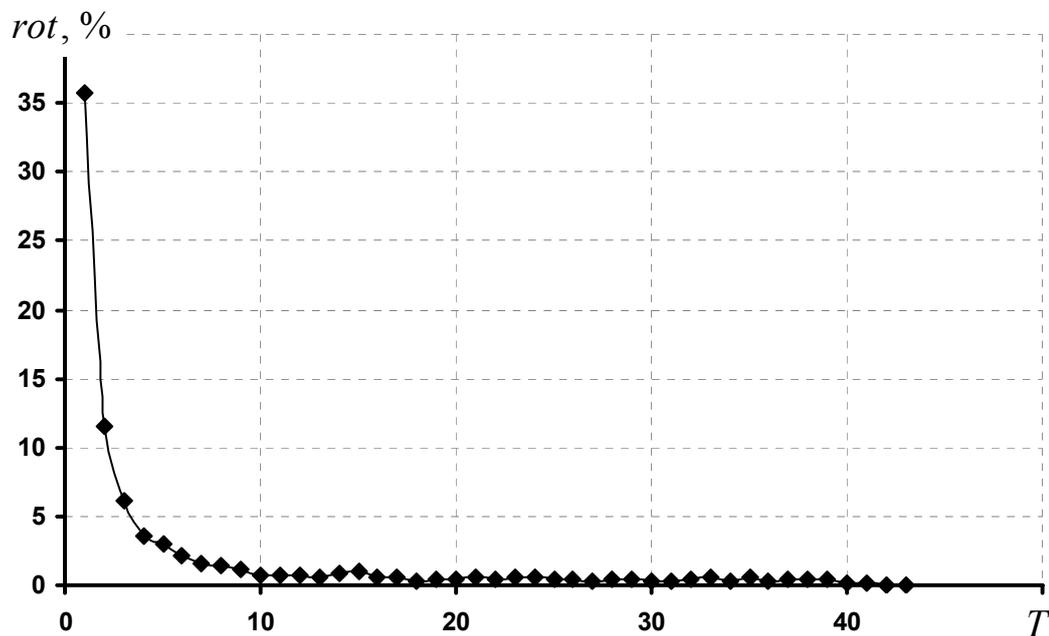
На рисунке 3.12 показано увеличение скорости алгоритма при использовании дискретизации ( $q = 1$ , в 4 байта записывалось  $p = 8$  чисел) по сравнению с алгоритмом Хопфилда. С увеличением размерности сети скорость алгоритма увеличивается и достигает своего предельного значения 7.3.



**Рис. 3.12.** Ускорение, получаемое при упаковке чисел ( $p = 8$ ) по сравнению с обычным методом.

*Модификация динамики.* С увеличением числа итераций, число нейронов, которые изменяют свое направление, неуклонно уменьшается (рис. 3.13). Это означает, что направления компонент локального поля  $\mathbf{H}$  также реже изменяется уже после 4-ой итерации (менее 5%). Поэтому вычисление на каждом шаге компоненты вектора  $\mathbf{H}$  не эффективно. В настоящее время используется другой метод расчета. В исходном состоянии  $\mathbf{S}$  вычисляются все компоненты  $\mathbf{H}$ . На каждом шаге процедуры при изменении состояния нейрона вектор  $\mathbf{H}$  модифицируется по правилу  $\mathbf{H} = \mathbf{H} \pm 2(\mathbf{A})_i$ , если

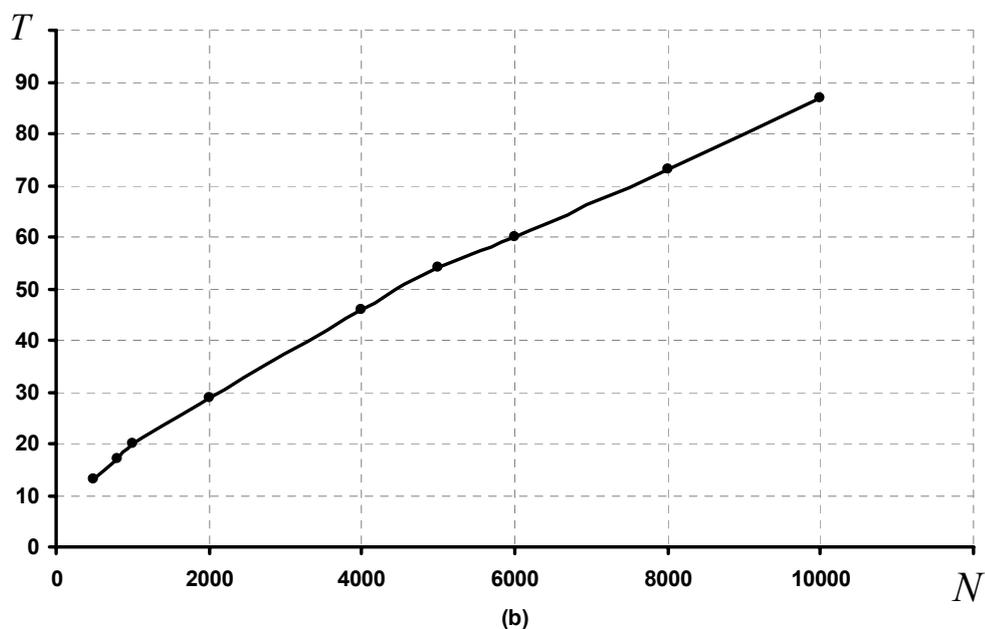
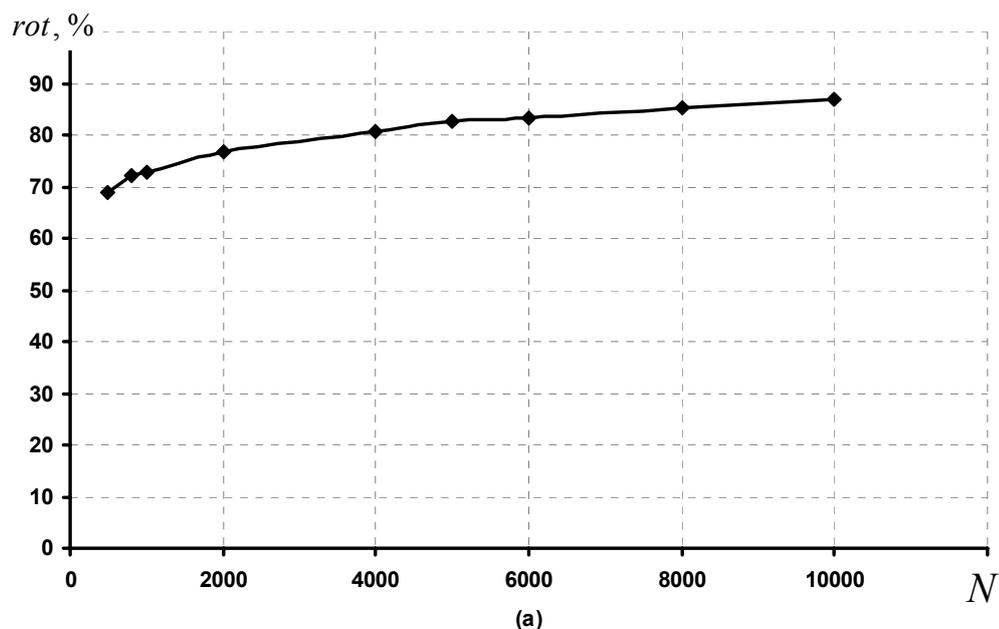
направление спина положительно/отрицательно,  $(\mathbf{A})_i$  –  $i$ -ый вектор-столбец матрицы  $\hat{\mathbf{A}}$ .



**Рис. 3.13.** Количество изменений состояний нейронов на каждой итерации. Размерность нейросети  $N = 4000$ ,  $T$  - номер итерации.

При перевороте каждого спина производится  $N$  операций. Число переворотов спинов не превышает  $N$  (рис. 3.14а), поэтому вычислительная сложность такого алгоритма  $\sim (1 + \alpha)N^2$ , а  $0.5 < \alpha < 1$ . Отношение объемов вычислений исходного метода к методу с обновлениями составляет величину порядка числа итераций  $\sim T$ , ( $T \gg 1$ ) (рис. 3.14б).

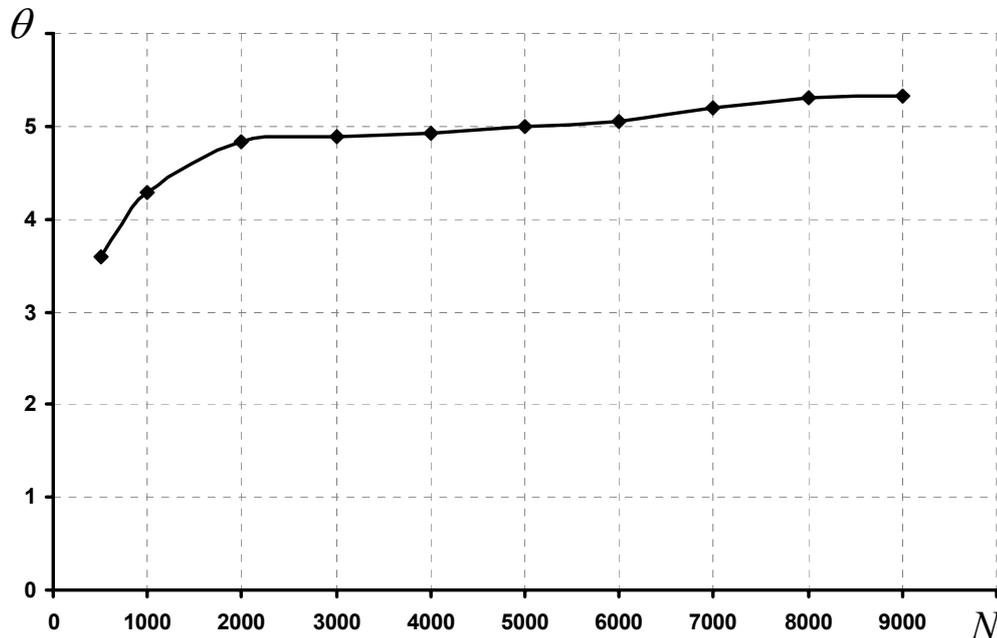
Рассмотрим далее ускорение алгоритмов (одноэтапного и двухэтапного) с учетом новой динамики с обновлениями.



**Рис. 3.14.** (а) – Полное число изменений состояний нейронов при увеличении размерности нейронной сети  $N$ ; (б) – Изменение числа итераций в зависимости от размерности сети  $N$ .

Увеличение быстродействия одноэтапного алгоритма при использовании дискретизации в динамике с обновлениями для различной размерности показано на рисунке 3.15. Установлено, что достигаемое ускорение при упаковке 8 чисел в 4 байта ( $q = 1, p = 8$ ) равно  $\theta \approx 5.3$ . При упаковке 4 чисел ( $p = 4$ ) в исходный формат ускорение составляет  $\theta \approx 3.8$

( $q < 16$ ). Таким образом, с помощью дискретизации можно добиться 8-кратного увеличения скорости работы алгоритма при использовании чисел малой разрядности  $\{-1; 0; +1\}$ .



**Рис. 3.15.** Ускорение, получаемое при упаковке чисел ( $p = 8$ ) по сравнению с обычным методом.

Объем вычислений двухэтапного алгоритма складывается из объема вычислений на первом и втором этапах. Согласно рис. 3.11 число итераций на первом этапе равно числу итераций обычного алгоритма  $O_H$ , однако, за счет дискретизации эти операции выполняются в  $\theta$  раз быстрее. Таким образом, объем вычислений на первом этапе по сравнению с алгоритмом Хопфилда составит

$$O_1 = \frac{O_H}{\theta}. \quad (3.4)$$

На втором этапе производится лишь часть пусков, произведенных на первом этапе. В среднем доля пусков составляет 0.05 от общего числа стартов. При этом на втором этапе (коррекции) производится лишь 0.3 от числа итераций стандартного алгоритма Хопфилда (рис. 3.11, нижняя кривая). Поэтому объем вычислений на втором этапе составит

$$O_2 = 0.05 \cdot 0.3 O_H. \quad (3.5)$$

Тогда ускорение двухэтапного алгоритма составит

$$\Theta = \frac{O_H}{0.05 \cdot 0.3O_H + O_H/\theta}. \quad (3.6)$$

В случае  $q = 1, p = 8 - \Theta = 5$ , а при  $q = 1, p = 16 - \Theta = 7.14$ .

В результате, применение дискретизации позволяет достичь 8-кратного ускорения без существенных потерь в эффективности минимизации. Данные значения быстродействия алгоритмов были получены для случая, когда элемент исходной матрицы занимает 4 байта памяти.

### **§3.4. Выводы**

В настоящей главе исследовались алгоритмы минимизации с применением процедуры дискретизации.

1. *Описывается одноэтапный алгоритм, позволяющий быстро минимизировать квадратичный функционал.* В одноэтапном алгоритме находятся минимумы дискретизированного функционала, из которых выбирается самый глубокий. Энергия минимумов определяется на дискретизированной матрице, так как вычисление энергии на исходной матрице является довольно долгой процедурой, сравнимой по вычислительной сложности с самим процессом спуска сети в минимум.
2. *Было показано, что энергетические поверхности дискретизированной нейросети и исходной сильно коррелируют.* Это означает, что глубина состояний дискретизированного функционала пропорциональна глубине этих же состояний на исходном функционале. Это соотношение нарушается для самых глубоких минимумов при малом числе градаций. Так для  $q = 1$  самый глубокий минимум дискретизированного функционала может оказаться на расстоянии 2-3% от самого глубоко минимума на исходном функционале. Тем не менее, при  $q = 16$  соответствие полное. Поэтому, найдя самый глубокий минимум дискретизированного функционала одноэтапным алгоритмом можно утверждать, что он окажется либо самым глубоким из найденных состояний для исходного функционала, либо окажется на расстоянии 2-3% от оптимума.
3. *С увеличением числа градаций распределение минимумов дискретизированного функционала смещается в сторону более глубоких состояний.* В среднем при  $q = 1$  дискретизированные минимумы не доходят 7% до минимумов, которые находятся обычной сетью Хопфилда. Однако с увеличением числа градаций дискретизированные минимумы все чаще совпадают с минимума

исходного функционала и при  $q = 16$  распределения дискретизированных минимумов и минимумов исходного функционала практически полностью совпадают.

4. Предложены оценочные выражения среднего значения энергии дискретизированных минимумов и ее стандартное отклонение:

$$\langle \varepsilon \rangle = -\frac{\sigma_c N^2}{4\sqrt{0.14N}}, \quad \sigma_\varepsilon = \frac{\sigma_c N \sqrt{2(1 - \langle \varepsilon \rangle^2 / N^4)}}{4}.$$

5. Одноэтапный алгоритм в 4-5 раз быстрее сети Хопфилда с исходной матрицей. При использовании дискретизации в 4-х байтную переменную, в которой хранится один элемент исходной матрицы, можно записать несколько дискретизированных элементов. Если в 4 байта записывать 4 дискретизированных числа ( $q = 16$ ), то ускорение алгоритма составит  $\sim 3.8$  раз. Если при  $q = 1$  записывать в 4-х байтную переменную сразу 8 чисел, то ускорение составит  $\sim 5.3$  раза. Это измерялось реальное время работы программы на различных размерностях матриц.
6. Упаковка нескольких чисел в одну переменную позволяет экономить оперативную память до 8-ми раз и решать задачи больших размерностей, недоступных обычной сети Хопфилда.
7. Описывается двухэтапный алгоритм, позволяющий быстро находить минимумы исходного функционала. На первом этапе находятся минимумы дискретизированного функционала. На втором этапе найденные минимумы дискретизированного функционала корректируются с помощью исходной сети Хопфилда.
8. Двухэтапный алгоритм более чем с 50% вероятностью находит более глубокий минимум, чем модель Хопфилда.
9. Предлагается модификация двухэтапного алгоритма для увеличения его быстродействия. Так как на втором этапе используется обычная сеть Хопфилда, то преимущества даваемые дискретизацией в

быстродействию и экономии памяти сильно снижаются. Поэтому на втором этапе предлагается использовать лишь несколько самых глубоких минимумов, найденных на первом этапе. Экспериментальные исследования показали, что, используя на втором этапе лишь 5% всех минимумов, можно добиться 5-кратного ускорения всего алгоритма. При этом глубина находимых минимумов (т.е. эффективность минимизации) снизится не более чем на 1%.

## ГЛАВА 4. СХЕМЫ АЛГОРИТМОВ ПОИСКА

В данной главе будут рассмотрены схемы алгоритмов поиска и модификации динамики нейронной сети Хопфилда при использовании процедуры дискретизации. Также будет произведен подробный анализ вычислительной сложности нейронной сети с дискретизированной матрицей связей и показано, какие преимущества и недостатки проявляются в данном подходе.

### *§4.1. Схема одноэтапного алгоритма*

На рисунке 4.1 представлена блок-схема одноэтапного алгоритма. Алгоритм начинается с дискретизации и упаковки матрицы связей. Число градаций при дискретизации определяется заранее исходя из требований задачи. Если необходимо добиться максимальной скорости алгоритма, то стоит выбрать минимальное число градаций ( $q=1$ ). Увеличив число градаций (до  $q=8 \div 16$ ), можно получить наиболее глубокие минимумы, при этом, конечно же, будут потери в скорости алгоритма. После того как матрица преобразована, можно делать случайные старты. Генерируется случайное состояние сети Хопфилда и с него начинается спуск сети в устойчивое состояние. В полученном минимуме определяется и запоминается значение дискретизированного функционала. Затем генерируется следующее начальное состояние. Когда все старты проведены, выбирается самое глубокое состояние. Как было показано в предыдущей главе, при малом числе градаций самое глубокое дискретизированное состояние не соответствует самому глубокому состоянию исходного функционала. Если необходимо, можно вычислять значения исходного функционала, но этот способ в  $N$ -раз медленнее.

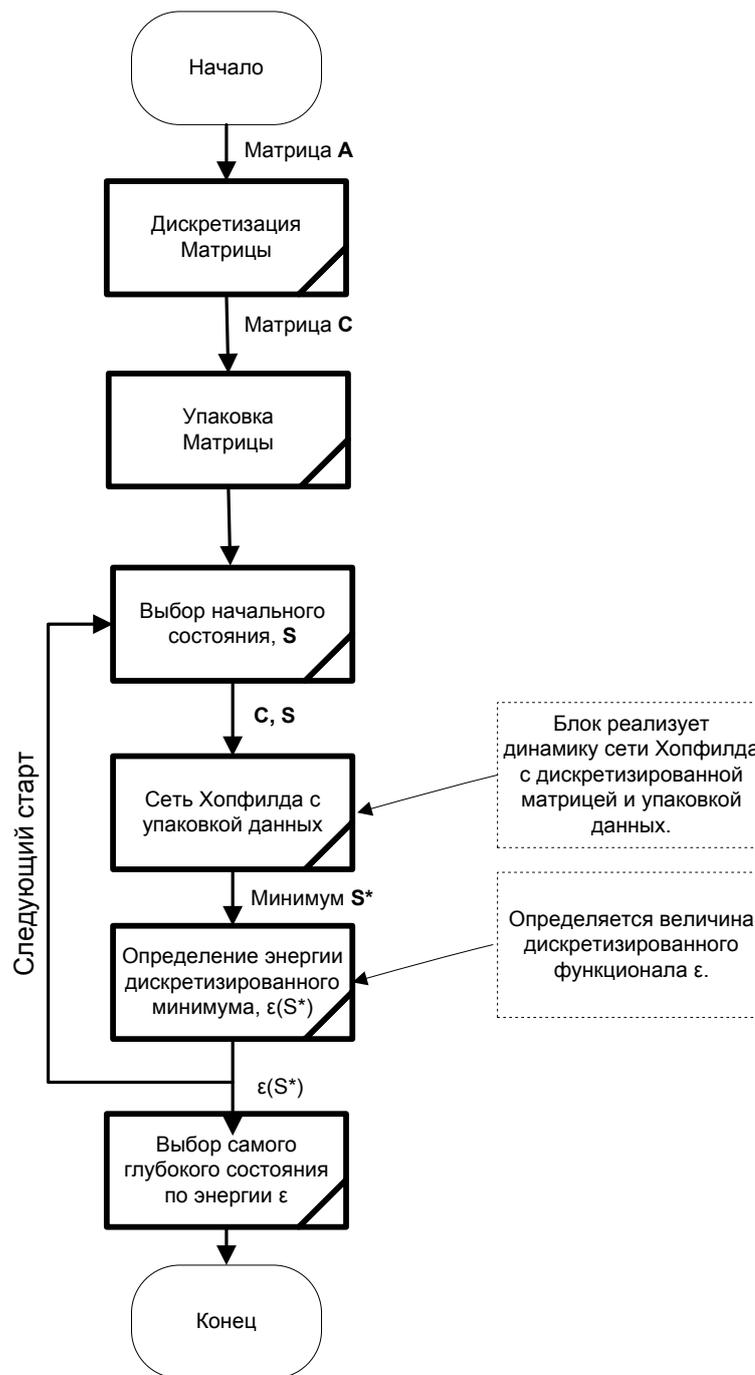


Рис. 4.1. Схема одноэтапного алгоритма.

#### **§4.2. Схема двухэтапного алгоритма**

На рисунке 4.2. представлена схема двухэтапного алгоритма. Алгоритм начинается с дискретизации и упаковки матрицы. На следующем этапе производятся старты со случайных состояний. После того, как генерируется случайное состояние, с него запускается нейронная сеть Хопфилда с дискретизированной матрицей связей. Когда сеть приходит в устойчивое состояние, определяется дискретизированная энергия полученного минимума. Если значение энергии меньше некоторого граничного значения, например,  $\varepsilon_{cp} - 2.0\sigma_\varepsilon$ , где  $\varepsilon_{cp}$  - оценочное среднее значение энергии минимумов. Если энергия выше заданной границы, то производится старт из следующего случайного состояния. В противном случае найденный дискретизированный минимум передается на следующий этап. На следующем этапе дискретизированный минимум используется как стартовое состояние для нейронной сети с исходной матрицей связей. В итоге сеть приходит в минимум исходного функционала. В итоге из полученных минимумов исходного функционала выбирается самый глубокий. В предыдущей главе было показано, что выбор верхней границы влияет на глубину минимумов, которые могут быть найдены алгоритмом, а так же на количество стартов на втором этапе, т.е. на скорость алгоритма.



### *§4.3. Нейронная сеть Хопфилда с дискретизированной матрицей*

Увеличение скорости алгоритма достигается за счет применения дискретизации в нейронной сети Хопфилда. Алгоритмически динамика сети практически не отличается от стандартной модели. После того как получено начальное состояние для сети, оно упаковывается в более плотный формат, как и матрица. Таким образом, получается новый вектор-состояние, которой в занимает в  $p$  раз меньший объем памяти ( $p$  - число чисел, которые помещаются в 4 байта). Затем вычисляются локальные поля, действующие на все нейроны. Так как все матрица и вектор-состояние находятся в упакованном состоянии, то из памяти на процессор для обработки поступают сразу несколько чисел. Например, если число градаций задано равным  $q = 1$ , то для хранения одного дискретизированного элемента достаточно выделить 4 бита. Тогда в 4 байта вместо одного вещественного числа можно записать 8 целых чисел (элементов дискретизированной матрицы). Поэтому при вычислении локальных полей из памяти сразу будет взято 8 элементов матрицы и 8 компонент вектора-состояния и одновременно перемножено попарно. Затем следующие 8 пар и т.д. Потом можно так же по 8 чисел просуммировать. Однако тут важно помнить, что при суммировании может возникнуть переполнение выделенной памяти. Поэтому после того, как к 4-х битному числу еще семь раз будут прибавлены такие же числа, необходимо результат переместить в переменную большего формата, например, 4-х байтную, в которую будет добавляться результат после каждого 8-го суммирования. Как только эти моменты учтены, алгоритм работы нейронной сети продолжает работать по стандартной схеме. Т.е. вычислив локальные поля в начальном состоянии, начинается проверка совпадения знака локального поля и нейрона. Для этого распаковывается первый элемент вектора  $X$  и все 8 компонент поочередно проверяются на совпадение знака. Если состояние нейрона совпадает с направлением локального поля, то

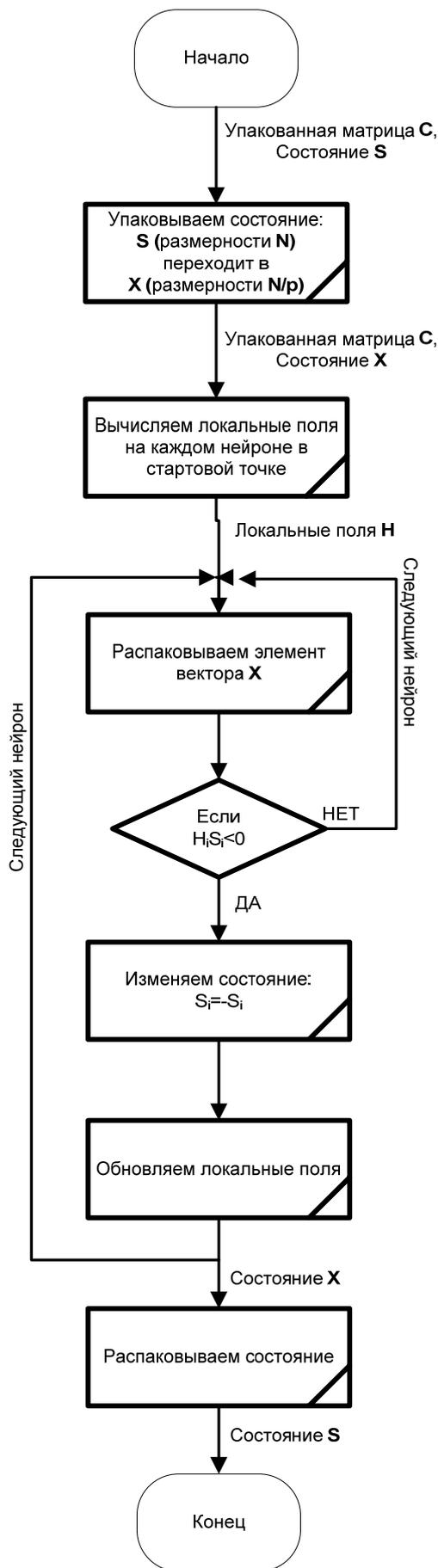


Рис. 4.3. Схема нейронной сети Хопфилда с дискретизированной матрицей.

переходим к следующему нейрону. Если же нет совпадения, то состояние нейрона меняется на противоположное, а все локальные поля обновляются. Как только все 8 нейронов проверены, они упаковываются обратно, и выбирается для обработки следующий элемент упакованного вектора  $X$ . Эта процедура продолжается пока происходят изменения в состояниях нейронов.

Проанализируем теперь получаемое при помощи дискретизации ускорение. Рассмотрим случай, когда под элемент исходной матрицы выделяется 4 байта. В зависимости от числа градаций под дискретизированный элемент можно выбирать различное число бит, желательно, с запасом, чтобы переполнение не происходило достаточно часто. Например, можно выделять  $r = 4$  бита, когда  $q \in \{1; 2\}$  и упаковывать  $p = 8$  дискретизированных чисел в 4 байта или  $r = 8$  бит, когда  $q \in [3; 16]$  и упаковывать  $p = 4$  дискретизированных числа в 4 байта. Тогда можно будет просуммировать  $v$  чисел разрядности  $r$ , прежде чем произойдет переполнение.

$$v = \left\lfloor \frac{2^{r-1}}{q} - 1 \right\rfloor. \quad (4.1)$$

Здесь  $\lfloor x \rfloor$  - наибольшее целое, меньшее или равное  $x$ .

С учетом сказанного выше можно оценить число операций, которое затратит нейронная сеть с упакованными данными на спуск из случайного состояния в минимум

$$O_p = 4NT + N \left[ \frac{2}{p} + \frac{1}{2v} \right] (N + rot). \quad (4.2)$$

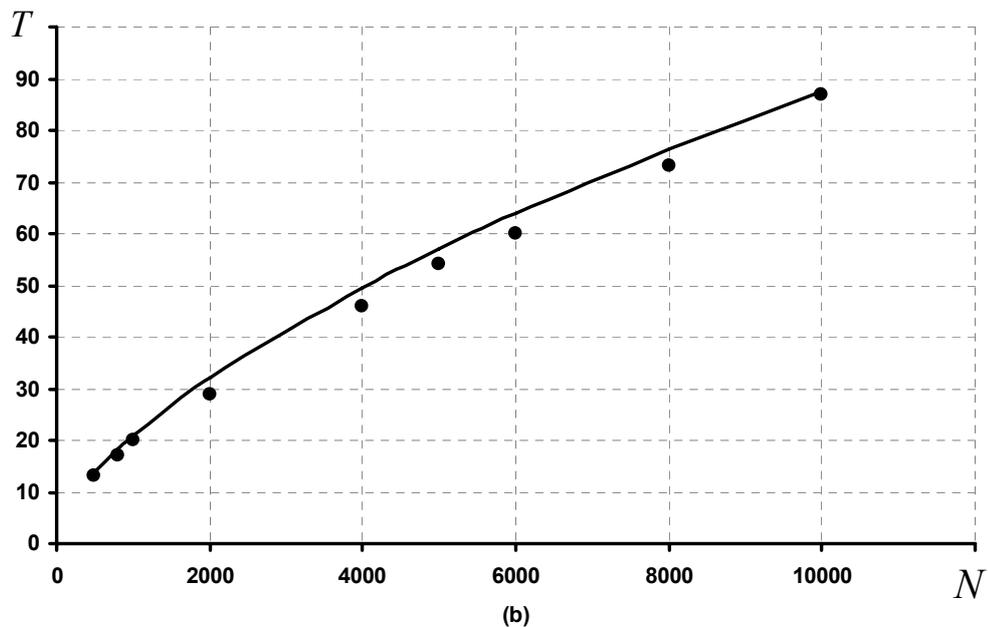
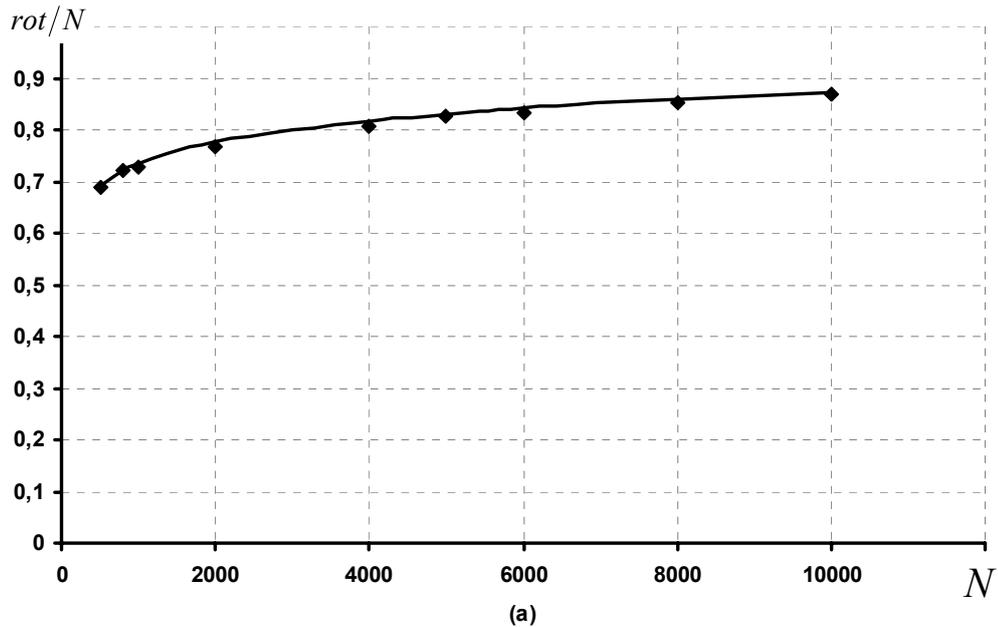
Стандартная сеть Хопфилда производит спуск из стартового состояния в минимум за число операций равное

$$O_H = 2N^2 + NT + N \cdot rot. \quad (4.3)$$

В результате многочисленных экспериментов была установлена зависимости между числом итераций  $T$ , числом изменений нейронами своего состояния  $rot$  и размерностью задачи  $N$ :

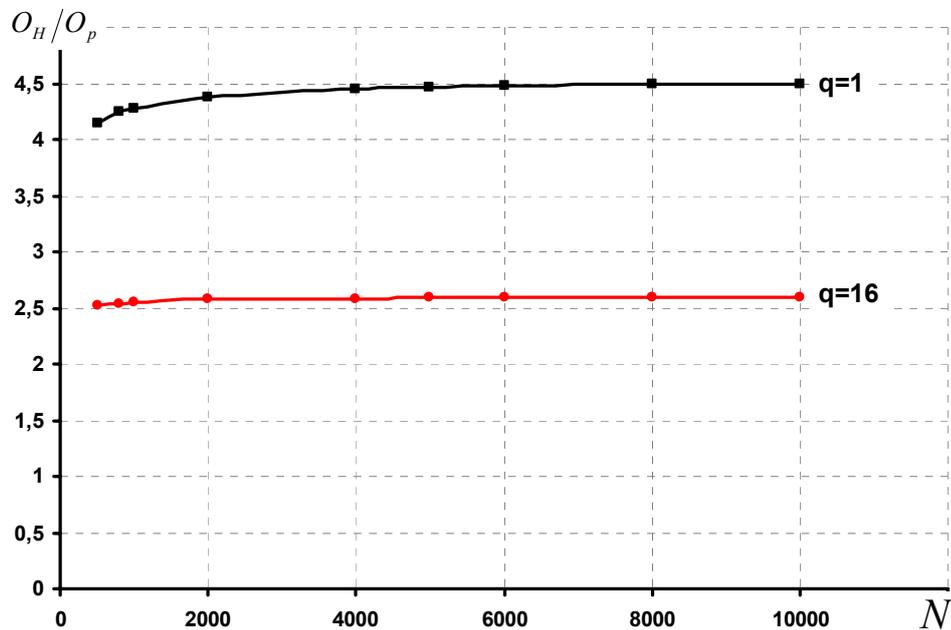
$$T \approx 0.29N^{0.62}, \quad rot \approx 0.32N(1 + 0.2\ln(N)). \quad (4.4)$$

На рисунке 4.4 представлены зависимости (4.4) и экспериментальные данные. Заметим, что число итераций значительно меньше числа изменений состояний нейронов. Поэтому при больших размерностях вклад операций, связанных с итерациями незначителен по сравнению с операциями при изменении нейронами своих состояний.



**Рис. 4.4.** (a) – число изменений состояний нейронов. (b) – число итераций. Маркеры – экспериментальные данные, сплошные кривые построены по формулам (4.4)

С учетом приведенных выше выражений можно оценить выигрыш по числу операций. На рисунке 4.5 представлено отношение числа операций алгоритма Хопфилда и алгоритма с упаковкой дискретизированных чисел. Видно, что за счет упаковки данных (матрицы и состояния) экономия операций составляет около 4.5 раз. С увеличением числа градаций скорость снизится, так как в 4-х байтную переменную удастся упаковать меньше дискретизированных элементов (например,  $q=16$  и  $p=4$ ). Восьмикратное ускорение не достигается в связи с тем, что при работе с упакованными данными периодически приходится их в переменные большего формата.



**Рис. 4.5.** Отношение числа операций стандартной модели Хопфилда к алгоритму с упаковкой в 4 байта для разного числа градаций.

Чтобы получить полное ускорение алгоритма необходимо также учесть время, затрачиваемое на передачу данных между памятью и процессором. Так как при упаковке передаются сразу несколько чисел, то число обращений процессора к памяти будет примерно в  $p$  раз меньше, чем при обычном алгоритме. Общее время работы алгоритмов можно оценить как

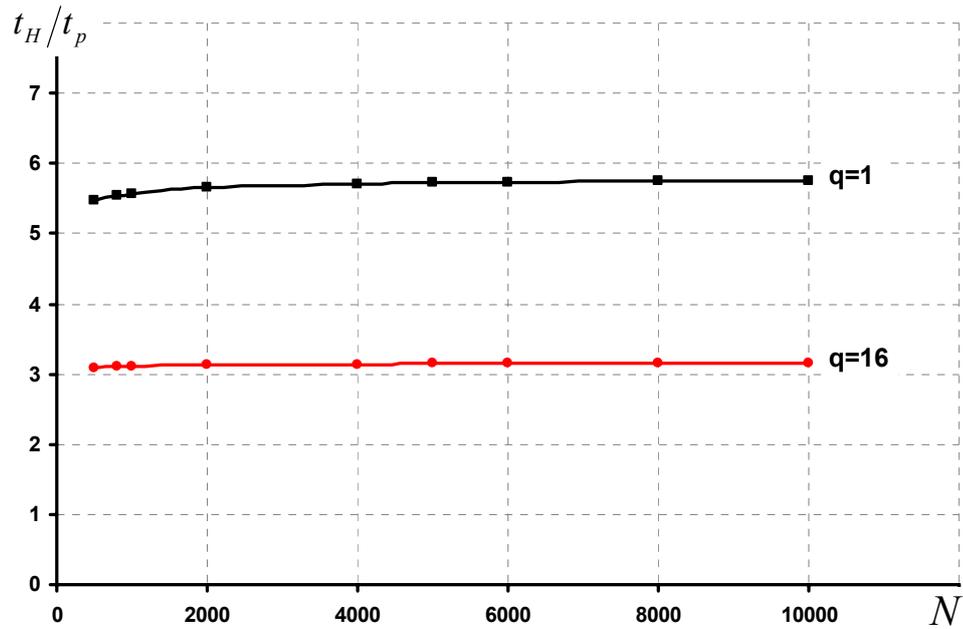
$$\begin{aligned}
 t_H &= O_H \cdot t_{ар} + t_{пер} \cdot O_H \\
 t_p &= O_p \cdot t_{ар} + t_{пер} \cdot O_H / p
 \end{aligned}
 \tag{4.5}$$

Здесь  $t_{ар}$  - время выполнения арифметической операции,  $t_{пер}$  - время передачи данных между процессором и памятью. Время передачи данных обычно много больше времени выполнения арифметических операций. Можно рассмотреть 2 граничных случая.

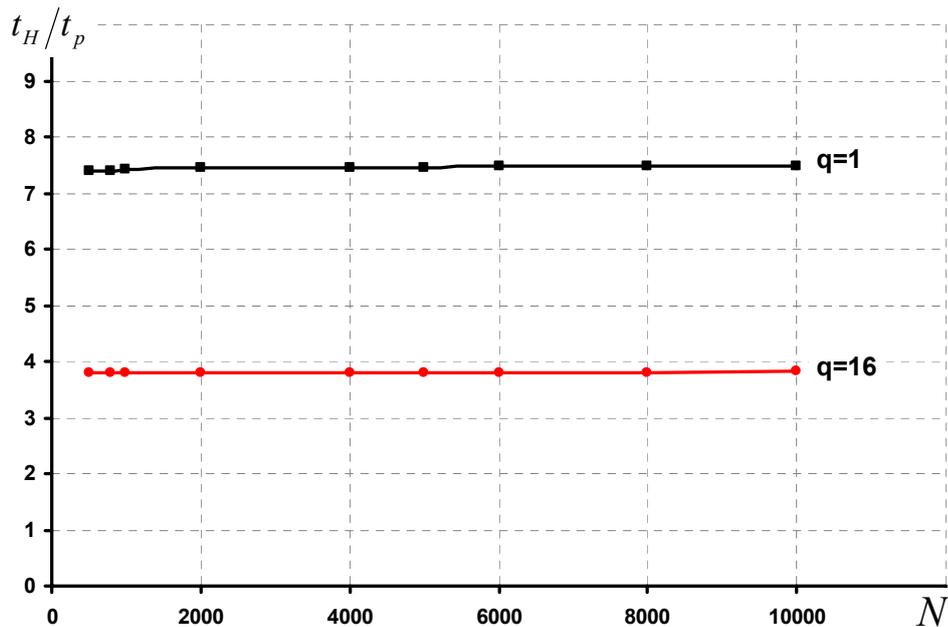
1.  $t_{пер} = t_{ар}$  (см. рис. 4.6): в этом случае, если вычислить отношение времен работы алгоритмов получим, что при  $q = 1$  алгоритм с упаковкой данных работает приблизительно в 5 раз быстрее, чем сеть Хопфилда. При  $q = 16$  ускорение составит более 3 раз.

2.  $t_{пер} \gg t_{ар}$  (см. рис. 4.7): например,  $t_{пер} = 10t_{ар}$ . В этом случае получим, что при  $q = 1$  алгоритм с упаковкой данных работает почти в 7.5 раз быстрее, чем сеть Хопфилда. А при  $q = 16$  ускорение приближается к 4 раз.

Таким образом, ускорение алгоритма с упаковкой данных в зависимости от архитектуры вычислительной машины позволяет достичь ускорения от 5.5 до 7.5 раз.



**Рис. 4.6.** Полное ускорение алгоритма с упаковкой по сравнению с сетью Хопфилда, когда время выполнения арифметических операций равно времени передачи данных между процессором и памятью.



**Рис. 4.7.** Полное ускорение алгоритма с упаковкой по сравнению с сетью Хопфилда, когда время выполнения арифметических операций в 10 раз меньше времени передачи данных между процессором и памятью.

#### §4.4. Описание программной реализации

На рисунке 4.6 показан алгоритм нейронной сети Хопфилда с обновлением локальных полей. В первых трех строках производится вычисление локальных полей, действующих на все нейроны в случайной (стартовой) точке. Строки 5 – 14 повторяются до тех пор, пока происходят изменения в состояниях нейронов. В процессе динамики проверяется каждый нейрон (строки 7 – 13). Для каждого нейрона проверяется совпадение его значения со знаком действующего на него локального поля (строка 8). Если знаки не совпадают, то нейрон меняет свое состояние на противоположное (строка 9), а локальные поля обновляются (строка 10).

```
1: for  $i = 1$  to  $N$  do begin  
2:    $H_i = \sum_{j=1}^N A_{ij} s_j$   
3: endfor  
4:  $isChanged = \mathbf{true}$   
5: while  $isChanged$  do begin  
6:    $isChanged = \mathbf{false}$   
7:   for  $i = 1$  to  $N$  do begin  
8:     if  $(H_i s_i < 0)$  then begin  
9:        $s_i = -s_i$   
10:      for  $i = 1$  to  $N$  do  $H_i = H_i + 2A_{ij} s_i$   
11:       $isChanged = \mathbf{true}$   
12:    endif  
13:  endfor  
14: endwhile
```

Рис. 4.6. Алгоритм нейронной сети Хопфилда.

Использование дискретизированной матрицы потребует некоторой модификации исходного алгоритма нейронной сети. В данном параграфе будут приведены примеры основных функций для случая упаковки 8 чисел (-1, 0, +1) в 4 байта.

```

1: Update(tmp, k)
2: begin
3:   if (tmp > 0) then begin
4:     for i = 1 to N/8 do  $h_i^+ = h_i^+ + 2\tilde{C}_{ik}$ 
5:      $m_1 = m_1 + 1$ 
6:     if ( $m_1 \bmod 8$ ) then Load(h,  $h^+$ )
7:   endif
8:   if (tmp < 0) then begin
9:     for i = 1 to N/8 do  $h_i^- = h_i^- + 2\tilde{C}_{ik}$ 
10:     $m_2 = m_2 + 1$ 
11:    if ( $m_2 \bmod 8$ ) then Load(h,  $h^-$ )
12:  endif
13: end

```

**Рис. 4.7.** Алгоритм процедуры обновления локальных полей.

На рисунке 4.7 показана процедура обновления локальных полей при изменении нейронном своего состояния. В процедуру передается значение текущего нейрона и его номер. Переменная *tmp* принимает значения  $\pm 1$ . Используется два массива  $h^+$  и  $h^-$ , в которые в зависимости от знака нейрона добавляется соответствующий элемент матрицы связей при обновлении локальных полей. В каждый элемент массивов  $h^+$  и  $h^-$  упакованы по 8 значений локальных полей, поэтому при обновлении добавляется сразу 8 чисел. Таким образом, процедура обновления локальных полей будет выполняться в 8 раз быстрее, чем обычная. Однако, так как числа итак упакованы, то на каждый элемент выделяется только 4 бита, что ограничивает размер числа, помещенного в них. Т.е. каждые 8 сложений необходимо выгружать сумматоры в переменные большего формата, чтобы не происходило переполнения. Это условие учитывают строки 6 и 11. Процедура **Load()** производит необходимую перегрузку. Пример такой процедуры приведен на рисунке 4.8.

```

1: Load( $h, h^+$ )
2:    $A, B$  – дополнительные переменные
3: begin
4:   const  $C = 15$ 
5:    $k = 0$ 
6:   for  $i = 1$  to  $N/8$  do begin
7:      $B = h_i^+; h_i^+ = 0$ 
8:      $A = C$  and  $B; h_{k+1} = h_{k+1} + A$ 
9:      $B = B$  shR 4;  $A = C$  and  $B; h_{k+2} = h_{k+2} + A$ 
10:     $B = B$  shR 4;  $A = C$  and  $B; h_{k+3} = h_{k+3} + A$ 
11:     $B = B$  shR 4;  $A = C$  and  $B; h_{k+4} = h_{k+4} + A$ 
12:     $B = B$  shR 4;  $A = C$  and  $B; h_{k+5} = h_{k+5} + A$ 
13:     $B = B$  shR 4;  $A = C$  and  $B; h_{k+6} = h_{k+6} + A$ 
14:     $B = B$  shR 4;  $A = C$  and  $B; h_{k+7} = h_{k+7} + A$ 
15:     $B = B$  shR 4;  $A = C$  and  $B; h_{k+8} = h_{k+8} + A$ 
16:     $k = k + 8$ 
17:   endfor
18: end

```

**Рис. 4.8.** Алгоритм выгрузки чисел в переменные большего формата.

В процедуре используются битовая операция сдвига вправо **shR** и логическое умножение **and**, которые выполняются много быстрее обычных арифметических операций.

Так же необходимы функции **Pack()** и **unPack()** – упаковки и распаковки, соответственно. Их алгоритмы приведены на рисунке 4.9. Эти процедуры выполняются достаточно быстро за счет использования в них только операций битового сдвига и логических умножения и сложения.

*Упаковывается 8 чисел массива B в переменную A*

```
1: Pack(A,B)
2: begin
3:   A = 0
4:   A = A shL 4; A = A or B8
5:   A = A shL 4; A = A or B7
6:   A = A shL 4; A = A or B6
7:   A = A shL 4; A = A or B5
8:   A = A shL 4; A = A or B4
9:   A = A shL 4; A = A or B3
10:  A = A shL 4; A = A or B2
11:  A = A shL 4; A = A or B1
12: end
```

*Распаковываются 8 чисел из переменной A в массив B*

```
1: unPack(B,A)
2: begin
3:   const C = 15
4:   B1 = C and A
5:   A = A shR 4; B2 = C and A
6:   A = A shR 4; B3 = C and A
7:   A = A shR 4; B4 = C and A
8:   A = A shR 4; B5 = C and A
9:   A = A shR 4; B6 = C and A
10:  A = A shR 4; B7 = C and A
11:  A = A shR 4; B8 = C and A
12: end
```

**Рис. 4.9.** Процедуры упаковки и распаковки 8 чисел в массив и обратно.

При упаковке состояние **S** упаковывается в массив **X**, размерность которого в 8 раз меньше, а элементы принимают значения 0 и +1. Так же дискретизированная матрица  $\hat{C}$  упаковывается в матрицу  $\tilde{C}$ . На рисунке 4.10 показан алгоритм сети Хопфилда с упакованными данными.

```

1: for  $i = 1$  to  $N/8$  do begin
2:   unPack( $u, x_i$ )
3:   for  $j = 1$  to  $8$  do begin
4:      $tmp = 2u_j - 1$ 
5:     Update( $tmp, k$ );  $k = k + N/8$ 
6:   end for
7: end for
8: Load( $h, h^+$ ); Load( $h, h^-$ )
9:  $isChanged = \text{true}$ 
10: while  $isChanged$  do begin
11:    $isChanged = \text{false}$ ;  $k = 0$ ;  $m = 0$ 
12:   for  $i = 1$  to  $N/8$  do begin
13:     unPack( $u, x_i$ ); unPack( $uh^+, h^+$ ); unPack( $uh^-, h^-$ )
14:     for  $j = 1$  to  $8$  do begin
15:        $tmp = 2u_j - 1$ 
16:        $h_{m+j} = h_{m+j} + uh_j^+ - uh_j^-$ 
17:       if ( $h_{m+j} \cdot tmp < 0$ ) then begin
18:          $tmp = -tmp$ ;  $u_j = 1 - u_j$ 
19:         Update( $tmp, k$ )
20:          $k = k + N/8$ ;  $isChanged = \text{true}$ 
21:         unPack( $uh^+, h^+$ ); unPack( $uh^-, h^-$ )
22:       endif
23:     end for
24:      $m = m + 8$ 
25:     Pack( $x_i, u$ )
26:   end for
27: end while

```

**Рис. 4.10.** Алгоритм сети Хопфилда с использованием упакованных данных.

В строках 1 – 7 определяются локальные поля в стартовой точке. Каждый дискретизированный элемент распаковывается (строка 2) и 8 элементов проверяются последовательно. В строках 10 – 27 происходит сам процесс спуска сети в минимум. Выбирается дискретизированный элемент, распаковывается состояние, а так же локальные поля. Эти 8 распакованных элементов проверяются на совпадение знаков с локальными полями и при необходимости обновляются поля (строки 14 – 23).

Может показаться, что число операций резко возрастет по сравнению с исходным алгоритмом без упаковки данных. Однако стоит учитывать, что сложение производится сразу по несколько чисел. Так же из памяти в процессор передаются блоки по 8 чисел, а не по одному. Число операций оценено в предыдущем параграфе.

#### **§4.5. Выводы**

В настоящей главе были получены следующие результаты:

1. Рассмотрены блок-схемы одноэтапного и двухэтапного алгоритмов.
2. Рассмотрена нейронная сеть Хопфилда с дискретизированной матрицей. Проведен анализ вычислительной сложности дискретизированной модели Хопфилда. Показано, что использование дискретизации позволяет уменьшить число операций от 5.5 до 7.5 раз по сравнению с сетью Хопфилда.
3. Предложены выражения для оценки числа итераций и числа изменений нейронами своего состояния в процессе работы сети Хопфилда. Так как дискретизация практически не затрагивает алгоритмическую структуру модели Хопфилда, то эти выражений применимы и для нейронной сети с дискретизированной матрицей связи.

$$T \approx 0.29N^{0.62}, \quad rot \approx 0.32N(1 + 0.2Ln(N)).$$

4. Приведены программные функции, реализующие алгоритм Хопфилда с дискретизированной матрицей.

## ЗАКЛЮЧЕНИЕ

В диссертационной работе получены следующие результаты.

1. Предложена и исследована процедура дискретизации матрицы связей, которая позволяет ускорить процесс минимизации квадратичного функционала. Удалось выделить два настроечных параметра: число градаций и размер нулевого отрезка, которые позволяют определить качество дискретизации и эффективность ее использования еще до начала проведения каких-либо экспериментов.
2. Определены оптимальные параметры дискретизации, при которых ошибка в направлениях локальных полей составляет менее 10% в любой точке конфигурационного пространства. Показано, что расстояние по Хеммингу и по энергии между минимумом дискретизированного функционала и минимумом исходного функционала, в который пришла бы исходная сеть из дискретизированного минимума, прямо пропорционально вероятности несовпадения направлений локальных полей в случайной точке. Хеммингово расстояние менее 12%, а по энергии меньше 7% для любого числа градаций.
3. На основе процедуры дискретизации разработаны алгоритмы минимизации квадратичного функционала: *одноэтапный* и *двухэтапный*. Исследовано быстроедействие предложенных алгоритмов, а также их эффективность по сравнению со стандартной моделью Хопфилда.
4. Применение одноэтапного алгоритма при использовании упакованных чисел позволяет достичь 7-кратного увеличения скорости. При этом глубина полученных минимумов по энергии на 7% меньше, чем при стандартной модели Хопфилда. При этом требуется в 8 раз меньше оперативной памяти, чем при стандартной модели Хопфилда. Таким образом, с помощью одноэтапного алгоритма можно решать задачи недоступные стандартным алгоритмам.

5. Для отыскания более глубоких минимумов предлагается двухэтапный алгоритм. Он позволяет отыскивать те же минимумы, что и стандартная модель, но с вероятностью более 50% находятся более глубокие. Использование только самых глубоких дискретизированных минимумов на втором этапе позволило сохранить скорость алгоритма, достигаемую за счет дискретизации.

## Приложение

### МАТРИЦЫ С РАВНОМЕРНЫМ РАСПРЕДЕЛЕНИЕМ ЭЛЕМЕНТОВ

СРЕДНЕЕ ЗНАЧЕНИЕ МАТРИЧНЫХ ЭЛЕМЕНТОВ:  $A_0 = 0$

ПОРОГИ НЕЙРОНОВ:  $B_i = 0$

### Вероятность несовпадения направлений локальных полей в случайной точке пространства

А) Исходная матрица  $\hat{\mathbf{A}}$ :

Равномерное распределение  $A_{ij} \in [-1;1]$ ;

$$\overline{A_{ij}} = 0; \sigma_A^2 = 1/3.$$

Б) Дискретизированная матрица  $\hat{\mathbf{C}}$ :

$$C_{ij} \in [-q; q];$$

$$C_{ij} = k, \text{ если } (2k-1)\mu \leq A_{ij} \leq (2k+1)\mu,$$

$$k \in [-q; q]; \mu = 1/(2q+1);$$

$$\overline{C_{ij}} = \mu \sum_{k=-q}^q k = 0;$$

$$\sigma_C^2 = \overline{C_{ij}^2} - \overline{C_{ij}}^2 = \overline{C_{ij}^2} = \mu \sum_{k=-q}^q k^2 = 2\mu \sum_{k=1}^q k^2 = 2\mu \frac{q(q+1)(2q+1)}{6} = \frac{2q(q+1)}{3} = \frac{1}{4\mu^2} \sigma_A^2 (1 - \mu^2).$$

В) Вектор-состояние  $\mathbf{S}$ :

$$s_i = \begin{cases} +1, & 1/2; \\ -1, & 1/2; \end{cases}$$

$$\overline{s_i} = 0; \sigma_s^2 = 1.$$

*Вектор-состояние случайной точки  $\mathbf{S}$  не коррелирует с матрицами  $\hat{\mathbf{A}}$  и  $\hat{\mathbf{C}}$ !*

Г) Локальное поле на исходной матрице  $\hat{\mathbf{A}}$ :

$$H_i = \sum_{j \neq i}^N A_{ij} s_j;$$

$$\overline{H} = \sum_{j \neq i}^N \overline{A_{ij} s_j} = \sum_{j \neq i}^N \overline{A_{ij}} \overline{s_j} = 0;$$

$$\sigma_H^2 = \sum_{j \neq i}^N \sigma^2(A_{ij} s_j) = \sum_{j \neq i}^N \sigma_A^2 \cdot \sigma_s^2 = N \sigma_A^2.$$

Д) Локальное поле на дискретизированной матрице  $\hat{\mathbf{C}}$ :

$$h_i = \sum_{j \neq i}^N C_{ij} s_j;$$

$$\bar{h} = \sum_{j \neq i}^N \overline{C_{ij} s_j} = \sum_{j \neq i}^N \overline{C_{ij}} \overline{s_j} = 0;$$

$$\sigma_h^2 = \sum_{j \neq i}^N \sigma^2(C_{ij} s_j) = \sum_{j \neq i}^N \sigma_C^2 \cdot \sigma_s^2 = N \sigma_C^2 = \frac{1}{4\mu^2} N \sigma_A^2 (1 - \mu^2).$$

Е) Среднее от произведения локальных полей  $\overline{H_i h_i}$ :

$$\overline{H_i h_i} = \overline{\left( \sum_{j \neq i}^N A_{ij} s_j \right) \cdot \left( \sum_{k \neq i}^N C_{ik} s_k \right)} = \sum_{j \neq i}^N \overline{A_{ij} C_{ij}} + \sum_{\substack{j,k \\ j \neq k \neq i}}^N \overline{A_{ij} C_{ij} s_j s_k} = \sum_{j \neq i}^N \overline{A_{ij} C_{ij}} + \sum_{\substack{j,k \\ j \neq k \neq i}}^N \overline{A_{ij} C_{ij} s_j s_k} = N \overline{A_{ij} C_{ij}};$$

$$\overline{A_{ij} C_{ij}} = \sum_{k=-q}^q \left( k \left[ \int_{(2k-1)\mu}^{(2k+1)\mu} \frac{1}{2} x dx \right] \right) = 4\mu^2 \sum_{k=1}^q k^2 = \frac{1}{2\mu} \sigma_A^2 (1 - \mu^2);$$

$$\overline{H_i h_i} = \frac{1}{2\mu} N \sigma_A^2 (1 - \mu^2).$$

Ж) Корреляция локальных полей  $\rho$ :

$$\rho = \frac{\overline{H_i h_i} - \overline{H_i} \cdot \overline{h_i}}{\sigma_H \cdot \sigma_h} = \frac{\frac{1}{2\mu} N \sigma_A^2 (1 - \mu^2)}{\sqrt{N \sigma_A^2 \cdot \frac{1}{4\mu^2} N \sigma_A^2 (1 - \mu^2)}} = \sqrt{1 - \mu^2}.$$

З) Вероятность несовпадения полей  $\Pr\{H_i h_i < 0\}$ :

$$P = 1 - \Pr\{H_i h_i > 0\} = 1 - P\{H_i > 0 \cap h_i > 0\} + P\{H_i < 0 \cap h_i < 0\};$$

$$P\{H_i > 0 \cap h_i > 0\} =$$

$$= \frac{1}{2\pi\sqrt{1-\rho^2}} \int_0^\infty \int_0^\infty \text{Exp} \left\{ -\frac{1}{2(1-\rho^2)} \left[ \left( \frac{H_i - \bar{H}}{\sigma_H} \right)^2 - 2\rho \left( \frac{H_i - \bar{H}}{\sigma_H} \right) \left( \frac{h_i - \bar{h}}{\sigma_h} \right) + \left( \frac{h_i - \bar{h}}{\sigma_h} \right)^2 \right] \right\} \frac{dH}{\sigma_H} \frac{dh}{\sigma_h} =$$

$$= \frac{1}{2\pi\sqrt{1-\rho^2}} \int_0^\infty \int_0^\infty \text{Exp} \left\{ -\frac{1}{2(1-\rho^2)} \left[ \left( \frac{H_i}{\sigma_H} \right)^2 - 2\rho \left( \frac{H_i}{\sigma_H} \right) \left( \frac{h_i}{\sigma_h} \right) + \left( \frac{h_i}{\sigma_h} \right)^2 \right] \right\} \frac{dH}{\sigma_H} \frac{dh}{\sigma_h};$$

$$P\{H_i < 0 \cap h_i < 0\} =$$

$$= \frac{1}{2\pi\sqrt{1-\rho^2}} \int_0^\infty \int_0^\infty \text{Exp} \left\{ -\frac{1}{2(1-\rho^2)} \left[ \left( \frac{H_i - \bar{H}}{\sigma_H} \right)^2 - 2\rho \left( \frac{H_i - \bar{H}}{\sigma_H} \right) \left( \frac{h_i - \bar{h}}{\sigma_h} \right) + \left( \frac{h_i - \bar{h}}{\sigma_h} \right)^2 \right] \right\} \frac{dH}{\sigma_H} \frac{dh}{\sigma_h} =$$

$$= \frac{1}{2\pi\sqrt{1-\rho^2}} \int_0^\infty \int_0^\infty \text{Exp} \left\{ -\frac{1}{2(1-\rho^2)} \left[ \left( \frac{H_i}{\sigma_H} \right)^2 - 2\rho \left( \frac{H_i}{\sigma_H} \right) \left( \frac{h_i}{\sigma_h} \right) + \left( \frac{h_i}{\sigma_h} \right)^2 \right] \right\} \frac{dH}{\sigma_H} \frac{dh}{\sigma_h};$$

$$\Pr\{H_i h_i > 0\} = \frac{1}{\pi\sqrt{1-\rho^2}} \int_0^\infty \int_0^\infty \text{Exp} \left\{ -\frac{1}{2(1-\rho^2)} \left[ \left( \frac{H_i}{\sigma_H} \right)^2 - 2\rho \left( \frac{H_i}{\sigma_H} \right) \left( \frac{h_i}{\sigma_h} \right) + \left( \frac{h_i}{\sigma_h} \right)^2 \right] \right\} \frac{dH}{\sigma_H} \frac{dh}{\sigma_h};$$

$$x = \frac{H_i}{\sigma_H}; \quad y = \frac{h_i}{\sigma_h};$$

$$\Pr\{H_i h_i > 0\} = 1 - \frac{1}{\pi\sqrt{1-\rho^2}} \int_0^\infty \int_0^\infty \text{Exp}\left\{-\frac{1}{2(1-\rho^2)}[x^2 - 2\rho xy + y^2]\right\} dx dy.$$

$$\Pr\{H_i h_i > 0\} = \frac{1}{2} - \frac{1}{\pi} \arcsin \rho \approx \frac{1}{\pi} \sqrt{2(1-\rho)}.$$

**Вероятность несовпадения направлений локальных полей  
в минимуме исходного функционала**

А) Минимум исходного функционала  $\mathbf{S}_0$ ;

$$s_{0i} = \begin{cases} +1, & 1/2; \\ -1, & 1/2; \end{cases}$$

$$\overline{s_{0i}} = 0; \quad \sigma_{s_0}^2 = 1.$$

Условие того, что  $\mathbf{S}_0$  минимум:

$$H_i s_{0i} > 0, \quad \forall i.$$

Б) Выделим минимум из матрицы  $\hat{\mathbf{A}}$ :

$$A_{ij} = A_{0ij} + A'_{ij} = r_0 \sigma_A s_{0i} s_{0j} + A'_{ij}; \quad \sum_{i=0}^{\infty} r_i^2 = 1$$

$$A_{0ij} = \frac{r_0 \sigma_A}{N^2} \sum_{\substack{i,j \\ i \neq j}}^N s_{0i} s_{0j} = \frac{r_0 \sigma_A}{N^2} \left[ \sum_{i,j}^N s_{0i} s_{0j} - \sum_i (s_{0i})^2 \right] = \frac{r_0 \sigma_A}{N^2} \left[ \sum_i s_{0i} \sum_j s_{0j} - N \right] =$$

$$= \frac{r_0 \sigma_A}{N^2} \left[ \left( \sum_i s_{0i} \right)^2 - N \right] = r_0 \sigma_A \delta;$$

$$\bar{\delta} = \frac{1}{N^2} \left[ \left( \sum_i s_{0i} \right)^2 - N \right] = \frac{1}{N^2} \cdot (-N) = -\frac{1}{N} \approx 0;$$

$$\overline{A_{0ij}} = \frac{r_0 \sigma_A}{N^2} \bar{\delta} = 0;$$

$$\overline{A_{0ij}^2} = \frac{r_0^2 \sigma_A^2}{N^2} \sum_{\substack{i,j \\ i \neq j}}^N s_{0i}^2 s_{0j}^2 = r_0^2 \sigma_A^2;$$

$$\sigma_{A_0}^2 = \overline{A_0^2} - \overline{A_0}^2 = r_0^2 \sigma_A^2 - r_0^2 \sigma_A^2 \delta^2 = r_0^2 \sigma_A^2 (1 - \delta^2);$$

$$\overline{A'_{ij}} = \overline{A_{ij}} - \overline{A_{0ij}} = \overline{A_{ij}} - r_0 \sigma_A \delta;$$

$$\sigma_{A'}^2 = \sigma_A^2 - \sigma_{A_0}^2 = \sigma_A^2 - r_0^2 \sigma_A^2 (1 - \delta^2) = \sigma_A^2 (1 - r_0^2);$$

$$\overline{A'A_0} = \overline{(A - A_0)A_0} = \overline{AA_0} - \overline{A_0^2} = r_0 \sigma_A \sum_{\substack{i,j \\ i \neq j}}^N A_{ij} s_{0i} s_{0j} - r_0^2 \sigma_A^2 = -r_0 \sigma_A E_0 - r_0^2 \sigma_A^2;$$

Условие некоррелированности матриц  $\hat{\mathbf{A}}_0$  и  $\hat{\mathbf{A}}'$ :

$$\text{cov} = \overline{A'A_0} - \overline{A'} \cdot \overline{A_0} = 0;$$

$$-r_0 \sigma_A E_0 - r_0^2 \sigma_A^2 - r_0 \sigma_A \delta (\overline{A} - r_0 \sigma_A \delta) = 0;$$

$$r_0 = -\frac{E_0 + \delta \overline{A}}{\sigma_A (1 - \delta^2)}.$$

В) Локальное поле на исходной матрице  $\hat{\mathbf{A}}$ :

$$\xi_i = H_i s_{0i} = \sum_{i \neq j}^N A_{ij} s_{0j} s_{0i} = \sum_{i \neq j}^N r_0 \sigma_A s_{0i}^2 s_{0j}^2 + \sum_{i \neq j}^N A'_{ij} s_{0j} s_{0i} = r_0 \sigma_A N + \sum_{i \neq j}^N A'_{ij} s_{0j} s_{0i};$$

$$\overline{\xi_i} = r_0 \sigma_A N + \sum_{i \neq j}^N \overline{A'_{ij} s_{0j} s_{0i}} = r_0 \sigma_A N;$$

$$\sigma_{\xi}^2 = \sum_{i \neq j}^N \sigma_{A'}^2 \sigma_{s_0}^2 \sigma_{s_0}^2 = N \sigma_A^2 (1 - r_0^2).$$

Г) Распределение  $f(A_{ij} s_{0j} s_{0i})$ :

$$A_{ij} s_{0j} s_{0i} = x;$$

$$\overline{A_{ij} s_{0j} s_{0i}} = r_0 \sigma_A + \overline{A'_{ij} \cdot s_{0j} s_{0i}} = r_0 \sigma_A;$$

$$f(A_{ij} s_{0j} s_{0i}) = (1 + ax) \cdot f(A_{ij}) = \frac{1}{2} (1 + ax);$$

$$r_0 \sigma_A = \overline{A_{ij} s_{0j} s_{0i}} = \int_{-1}^1 x f(x) dx = \int_{-1}^1 x \cdot \frac{1}{2} (1 + ax) dx = \frac{1}{2} \left( \frac{x^2}{2} + a \frac{x^3}{3} \right) \Big|_{-1}^1 = \frac{a}{3};$$

$$a = 3r_0 \sigma_A;$$

$$f(A_{ij} s_{0j} s_{0i}) = \frac{1}{2} (1 + 3r_0 \sigma_A x).$$

Д) Локальное поле на дискретизированной матрице  $\hat{C}$ :

$$\eta_i = h_i s_{0i} = \sum_{j \neq i}^N C_{ij} s_{0i} s_{0j} \equiv \sum_{j \neq i}^N \theta_j;$$

$$\theta_j = k, \text{ если } (2k - 1)\mu \leq A_{ij} s_{0i} s_{0j} \leq (2k + 1)\mu,$$

$$k \in [-q; q]; \quad \mu = 1/(2q + 1);$$

$$P_k = \Pr\{(2k - 1)\mu \leq A_{ij} s_{0i} s_{0j} \leq (2k + 1)\mu\} =$$

$$= 2\mu \frac{\frac{1}{2} + \frac{3}{2} r_0 \sigma_A \mu (2k - 1) + \frac{1}{2} + \frac{3}{2} r_0 \sigma_A \mu (2k + 1)}{2} = \mu(1 + 6\mu r_0 \sigma_A k);$$

$$\overline{\theta_j} = \sum_{k=-q}^q k \cdot \mu(1 + 6\mu r_0 \sigma_A k) = 12\mu^2 r_0 \sigma_A \sum_{k=1}^q k^2 = 12\mu^2 r_0 \sigma_A \frac{q(q+1)(2q+1)}{6} = \frac{1}{2\mu} r_0 \sigma_A (1 - \mu^2);$$

$$\sigma_{\theta}^2 = \sum_{k=-q}^q k^2 \cdot \mu(1 + 6\mu r_0 \sigma_A k) - \left( \frac{1}{2\mu} r_0 \sigma_A (1 - \mu^2) \right)^2 = \frac{1}{4\mu^2} \sigma_A^2 (1 - \mu^2) [1 - r_0^2 (1 - \mu^2)];$$

$$\overline{\eta_i} = N \overline{\theta_j} = \frac{1}{2\mu} N r_0 \sigma_A (1 - \mu^2);$$

$$\sigma_{\eta}^2 = N \sigma_{\theta}^2 = \frac{1}{4\mu^2} N \sigma_A^2 (1 - \mu^2) [1 - r_0^2 (1 - \mu^2)].$$

Е) Среднее от произведения локальных полей  $\overline{\xi_i \eta_i}$ :

$$\overline{\xi_i \eta_i} = \overline{\sum_{j \neq i}^N A_{ij} s_{0i} s_{0j} \cdot \sum_{k \neq i}^N C_{ik} s_{0i} s_{0k}} = \sum_{j \neq i}^N \overline{A_{ij} C_{ij}} + N(N-1) \cdot \overline{A_{ij} s_{0i} s_{0j}} \cdot \overline{C_{ik} s_{0i} s_{0k}} =$$

$$= N \sigma_A^2 (1 - \mu^2) + \overline{\xi_i} \cdot \overline{\eta_i} - \frac{1}{N} \overline{\xi_i} \cdot \overline{\eta_i}.$$

Ж) Корреляция локальных полей  $\rho$ :

$$\rho = \frac{\overline{\xi_i \eta_i} - \overline{\xi_i} \cdot \overline{\eta_i}}{\sigma_{\xi} \cdot \sigma_{\eta}} = \frac{N\sigma_A^2(1-\mu^2) - \frac{1}{N}r_0\sigma_A N \cdot Nr_0\sigma_A(1-\mu^2)}{\sqrt{N\sigma_A^2(1-r_0^2) \cdot N\sigma_A^2(1-\mu^2)[1-r_0^2(1-\mu^2)]}} = \sqrt{1 - \frac{\mu^2}{1-r_0^2(1-\mu^2)}}.$$

3) Вероятность несовпадения полей  $P = 1 - \Pr\{H_i h_i > 0 | H_i s_{0i} > 0\}$ :

$$\begin{aligned} \Pr\{H_i h_i > 0 | H_i s_{0i} > 0\} &= \Pr\{H_i h_i s_{0i}^2 > 0 | H_i s_{0i} > 0\} = \Pr\{H_i s_{0i} \cdot h_i s_{0i} > 0 | H_i s_{0i} > 0\} = \\ &= \frac{P\{H_i s_{0i} > 0 \cap h_i s_{0i} > 0 \cap H_i s_{0i} > 0\} + P\{H_i s_{0i} < 0 \cap h_i s_{0i} < 0 \cap H_i s_{0i} > 0\}}{\Pr\{H_i s_{0i} > 0\}} = \\ &= \frac{P\{H_i s_{0i} > 0 \cap h_i s_{0i} > 0\}}{\Pr\{H_i s_{0i} > 0\}}, \end{aligned}$$

$$P = 1 - \Pr\{H_i h_i > 0 | H_i s_{0i} > 0\} = 1 - \frac{P\{H_i s_{0i} > 0 \cap h_i s_{0i} > 0\}}{\Pr\{H_i s_{0i} > 0\}},$$

$$P = \frac{\frac{1}{2\pi\sqrt{1-\rho^2}} \int_{-\alpha}^{+\infty} \int_{-\beta}^{+\infty} \exp\left\{-\frac{1}{2(1-\rho^2)}[x^2 - 2\rho xy + y^2]\right\} dx dy}{\frac{1}{\sqrt{2\pi}} \int_{-\alpha}^{+\infty} \exp\left\{-\frac{x^2}{2}\right\} dx},$$

$$x = \frac{\xi}{\sigma_{\xi}}; \quad y = \frac{\eta}{\sigma_{\eta}}; \quad \alpha = \frac{\overline{\xi}}{\sigma_{\xi}}; \quad \beta = \frac{\overline{\eta}}{\sigma_{\eta}}; \quad \xi_i = H_i s_{0i}; \quad \eta_i = h_i s_{0i}.$$

СРЕДНЕЕ ЗНАЧЕНИЕ МАТРИЧНЫХ ЭЛЕМЕНТОВ:  $A_0 \neq 0$   
 ПОРОГИ НЕЙРОНОВ:  $B_i \neq 0$

**Вероятность несовпадения направлений локальных полей  
 в случайной точке пространства**

А) Исходная матрица  $\hat{\mathbf{A}}$  :

Равномерное распределение  $A_{ij}$  ;

$$\overline{A_{ij}} = A_0 ; \sigma_A^2 = 1/3 .$$

Б) Дискретизированная матрица  $\hat{\mathbf{C}}$  :

$$C_{ij} = C_0 + \text{grad}(A_{ij} - A_0) C_{ij} \in [-q; q] ;$$

$$\text{grad}(A_{ij} - A_0) = k , \text{ если } (2k - 1)\mu \leq (A_{ij} - A_0) \leq (2k + 1)\mu ,$$

$$k \in [-q; q] ; \mu = 1/(2q + 1) ;$$

$$\overline{C_{ij}} = C_0 + \mu \sum_{k=-q}^q k = C_0 ;$$

$$\sigma_C^2 = \sigma^2(\text{grad}(A_{ij} - A_0)) = \mu \sum_{k=-q}^q k^2 = 2\mu \sum_{k=1}^q k^2 = 2\mu \frac{q(q+1)(2q+1)}{6} = \frac{2q(q+1)}{3} = \frac{1}{4\mu^2} \sigma_A^2 (1 - \mu^2) .$$

В) Вектор-состояние  $\mathbf{S}$  :

$$s_i = \begin{cases} +1, & 1/2 ; \\ -1, & 1/2 ; \end{cases}$$

$$\overline{s_i} = 0 ; \sigma_s^2 = 1 .$$

*Вектор-состояние случайной точки  $\mathbf{S}$  не коррелирует с матрицами  $\hat{\mathbf{A}}$  и  $\hat{\mathbf{C}}$  !*

Г) Локальное поле на исходной матрице  $\hat{\mathbf{A}}$  :

$$H_i = -B_i + \sum_{j \neq i}^N A_{ij} s_j ;$$

$$\overline{H} = -B_i + \sum_{j \neq i}^N \overline{A_{ij} s_j} = -B_i + \sum_{j \neq i}^N \overline{A_{ij}} \overline{s_j} = -B_i ;$$

$$\sigma_H^2 = \sum_{j \neq i}^N \sigma^2(A_{ij} s_j) = \sum_{j \neq i}^N \sigma_A^2 \cdot \sigma_s^2 = N(\sigma_A^2 + A_0^2) .$$

Д) Локальное поле на дискретизированной матрице  $\hat{\mathbf{C}}$  :

$$h_i = -b_i + \sum_{j \neq i}^N C_{ij} s_j ;$$

$$\overline{h} = -b_i + \sum_{j \neq i}^N \overline{C_{ij} s_j} = -b_i + \sum_{j \neq i}^N \overline{C_{ij}} \overline{s_j} = -b_i ;$$

$$\sigma_h^2 = \sum_{j \neq i}^N \sigma^2(C_{ij} s_j) = \sum_{j \neq i}^N (\sigma_c^2 \cdot \sigma_s^2 + \overline{C_{ij}^2} \cdot \sigma_s^2) = N(\sigma_c^2 + C_0^2) = N \left( \frac{1}{4\mu^2} \sigma_A^2 (1 - \mu^2) + C_0^2 \right).$$

Е) Среднее от произведения локальных полей  $\overline{H_i h_i}$ :

$$\begin{aligned} \overline{H_i h_i} &= \overline{\left( -B_i + \sum_{j \neq i}^N A_{ij} s_j \right) \cdot \left( -b_i + \sum_{k \neq i}^N C_{ik} s_k \right)} = B_i b_i + \sum_{j \neq i}^N \overline{A_{ij} C_{ij}} + \sum_{\substack{j,k \\ j \neq k \neq i}}^N \overline{A_{ij} C_{ij} s_j s_k} = \\ &= B_i b_i + \sum_{j \neq i}^N \overline{A_{ij} C_{ij}} + \sum_{\substack{j,k \\ j \neq k \neq i}}^N \overline{A_{ij} C_{ij} s_j s_k} = N(B_i b_i + \overline{A_{ij} C_{ij}}); \end{aligned}$$

$$\begin{aligned} \overline{A_{ij} C_{ij}} &= \overline{A_{ij} (C_0 + \text{grad}(A_{ij} - A_0))} = A_0 C_0 + \sum_{k=-q}^q \left( k \int_{(2k-1)\mu}^{(2k+1)\mu} \frac{1}{2} x dx \right) = \\ &= A_0 C_0 + 4\mu^2 \sum_{k=1}^q k^2 = A_0 C_0 + \frac{1}{2\mu} \sigma_A^2 (1 - \mu^2); \end{aligned}$$

$$\gamma = \frac{1}{2\mu} \sigma_A^2 (1 - \mu^2);$$

$$\overline{H_i h_i} = N \left[ \frac{1}{2\mu} \sigma_A^2 (1 - \mu^2) + A_0 C_0 \right].$$

Ж) Корреляция локальных полей  $\rho$ :

$$\rho = \frac{\overline{H_i h_i} - \overline{H_i} \cdot \overline{h_i}}{\sigma_H \cdot \sigma_h} = \frac{N \left( \frac{1}{2\mu} \sigma_A^2 (1 - \mu^2) + A_0 C_0 \right)}{\sqrt{N(\sigma_A^2 + A_0)} \cdot \sqrt{N \left( \frac{1}{4\mu^2} \sigma_A^2 (1 - \mu^2) + C_0^2 \right)}} = \frac{\sigma_A^2 (1 - \mu^2) + A_0 C_0}{\sqrt{(\sigma_A^2 + A_0)(\sigma_A^2 (1 - \mu^2) + C_0^2)}}.$$

З) Вероятность несовпадения полей  $P = 1 - \Pr\{H_i h_i > 0\}$ :

$$P = 1 - \Pr\{H_i h_i > 0\} = 1 - P\{H_i > 0 \cap h_i > 0\} + P\{H_i < 0 \cap h_i < 0\};$$

$$\begin{aligned} &P\{H_i > 0 \cap h_i > 0\} = \\ &= \frac{1}{2\pi\sqrt{1-\rho^2}} \int_0^\infty \int_0^\infty \text{Exp} \left\{ -\frac{1}{2(1-\rho^2)} \left[ \left( \frac{H_i - \overline{H}}{\sigma_H} \right)^2 - 2\rho \left( \frac{H_i - \overline{H}}{\sigma_H} \right) \left( \frac{h_i - \overline{h}}{\sigma_h} \right) + \left( \frac{h_i - \overline{h}}{\sigma_h} \right)^2 \right] \right\} \frac{dH}{\sigma_H} \frac{dh}{\sigma_h} + \\ &= \frac{1}{2\pi\sqrt{1-\rho^2}} \int_0^\infty \int_0^\infty \text{Exp} \left\{ -\frac{1}{2(1-\rho^2)} \left[ \left( \frac{H_i + \overline{H}}{\sigma_H} \right)^2 - 2\rho \left( \frac{H_i + \overline{H}}{\sigma_H} \right) \left( \frac{h_i + \overline{h}}{\sigma_h} \right) + \left( \frac{h_i + \overline{h}}{\sigma_h} \right)^2 \right] \right\} \frac{dH}{\sigma_H} \frac{dh}{\sigma_h}; \end{aligned}$$

$$x = \frac{H_i}{\sigma_H}; \quad y = \frac{h_i}{\sigma_h}; \quad \alpha = \frac{\overline{H}}{\sigma_H}; \quad \beta = \frac{\overline{h}}{\sigma_h};$$

$$\begin{aligned} P &= 1 - \frac{1}{2\pi\sqrt{1-\rho^2}} \int_{-\alpha}^\infty \int_{-\beta}^\infty \text{Exp} \left\{ -\frac{1}{2(1-\rho^2)} [x^2 - 2\rho xy + y^2] \right\} dx dy + \\ &+ \frac{1}{2\pi\sqrt{1-\rho^2}} \int_{\alpha}^\infty \int_{\beta}^\infty \text{Exp} \left\{ -\frac{1}{2(1-\rho^2)} [x^2 - 2\rho xy + y^2] \right\} dx dy. \end{aligned}$$

И) Оптимальное  $C_0$ :

$$P = 1 - \frac{1}{2\pi} \int_{-\alpha}^{+\infty} \int_u^{+\infty} e^{-\frac{x^2}{2} - \frac{t^2}{2}} dx dt + \frac{1}{2\pi} \int_{\alpha}^{+\infty} \int_v^{+\infty} e^{-\frac{x^2}{2} - \frac{t^2}{2}} dx dt;$$

$$\frac{dP}{dC_0} = \frac{\partial P}{\partial u} \frac{\partial u}{\partial \rho} \frac{\partial \rho}{\partial C_0} + \frac{\partial P}{\partial v} \frac{\partial v}{\partial \rho} \frac{\partial \rho}{\partial C_0} = \left[ \frac{\partial P}{\partial u} \frac{\partial u}{\partial \rho} + \frac{\partial P}{\partial v} \frac{\partial v}{\partial \rho} \right] \cdot \frac{\partial \rho}{\partial C_0} = 0;$$

$$\frac{\partial \rho}{\partial C_0} = 0; \quad \frac{\partial \rho}{\partial C_0} = \frac{A_0(\sigma_C^2 + C_0^2) - (\gamma + A_0 C_0)C_0}{(\sigma_C^2 + C_0^2)\sqrt{(\sigma_A^2 + A_0^2)(\sigma_C^2 + C_0^2)}} = 0;$$

$$C_0 = \frac{\sigma_C^2}{\gamma} A_0 = \frac{\frac{1}{4\mu^2} \sigma_A^2 (1 - \mu^2)}{\frac{1}{2\mu} \sigma_A^2 (1 - \mu^2)} A_0 = \frac{A_0}{2\mu};$$

$$C_0 = \frac{A_0}{2\mu}.$$

К) Оптимальное  $b_i$ :

$$\frac{dP}{db_i} = \frac{\partial P}{\partial \alpha} \frac{\partial \alpha}{\partial b_i} + \frac{\partial P}{\partial \beta} \frac{\partial \beta}{\partial b_i} + \frac{\partial P}{\partial \rho} \frac{\partial \rho}{\partial b_i} = \frac{\partial P}{\partial \beta} \frac{\partial \beta}{\partial b_i} = 0;$$

$$\frac{\partial P}{\partial \beta} \frac{\partial \beta}{\partial b_i} = \frac{1}{2\pi\sigma_h\sqrt{1-\rho^2}} \int_0^{\infty} \text{Exp} \left\{ -\frac{1}{2(1-\rho^2)} [(x-\alpha)^2 - 2\rho(x-\alpha)\beta + \beta^2] \right\} dx -$$

$$- \frac{1}{2\pi\sigma_h\sqrt{1-\rho^2}} \int_0^{\infty} \text{Exp} \left\{ -\frac{1}{2(1-\rho^2)} [(x+\alpha)^2 - 2\rho(x+\alpha)\beta + \beta^2] \right\} dx = 0;$$

$$(x-\alpha)^2 - 2\rho(x-\alpha)\beta + \beta^2 = (x+\alpha)^2 - 2\rho(x+\alpha)\beta + \beta^2;$$

$$\alpha = \rho\beta;$$

$$b_i = \frac{1}{\rho} \frac{\sigma_h}{\sigma_H} B_i = \frac{\sigma_C^2}{\gamma} B_i;$$

$$b_i = \frac{B_i}{2\mu}.$$

$$\text{Корреляция: } \rho = \sqrt{1 - \frac{\mu^2}{1 + A_0^2/\sigma_A^2}}.$$

**МАТРИЦЫ С НОРМАЛЬНЫМ РАСПРЕДЕЛЕНИЕМ ЭЛЕМЕНТОВ**  
**СРЕДНЕЕ ЗНАЧЕНИЕ МАТРИЧНЫХ ЭЛЕМЕНТОВ:  $A_0 = 0$**   
**ПОРОГИ НЕЙРОНОВ:  $B_i = 0$**

**Вероятность несовпадения направлений локальных полей  
в случайной точке пространства**

А) Исходная матрица  $\hat{\mathbf{A}}$  :

Нормальное распределение  $A_{ij}$  ;  
 $\overline{A_{ij}} = 0$  ;  $\sigma_A^2 = 1$  .

Б) Дискретизированная матрица  $\hat{\mathbf{C}}$  :

$$C_{ij} \in [-q; q];$$

$$C_{ij} = k, \text{ если } x_{k-1} \leq A_{ij} \leq x_k, \quad k \in [-q; q];$$

$$\overline{C_{ij}} = \sum_{k=-q}^q k P_k = 0;$$

$$\sigma_C^2 = \overline{C_{ij}^2} - \overline{C_{ij}}^2 = \overline{C_{ij}^2} = \sum_{k=-q}^q k^2 P_k = 2 \sum_{k=1}^q k^2 P_k .$$

В) Вектор-состояние  $\mathbf{S}$  :

$$s_i = \begin{cases} +1, & 1/2; \\ -1, & 1/2; \end{cases}$$

$$\overline{s_i} = 0; \quad \sigma_s^2 = 1 .$$

*Вектор-состояние случайной точки  $\mathbf{S}$  не коррелирует с матрицами  $\hat{\mathbf{A}}$  и  $\hat{\mathbf{C}}$  !*

Г) Локальное поле на исходной матрице  $\hat{\mathbf{A}}$  :

$$H_i = \sum_{j \neq i}^N A_{ij} s_j ;$$

$$\overline{H} = \sum_{j \neq i}^N \overline{A_{ij} s_j} = \sum_{j \neq i}^N \overline{A_{ij}} \overline{s_j} = 0 ;$$

$$\sigma_H^2 = \sum_{j \neq i}^N \sigma^2(A_{ij} s_j) = \sum_{j \neq i}^N \sigma_A^2 \cdot \sigma_s^2 = N \sigma_A^2 .$$

Д) Локальное поле на дискретизированной матрице  $\hat{\mathbf{C}}$  :

$$h_i = \sum_{j \neq i}^N C_{ij} s_j ;$$

$$\overline{h} = \sum_{j \neq i}^N \overline{C_{ij} s_j} = \sum_{j \neq i}^N \overline{C_{ij}} \overline{s_j} = 0 ;$$

$$\sigma_h^2 = \sum_{j \neq i}^N \sigma^2(C_{ij} s_j) = \sum_{j \neq i}^N \sigma_C^2 \cdot \sigma_s^2 = N \sigma_C^2 = 2N \sum_{k=-q}^q k^2 P_k .$$

Е) Среднее от произведения локальных полей  $\overline{H_i h_i}$ :

$$\begin{aligned}\overline{H_i h_i} &= \overline{\left( \sum_{j \neq i}^N A_{ij} s_j \right) \cdot \left( \sum_{k \neq i}^N C_{ik} s_k \right)} = \overline{\sum_{j \neq i}^N A_{ij} C_{ij}} + \overline{\sum_{\substack{j,k \\ j \neq k \neq i}}^N A_{ij} C_{ij} s_j s_k} = \sum_{j \neq i}^N \overline{A_{ij} C_{ij}} + \sum_{\substack{j,k \\ j \neq k \neq i}}^N \overline{A_{ij} C_{ij} s_j s_k} = N \overline{A_{ij} C_{ij}}; \\ \gamma &= \overline{A_{ij} C_{ij}} = \sum_{k=-q}^q \left( k \left[ \frac{1}{\sqrt{2\pi\sigma_A}} \int_{x_{k-1}}^{x_k} x e^{-\frac{x^2}{2\sigma_A^2}} dx \right] \right) = \frac{2\sigma_A}{\sqrt{2\pi}} \sum_{k=1}^q k \left( -e^{-\frac{x^2}{2\sigma_A^2}} \right) \Big|_{x_{k-1}}^{x_k} = \\ &= \frac{2\sigma_A}{\sqrt{2\pi}} \sum_{k=1}^q k \left[ e^{-\frac{x_0^2}{2\sigma_A^2}} \cdot \left( 1 - \frac{k(k-1)}{q(q+1)} \right) - e^{-\frac{x_0^2}{2\sigma_A^2}} \cdot \left( 1 - \frac{k(k+1)}{q(q+1)} \right) \right] = \frac{2q+1}{3} \sigma_A \sqrt{\frac{2}{\pi}} e^{-\frac{x_0^2}{2\sigma_A^2}}; \\ \overline{H_i h_i} &= N\gamma = N \frac{2q+1}{3} \sigma_A \sqrt{\frac{2}{\pi}} e^{-\frac{x_0^2}{2\sigma_A^2}}.\end{aligned}$$

Ж) Корреляция локальных полей  $\rho$ :

$$\rho = \frac{\overline{H_i h_i} - \overline{H_i} \cdot \overline{h_i}}{\sigma_H \cdot \sigma_h} = \frac{N\gamma}{\sqrt{N\sigma_A^2} \cdot N\sigma_C^2} = \frac{(2q+1)e^{-\frac{x_0^2}{2\sigma_A^2}}}{3\sqrt{\pi} \sum_{k=1}^q k^2 P_k}.$$

З) Вероятность несовпадения полей  $\Pr\{H_i h_i > 0\}$ :

$$P = 1 - \Pr\{H_i h_i > 0\} = 1 - P\{H_i > 0 \cap h_i > 0\} + P\{H_i < 0 \cap h_i < 0\};$$

$$\begin{aligned}P\{H_i > 0 \cap h_i > 0\} &= \\ &= \frac{1}{2\pi\sqrt{1-\rho^2}} \int_0^\infty \int_0^\infty \text{Exp} \left\{ -\frac{1}{2(1-\rho^2)} \left[ \left( \frac{H_i - \bar{H}}{\sigma_H} \right)^2 - 2\rho \left( \frac{H_i - \bar{H}}{\sigma_H} \right) \left( \frac{h_i - \bar{h}}{\sigma_h} \right) + \left( \frac{h_i - \bar{h}}{\sigma_h} \right)^2 \right] \right\} \frac{dH}{\sigma_H} \frac{dh}{\sigma_h} = \\ &= \frac{1}{2\pi\sqrt{1-\rho^2}} \int_0^\infty \int_0^\infty \text{Exp} \left\{ -\frac{1}{2(1-\rho^2)} \left[ \left( \frac{H_i}{\sigma_H} \right)^2 - 2\rho \left( \frac{H_i}{\sigma_H} \right) \left( \frac{h_i}{\sigma_h} \right) + \left( \frac{h_i}{\sigma_h} \right)^2 \right] \right\} \frac{dH}{\sigma_H} \frac{dh}{\sigma_h};\end{aligned}$$

$$\begin{aligned}P\{H_i < 0 \cap h_i < 0\} &= \\ &= \frac{1}{2\pi\sqrt{1-\rho^2}} \int_0^\infty \int_0^\infty \text{Exp} \left\{ -\frac{1}{2(1-\rho^2)} \left[ \left( \frac{H_i - \bar{H}}{\sigma_H} \right)^2 - 2\rho \left( \frac{H_i - \bar{H}}{\sigma_H} \right) \left( \frac{h_i - \bar{h}}{\sigma_h} \right) + \left( \frac{h_i - \bar{h}}{\sigma_h} \right)^2 \right] \right\} \frac{dH}{\sigma_H} \frac{dh}{\sigma_h} = \\ &= \frac{1}{2\pi\sqrt{1-\rho^2}} \int_0^\infty \int_0^\infty \text{Exp} \left\{ -\frac{1}{2(1-\rho^2)} \left[ \left( \frac{H_i}{\sigma_H} \right)^2 - 2\rho \left( \frac{H_i}{\sigma_H} \right) \left( \frac{h_i}{\sigma_h} \right) + \left( \frac{h_i}{\sigma_h} \right)^2 \right] \right\} \frac{dH}{\sigma_H} \frac{dh}{\sigma_h};\end{aligned}$$

$$\begin{aligned}\Pr\{H_i h_i > 0\} &= \frac{1}{\pi\sqrt{1-\rho^2}} \int_0^\infty \int_0^\infty \text{Exp} \left\{ -\frac{1}{2(1-\rho^2)} \left[ \left( \frac{H_i}{\sigma_H} \right)^2 - 2\rho \left( \frac{H_i}{\sigma_H} \right) \left( \frac{h_i}{\sigma_h} \right) + \left( \frac{h_i}{\sigma_h} \right)^2 \right] \right\} \frac{dH}{\sigma_H} \frac{dh}{\sigma_h}; \\ & \quad x = \frac{H_i}{\sigma_H}; \quad y = \frac{h_i}{\sigma_h};\end{aligned}$$

$$P = 1 - \Pr\{H_i h_i > 0\} = 1 - \frac{1}{\pi\sqrt{1-\rho^2}} \int_0^\infty \int_0^\infty \text{Exp}\left\{-\frac{1}{2(1-\rho^2)}[x^2 - 2\rho xy + y^2]\right\} dx dy.$$

$$P = \frac{1}{2} - \frac{1}{\pi} \arcsin \rho \approx \frac{1}{\pi} \sqrt{2(1-\rho)}.$$

### Вероятность несовпадения направлений локальных полей в минимуме исходного функционала

А) Минимум исходного функционала  $\mathbf{S}_0$ ;

$$s_{0i} = \begin{cases} +1, & 1/2 \\ -1, & 1/2 \end{cases};$$

$$\overline{s_{0i}} = 0; \quad \sigma_{s_0}^2 = 1.$$

Условие того, что  $\mathbf{S}_0$  минимум:

$$H_i s_{0i} > 0, \quad \forall i.$$

Б) Выделим минимум из матрицы  $\hat{\mathbf{A}}$ :

$$A_{ij} = A_{0ij} + A'_{ij} = r_0 \sigma_A s_{0i} s_{0j} + A'_{ij}; \quad \sum_{i=0}^{\infty} r_i^2 = 1$$

$$A_{0ij} = \frac{r_0 \sigma_A}{N^2} \sum_{\substack{i,j \\ i \neq j}}^N s_{0i} s_{0j} = \frac{r_0 \sigma_A}{N^2} \left[ \sum_{i,j}^N s_{0i} s_{0j} - \sum_i^N (s_{0i})^2 \right] = \frac{r_0 \sigma_A}{N^2} \left[ \sum_i^N s_{0i} \sum_j^N s_{0j} - N \right] =$$

$$= \frac{r_0 \sigma_A}{N^2} \left[ \left( \sum_i^N s_{0i} \right)^2 - N \right] = r_0 \sigma_A \delta;$$

$$\bar{\delta} = \frac{1}{N^2} \left[ \overline{\left( \sum_i^N s_{0i} \right)^2} - N \right] = \frac{1}{N^2} \cdot (-N) = -\frac{1}{N} \approx 0;$$

$$\overline{A_{0ij}} = \frac{r_0 \sigma_A}{N^2} \bar{\delta} = 0;$$

$$\overline{A_{0ij}^2} = \frac{r_0^2 \sigma_A^2}{N^2} \sum_{\substack{i,j \\ i \neq j}}^N s_{0i}^2 s_{0j}^2 = r_0^2 \sigma_A^2;$$

$$\sigma_{A_0}^2 = \overline{A_0^2} - \overline{A_0}^2 = r_0^2 \sigma_A^2 - r_0^2 \sigma_A^2 \delta^2 = r_0^2 \sigma_A^2 (1 - \delta^2);$$

$$\overline{A'_{ij}} = \overline{A_{ij}} - \overline{A_{0ij}} = \overline{A_{ij}} - r_0 \sigma_A \delta;$$

$$\sigma_{A'}^2 = \sigma_A^2 - \sigma_{A_0}^2 = \sigma_A^2 - r_0^2 \sigma_A^2 (1 - \delta^2) = \sigma_A^2 (1 - r_0^2);$$

$$\overline{A'A_0} = \overline{(A - A_0)A_0} = \overline{AA_0} - \overline{A_0^2} = r_0 \sigma_A \sum_{\substack{i,j \\ i \neq j}}^N A_{ij} s_{0i} s_{0j} - r_0^2 \sigma_A^2 = -r_0 \sigma_A E_0 - r_0^2 \sigma_A^2;$$

Условие некоррелированности матриц  $\hat{\mathbf{A}}_0$  и  $\hat{\mathbf{A}}'$ :

$$\text{cov} = \overline{A'A_0} - \overline{A'} \cdot \overline{A_0} = 0;$$

$$-r_0 \sigma_A E_0 - r_0^2 \sigma_A^2 - r_0 \sigma_A \delta (\overline{A} - r_0 \sigma_A \delta) = 0;$$

$$r_0 = -\frac{E_0 + \delta \bar{A}}{\sigma_A (1 - \delta^2)}.$$

В) Локальное поле на исходной матрице  $\hat{\mathbf{A}}$ :

$$\begin{aligned}\xi_i &= H_i s_{0i} = \sum_{i \neq j}^N A_{ij} s_{0j} s_{0i} = \sum_{i \neq j}^N r_0 \sigma_A s_{0i}^2 s_{0j}^2 + \sum_{i \neq j}^N A'_{ij} s_{0j} s_{0i} = r_0 \sigma_A N + \sum_{i \neq j}^N A'_{ij} s_{0j} s_{0i}; \\ \bar{\xi}_i &= r_0 \sigma_A N + \sum_{i \neq j}^N \overline{A'_{ij} s_{0j} s_{0i}} = r_0 \sigma_A N; \\ \sigma_\xi^2 &= \sum_{i \neq j}^N \sigma_{A'}^2 \sigma_{s_0}^2 = N \sigma_A^2 (1 - r_0^2).\end{aligned}$$

Г) Распределение  $f(A_{ij} s_{0j} s_{0i})$ :

$$\begin{aligned}A_{ij} s_{0j} s_{0i} &= x; \\ \overline{A_{ij} s_{0j} s_{0i}} &= r_0 \sigma_A + \overline{A'_{ij} \cdot s_{0j} s_{0i}} = r_0 \sigma_A; \\ \sigma^2(A_{ij} s_{0j} s_{0i}) &= N \sigma_A^2 (1 - r_0^2) \\ f(A_{ij} s_{0j} s_{0i}) &= \frac{1}{\sigma_A \sqrt{2\pi(1-r_0^2)}} \int_{-\infty}^{+\infty} e^{-\frac{(x-r_0\sigma_A)^2}{2\sigma_A^2(1-r_0^2)}} dx.\end{aligned}$$

Д) Локальное поле на дискретизированной матрице  $\hat{\mathbf{C}}$ :

$$\eta_i = h_i s_{0i} = \sum_{j \neq i}^N C_{ij} s_{0i} s_{0j} \equiv \sum_{j \neq i}^N \theta_j;$$

$$\theta_j = k, \text{ если } x_{k-1} \leq A_{ij} \leq x_k, \quad k \in [-q; q];$$

$$P_k = \Pr\{x_{k-1} \leq A_{ij} s_{0i} s_{0j} \leq x_k\} = \frac{1}{\sigma_A \sqrt{2\pi(1-r_0^2)}} \int_{x_{k-1}}^{x_k} e^{-\frac{(x-r_0\sigma_A)^2}{2\sigma_A^2(1-r_0^2)}} dx;$$

$$\bar{\theta}_j = \sum_{k=-q}^q k P_k;$$

$$\sigma_\theta^2 = \sum_{k=-q}^q k^2 P_k - \bar{\theta}^2 = \sum_{k=-q}^q k^2 P_k - \left( \sum_{k=-q}^q k P_k \right)^2;$$

$$\bar{\eta}_i = N \bar{\theta}_j = N \sum_{k=-q}^q k P_k;$$

$$\sigma_\eta^2 = N \sigma_\theta^2 = N \left[ \sum_{k=-q}^q k^2 P_k - \left( \sum_{k=-q}^q k P_k \right)^2 \right].$$

Е) Среднее от произведения локальных полей  $\overline{\xi_i \eta_i}$ :

$$\overline{\xi_i \eta_i} = \sum_{j \neq i}^N \overline{A_{ij} s_{0i} s_{0j}} \cdot \sum_{k \neq i}^N \overline{C_{ik} s_{0i} s_{0k}} = \sum_{j \neq i}^N \overline{A_{ij} C_{ij}} + N(N-1) \cdot \overline{A_{ij} s_{0i} s_{0j}} \cdot \overline{C_{ik} s_{0i} s_{0k}} = N\gamma + \bar{\xi}_i \cdot \bar{\eta}_i - \frac{1}{N} \bar{\xi}_i \cdot \bar{\eta}_i.$$

Ж) Корреляция локальных полей  $\rho$ :

$$\rho = \frac{\overline{\xi_i \eta_i} - \overline{\xi_i} \cdot \overline{\eta_i}}{\sigma_\xi \cdot \sigma_\eta} = \frac{N\gamma - \frac{1}{N} \overline{\xi} \cdot \overline{\eta}}{\sigma_\xi \sigma_\eta}.$$

3) Вероятность несовпадения полей  $P = 1 - \Pr\{H_i h_i > 0 | H_i s_{0i} > 0\}$ :

$$\begin{aligned} \Pr\{H_i h_i > 0 | H_i s_{0i} > 0\} &= \Pr\{H_i h_i s_{0i}^2 > 0 | H_i s_{0i} > 0\} = \Pr\{H_i s_{0i} \cdot h_i s_{0i} > 0 | H_i s_{0i} > 0\} = \\ &= \frac{P\{H_i s_{0i} > 0 \cap h_i s_{0i} > 0 \cap H_i s_{0i} > 0\} + P\{H_i s_{0i} < 0 \cap h_i s_{0i} < 0 \cap H_i s_{0i} > 0\}}{\Pr\{H_i s_{0i} > 0\}} = \frac{P\{H_i s_{0i} > 0 \cap h_i s_{0i} > 0\}}{\Pr\{H_i s_{0i} > 0\}} \end{aligned}$$

$$P = \Pr\{H_i h_i > 0 | H_i s_{0i} > 0\} = \frac{P\{H_i s_{0i} > 0 \cap h_i s_{0i} > 0\}}{\Pr\{H_i s_{0i} > 0\}},$$

$$P = 1 - \Pr\{H_i h_i > 0 | H_i s_{0i} > 0\} = 1 - \frac{\frac{1}{2\pi\sqrt{1-\rho^2}} \int_{-\alpha}^{+\infty} \int_{-\beta}^{+\infty} \exp\left\{-\frac{1}{2(1-\rho^2)}[x^2 - 2\rho xy + y^2]\right\} dx dy}{\frac{1}{\sqrt{2\pi}} \int_{-\alpha}^{+\infty} \exp\left\{-\frac{x^2}{2}\right\} dx},$$

$$x = \frac{\xi}{\sigma_\xi}; \quad y = \frac{\eta}{\sigma_\eta}; \quad \alpha = \frac{\overline{\xi}}{\sigma_\xi}; \quad \beta = \frac{\overline{\eta}}{\sigma_\eta}; \quad \xi_i = H_i s_{0i}; \quad \eta_i = h_i s_{0i}.$$

СРЕДНЕЕ ЗНАЧЕНИЕ МАТРИЧНЫХ ЭЛЕМЕНТОВ:  $A_0 \neq 0$

ПОРОГИ НЕЙРОНОВ:  $B_i \neq 0$

**Вероятность совпадения направлений локальных полей  
в случайной точке пространства**

А) Исходная матрица  $\hat{A}$  :

Нормальное распределение  $A_{ij}$  ;

$$\overline{A_{ij}} = A_0 ; \sigma_A^2 = 1.$$

Б) Дискретизированная матрица  $\hat{C}$  :

$$C_{ij} = C_0 + \text{grad}(A_{ij} - A_0) ; C_{ij} \in [-q; q] ;$$

$$\text{grad}(A_{ij} - A_0) = k, \text{ если } x_{k-1} \leq (A_{ij} - A_0) \leq x_k, k \in [-q; q] ;$$

$$\overline{C_{ij}} = C_0 + \sum_{k=-q}^q k P_k = C_0 ;$$

$$\sigma_C^2 = \sigma^2(\text{grad}(A_{ij} - A_0)) = \sum_{k=-q}^q k^2 P_k = 2 \sum_{k=1}^q k^2 P_k .$$

В) Вектор-состояние  $\mathbf{S}$  :

$$s_i = \begin{cases} +1, & 1/2 ; \\ -1, & 1/2 ; \end{cases}$$

$$\overline{s_i} = 0 ; \sigma_s^2 = 1 .$$

*Вектор-состояние случайной точки  $\mathbf{S}$  не коррелирует с матрицами  $\hat{A}$  и  $\hat{C}$  !*

Г) Локальное поле на исходной матрице  $\hat{A}$  :

$$H_i = -B_i + \sum_{j \neq i}^N A_{ij} s_j ;$$

$$\overline{H} = -B_i + \sum_{j \neq i}^N \overline{A_{ij} s_j} = -B_i + \sum_{j \neq i}^N \overline{A_{ij}} \overline{s_j} = -B_i ;$$

$$\sigma_H^2 = \sum_{j \neq i}^N \sigma^2(A_{ij} s_j) = \sum_{j \neq i}^N \sigma_A^2 \cdot \sigma_s^2 = N(\sigma_A^2 + A_0^2) .$$

Д) Локальное поле на дискретизированной матрице  $\hat{C}$  :

$$h_i = -b_i + \sum_{j \neq i}^N C_{ij} s_j ;$$

$$\overline{h} = -b_i + \sum_{j \neq i}^N \overline{C_{ij} s_j} = -b_i + \sum_{j \neq i}^N \overline{C_{ij}} \overline{s_j} = -b_i ;$$

$$\sigma_h^2 = \sum_{j \neq i}^N \sigma^2(C_{ij} s_j) = \sum_{j \neq i}^N (\sigma_C^2 \cdot \sigma_s^2 + \overline{C_{ij}}^2 \cdot \sigma_s^2) = N(\sigma_C^2 + C_0^2) = N \left( 2 \sum_{k=1}^q k^2 P_k + C_0^2 \right) .$$

Е) Среднее от произведения локальных полей  $\overline{H_i h_i}$  :

$$\begin{aligned}\overline{H_i h_i} &= \overline{\left(-B_i + \sum_{j \neq i}^N A_{ij} s_j\right) \cdot \left(-b_i + \sum_{k \neq i}^N C_{ik} s_k\right)} = \overline{B_i b_i} + \overline{\sum_{j \neq i}^N A_{ij} C_{ij}} + \overline{\sum_{\substack{j,k \\ j \neq k \neq i}}^N A_{ij} C_{ij} s_j s_k} = \\ &= B_i b_i + \sum_{j \neq i}^N \overline{A_{ij} C_{ij}} + \sum_{\substack{j,k \\ j \neq k \neq i}}^N \overline{A_{ij} C_{ij} s_j s_k} = N(B_i b_i + \overline{A_{ij} C_{ij}}); \\ \overline{A_{ij} C_{ij}} &= \overline{A_{ij} (C_0 + \text{grad}(A_{ij} - A_0))} = A_0 C_0 + \sum_{k=-q}^q \left( k \left[ \frac{1}{\sqrt{2\pi\sigma_A}} \int_{x_{k-1}}^{x_k} x e^{-\frac{x^2}{2\sigma_A^2}} dx \right] \right) = A_0 C_0 + \gamma; \\ \gamma &= \frac{2q+1}{3} \sigma_A \sqrt{\frac{2}{\pi}} e^{-\frac{x_0^2}{2\sigma_A^2}}; \\ \overline{H_i h_i} &= N[\gamma + A_0 C_0].\end{aligned}$$

Ж) Корреляция локальных полей  $\rho$ :

$$\rho = \frac{\overline{H_i h_i} - \overline{H_i} \cdot \overline{h_i}}{\sigma_H \cdot \sigma_h} = \frac{N(\gamma + A_0 C_0)}{\sqrt{N(\sigma_A^2 + A_0)} \cdot \sqrt{N(\sigma_C^2 + C_0^2)}} = \frac{\gamma + A_0 C_0}{\sqrt{(\sigma_A^2 + A_0)(\sigma_C^2 + C_0^2)}}.$$

З) Вероятность несовпадения полей  $P = 1 - \Pr\{H_i h_i > 0\}$ :

$$P = 1 - \Pr\{H_i h_i > 0\} = P\{H_i > 0 \cap h_i > 0\} + P\{H_i < 0 \cap h_i < 0\};$$

$$\begin{aligned}P\{H_i > 0 \cap h_i > 0\} &= \\ &= \frac{1}{2\pi\sqrt{1-\rho^2}} \int_0^\infty \int_0^\infty \text{Exp} \left\{ -\frac{1}{2(1-\rho^2)} \left[ \left( \frac{H_i - \overline{H}}{\sigma_H} \right)^2 - 2\rho \left( \frac{H_i - \overline{H}}{\sigma_H} \right) \left( \frac{h_i - \overline{h}}{\sigma_h} \right) + \left( \frac{h_i - \overline{h}}{\sigma_h} \right)^2 \right] \right\} \frac{dH}{\sigma_H} \frac{dh}{\sigma_h} + \\ &= \frac{1}{2\pi\sqrt{1-\rho^2}} \int_0^\infty \int_0^\infty \text{Exp} \left\{ -\frac{1}{2(1-\rho^2)} \left[ \left( \frac{H_i + \overline{H}}{\sigma_H} \right)^2 - 2\rho \left( \frac{H_i + \overline{H}}{\sigma_H} \right) \left( \frac{h_i + \overline{h}}{\sigma_h} \right) + \left( \frac{h_i + \overline{h}}{\sigma_h} \right)^2 \right] \right\} \frac{dH}{\sigma_H} \frac{dh}{\sigma_h};\end{aligned}$$

$$x = \frac{H_i}{\sigma_H}; \quad y = \frac{h_i}{\sigma_h}; \quad \alpha = \frac{\overline{H}}{\sigma_H}; \quad \beta = \frac{\overline{h}}{\sigma_h};$$

$$\begin{aligned}P &= 1 - \frac{1}{2\pi\sqrt{1-\rho^2}} \int_{-\alpha}^\infty \int_{-\beta}^\infty \text{Exp} \left\{ -\frac{1}{2(1-\rho^2)} [x^2 - 2\rho xy + y^2] \right\} dx dy + \\ &+ \frac{1}{2\pi\sqrt{1-\rho^2}} \int_{\alpha}^\infty \int_{\beta}^\infty \text{Exp} \left\{ -\frac{1}{2(1-\rho^2)} [x^2 - 2\rho xy + y^2] \right\} dx dy.\end{aligned}$$

И) Оптимальное  $C_0$ :

$$\begin{aligned}P &= 1 - \frac{1}{2\pi} \int_{-\alpha}^{+\infty} \int_u^{+\infty} e^{-\frac{x^2}{2} - \frac{t^2}{2}} dx dt + \frac{1}{2\pi} \int_{\alpha}^{+\infty} \int_v^{+\infty} e^{-\frac{x^2}{2} - \frac{t^2}{2}} dx dt; \\ \frac{dP}{dC_0} &= \frac{\partial P}{\partial u} \frac{\partial u}{\partial \rho} \frac{\partial \rho}{\partial C_0} + \frac{\partial P}{\partial v} \frac{\partial v}{\partial \rho} \frac{\partial \rho}{\partial C_0} = \left[ \frac{\partial P}{\partial u} \frac{\partial u}{\partial \rho} + \frac{\partial P}{\partial v} \frac{\partial v}{\partial \rho} \right] \cdot \frac{\partial \rho}{\partial C_0} = 0; \\ \frac{\partial \rho}{\partial C_0} &= 0; \quad \frac{\partial \rho}{\partial C_0} = \frac{A_0(\sigma_C^2 + C_0^2) - (\gamma + A_0 C_0)C_0}{(\sigma_C^2 + C_0^2)\sqrt{(\sigma_A^2 + A_0)(\sigma_C^2 + C_0^2)}} = 0;\end{aligned}$$

$$C_0 = \frac{\sigma_c^2}{\gamma} A_0.$$

К) Оптимальное  $b_i$ :

$$\begin{aligned} \frac{dP}{db_i} &= \frac{\partial P}{\partial \alpha} \frac{\partial \alpha}{\partial b_i} + \frac{\partial P}{\partial \beta} \frac{\partial \beta}{\partial b_i} + \frac{\partial P}{\partial \rho} \frac{\partial \rho}{\partial b_i} = \frac{\partial P}{\partial \beta} \frac{\partial \beta}{\partial b_i} = 0; \\ \frac{\partial P}{\partial \beta} \frac{\partial \beta}{\partial b_i} &= \frac{1}{2\pi\sigma_h\sqrt{1-\rho^2}} \int_0^\infty \text{Exp}\left\{-\frac{1}{2(1-\rho^2)}[(x-\alpha)^2 - 2\rho(x-\alpha)\beta + \beta^2]\right\} dx - \\ &- \frac{1}{2\pi\sigma_h\sqrt{1-\rho^2}} \int_0^\infty \text{Exp}\left\{-\frac{1}{2(1-\rho^2)}[(x+\alpha)^2 - 2\rho(x+\alpha)\beta + \beta^2]\right\} dx = 0; \\ (x-\alpha)^2 - 2\rho(x-\alpha)\beta + \beta^2 &= (x+\alpha)^2 - 2\rho(x+\alpha)\beta + \beta^2; \\ \alpha &= \rho\beta; \\ b_i &= \frac{1}{\rho} \frac{\sigma_h}{\sigma_H} B_i = \frac{\sigma_c^2}{\gamma} B_i; \\ b_i &= \frac{\sigma_c^2}{\gamma} B_i. \end{aligned}$$

$$\text{Корреляция: } \rho = \sqrt{\frac{A_0^2 + \gamma^2 / \sigma_c^2}{A_0^2 + \sigma_A^2}}.$$

## Список литературы

1. *Гэри М., Джонсон Д.* Вычислительные машины и труднорешаемые задачи // М.: Мир, 1982.
2. *Пападимитриу Х., Стайглиц К.* Комбинаторная оптимизация. Алгоритмы и сложность // М.: Мир, 1985.
3. *Papadimitriou C.H., Steiglitz K.* Combinatorial Optimization // Prentice-Hall, Englewood, Cliffs, New Jersey, 1982.
4. *Cook W.J., Cunningham W.H., Pulleyblank W.R., and Schrijver A.* Combinatorial Optimization // J. Wiley & Sons, New York, 1998.
5. *Korte B., Vygen J.* Combinatorial Optimization // Springer, Berlin and Heidelberg, 2000.
6. *Taha Х.* Введение в исследование операций: в 2-х книгах // кн. 2.: Перев. с англ.– М.: Мир, 1985.
7. *Davis M.D., Sigal R., and Wyuker E.J.* Computability, Complexity and Languages // Academic Press, San Diego, 1994.
8. *Рейнгольд Э., Нивергельт Ю., Део Н.* Комбинаторные алгоритмы. Теория и практика // М.: Мир, 1980.
9. *Липский В.* Комбинаторика для программистов // М.: Мир, 1988.
10. *Ху Т.* Целочисленное программирование и потоки в сетях // М.: Мир, 1974.
11. *Сергиенко И.В., Лебедева Т.Т., Роцин В.А.* Приближенные методы решения дискретных задач оптимизации // Киев: Наука думка, 1980.
12. *Михалевич В.С., Кукса А.И.* Методы последовательной оптимизации в дискретных сетевых задачах оптимального распределения ресурсов // М.: Наука, 1983.
13. *Васильев Ф.П.* Методы оптимизации // М.: Факториал Пресс, 2002.
14. *Вентцель Е.С.* Исследование операций. Задачи, принципы, методология // М.: Наука, 1988.
15. *Ахо А., Хопкрофт Дж., Ульман Дж.* Построение и анализ вычислительных алгоритмов // М.: Мир, 1979.

16. *Hartwig A., Daske F. and Kobe S.* A recursive branch-and-bound algorithm for the exact ground state of Ising spin-glass models // *Comput. Phys. Commun.*– 1984.– Vol. 32.– pp. 133-138.
17. *Кофман А., Анри-Лабордер А.* Методы и модели исследования операций. Целочисленное программирование // М.: Мир.– Т.3.– 1977.
18. *Конвей Р.Б., Максвелл В.Л., Миллер Л.В.* Теория расписаний // М.: Наука, 1975.
19. *Klotz T., Kobe S.* "Valley structures" in the phase space of a finite 3D Ising spin glass with +I interactions // *J. Phys.*– 1994.– A27.– pp. 95-100.
20. *Klotz T., Kobe S.* Exact low-energy landscape and relaxation phenomena in Ising spin glasses // *Act. Phys. Slov.*– 1994.– Vol. 44.– pp. 347-356.
21. *Klotz T., Kobe S.* Cluster structure in the configuration space and relaxation in 3d +I Ising spin-glass models // *Hayashibara Forum '95. International Symposium on Coherent Approaches to Fluctuations, World Scientific.*– Singapore.– 1996.– pp. 192-195.
22. *Беллман Р.* Динамическое программирование // М.: ИЛ, 1960.
23. *Беллман Р., Дрейфус С.* Прикладные задачи динамического программирования // М.: Наука, 1965.
24. *Беллман Р., Энджел Э.* Динамическое программирование и уравнения в частных производных // М.: Мир, 1974.
25. *Основы динамического программирования.* // Минск: Изд-во Бел. ун-та, 1975.
26. *Simone C., Diehl M., Junger M., Mutzel P., Reinelt G. and Rinaldi G.* Exact ground states of Ising spin glasses: New experimental results with a branch and cut algorithm // *Stat. Phys.*– 1995.– Vol. 80.– pp. 487-496.
27. *Simone C., Diehl M., Junger M., Mutzel P., Reinelt G. and Rinaldi G.* Exact Ground States of Two-Dimensional +-J Ising Spin Glasses // *Stat. Phys.*– 1996.– Vol. 84.– pp. 1363-1371.

28. *Rieger H., Santen L., Blasum U., Diehl M., Junger M., Rinaldi G.* The critical exponents of the two-dimensional Ising spin glass revisited: Exact ground-state calculations, Monte Carlo simulations // *J. Phys.*– 1996.– A29.
29. *Palassini M. and Young A.P.* Trivial Ground State Structure in the Two-Dimensional Ising Spin Glass // *Phys. Rev.*– 1999.– B60.
30. *Гладков Л.А., Курейчик В.В., Курейчик В.М.* Генетические алгоритмы // 2-е издание.: М.: Физматлит, 2006.
31. *Еремеев А.В.* Разработка и анализ генетических и гибридных алгоритмов для решения задач дискретной оптимизации // Диссертация кандидата физико-математических наук.– Омск.– 2000.
32. *Aggarwal C.C., Orlin J.B., Tai R.P.* Optimized crossover for maximum independent set // *Oper. Res.*– 1997.– Vol. 45.– pp. 225-234.
33. *Balas E., Niehaus W.* Finding large cliques in arbitrary graphs by bipartite matching. Cliques, coloring, and satisfiability // *DIMACS Ser. Discrete Math. Theoretical Computer Science.*– 1996.– Vol. 26.– pp 29-49.
34. *Balas E., Niehaus W.* Optimized crossover-based genetic algorithms for the maximum cardinality and maximum weight clique problems // *Journal of Heuristics.*– 1998.– Vol. 4, №4.– pp. 107-122.
35. *Boese K.D., Kahng A.B., Muddu S.* A new adaptive multi-start technique for combinatorial global optimizations // *Operations Research Letters.*– 1994.– Vol. 16, №2.– pp. 101-114.
36. *Bremermann H.J., Roghson J., Salaff S.* Global properties of evolution processes. Natural automata and useful simulations // London: Macmillan.– 1966.– pp. 3-42.
37. *Eiben A.E., Raue P.E., Ruttkay Zs.* Genetic Algorithms with multiparent recombination. Parallel Problem Solving from Nature III // Berlin: Springer Verlag, (LNCS).– 1994.– Vol. 866.– pp. 78-87.
38. *Eremeev A.V.* A genetic algorithm with a non-binary representation for the set covering problem // *Operations Research Proceedings 1998.* Berlin: Springer Verlag.– 1999.– pp. 175-181.

39. *Хайкин С.* Нейронные сети. Полный курс // 2-е издание.: Пер. с англ.– М.: Издательский дом "Вильямс", 2006.
40. *Takefuji Y., Wang J.* Neural computing for optimization and combinatorics // Singapore: World Scientific, 1996.
41. *Галушкин А.И.* Теория нейронных сетей // М.: ИПРЖР, 2000.
42. *Головкин В.А.* Нейронные сети: обучение, организация и применение // М.: ИПРЖР, 2001.
43. *Уоссермен Ф.* Нейрокомпьютерная техника: Теория и практика // М.: Мир, 1992.
44. *McCulloch W.W., Pitts W.* A logical calculus of the ideas imminent in nervous activiti // Bulletin of Mathematical Biophysics 5: 115-33, 1943.
45. *Hebb D.O.* The Organization of Behavior // New York: Wiley, 1949.
46. *Розенблат Ф.* Принципы нейродинамики: перцептрон и теория механизмов мозга // М.: Мир, 1965.
47. *Minsky M., Papert S.* Perceptrons // Cambridge, MA: MIT Press, 1969.
48. *Hopfield J.J.* Neural networks and physical systems with emergent collective computational abilities // Proc.Nat.Acad.Sci.USA.– 1982.– Vol. 79.– pp. 2554-2588.
49. *Binder K., Young A.P.* Spin glasses: Experimental facts, theoretical concepts and open questions // Rev. Mod. Phys.– 1986.– Vol. 58.– issue 4.– pp. 801-976.
50. *Mezard M., Parisi G. and Virasoro M.A.* Spin Glass Theory and Beyond // World Scientific, Singapore, 1987.
51. *Fisher K.H. and Hertz J.A.* Spin Glasses // Cambridge University Press, Cambridge, 1993.
52. *Mydosh J.A.* Spin Glasses: an Experimental Introduction // Taylor and Francis, London, 1993.
53. *Amit D., Gutfreund H., Sompolinsky H.* Storing Infinite Numbers of Patterns in a Spin-Glass Model of Neural Networks // Phys. Rev. Lett.– 1985.– Vol. 55.– pp. 1530-1533.

54. *Gardner E.* The Space of Interactions in Neural Network Models // J. Phys. A.– 1988.– Vol. 21.– p. 257.
55. *Gardner E. and Derrida B.* Optimal Storage Properties of Neural Network Models // J. Phys. A.– 1988.– Vol. 21.– p. 271.
56. *Cover T.* Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition // IEEE Trans. Electr. Comp.– 1965.– Vol. 14.– p. 326.
57. *Krauth W., Mezard M.* Learning Algorithms with Optimal Stability in Neural Networks // J. Phys. A.– 1987.– Vol. 20.– p. 745.
58. *McEliece R.J., Posner E.C., Rodemich E.R., Venkatesh S.S.* The Capacity of the Hopfield Associative Memory // IEEE Trans. Inform. Theory.– 1987.– Vol. 33.– p. 461.
59. *Карандашев Я.М., Крыжановский Б.В., Литинский Л.Б.* Обобщение модели Хопфилда и статфизический подход: основное уравнение // Сборник трудов XIII Всероссийской научно-технической конференции «Нейроинформатика-2011».– М.: МИФИ, 2011.– ч.1.– сс. 79-88.
60. *Карандашев Я.М., Крыжановский Б.В., Литинский Л.Б.* Обобщенная модель Хопфилда и статфизический подход: основное уравнение // Сборник трудов XIII Всероссийской научно-технической конференции «Нейроинформатика-2011».– М.: МИФИ, 2011.– ч.3.– сс. 181-190.
61. *Kryzhanovsky B.V.* Expansion of a matrix in terms of external products of configuration vectors // Optical Memory and Neural Networks (Information Optics) .– 2008.– Vol. 17, № 1.– pp. 62–68.
62. *Amiit D., Gutfreund H., Sompolins H.* Spin-glass models of neural networks // Physical Review A.– 1986.– Vol. 32.– pp. 1007-1018.
63. *Крыжановский Б.В., Магомедов Б.М., Микаэлян А.Л.* Взаимосвязь глубины локального минимума и вероятности его обнаружения в обобщенной модели Хопфилда // Доклады Академии наук.– 2005.– Т.405, №3.– сс. 320-324.

64. *Frolov A.A., Mushinsky A.M., Tsodyks M.V.* Imitation model of associative memory as a neuron network with low activity level // *Biofizika.*– 1991.– Vol. 36.– pp. 339-343.
65. *Frolov A.A.* Limit informational characteristics of neural networks capable of associative learning based on Hebbian plasticity // *Neural Network World.*– 1991.– Vol. 1.– pp. 97-104.
66. *Frolov A.A., Muraviev I.P.* Informational characteristics of neural networks capable of associative learning based on Hebbian plasticity // *Network.*– 1993.– Vol. 4.– pp. 495-536.
67. *Frolov A.A., Husek D., Muraviev I.P.* Statistical neurodynamics of sparsely encoded Hopfield-like associative memory // *Neural Network World.*– 1996.– Vol. 6.– pp. 609-617.
68. *Frolov A.A., Husek D., Muraviev I.P.* Informational capacity and recall quality in sparsely encoded Hopfield-like neural network: Analytical approaches and computer simulation // *Neural Networks.*– 1997.– Vol. 10.– pp. 845-855.
69. *Frolov A.A., Husek D., Snasel V.* New computer algorithm enabling simulation of the recall process in sparsely encoded Hopfield-like neural network of an extremely large size // *Neural Network World.*– 1999.– Vol. 9.– pp. 509-515.
70. *Фролов А.А., Гусек Д., Муравьев И.П.* Информационная эффективность ассоциативной памяти типа Хопфилда с разреженным кодированием // *Информационные технологии и вычислительные системы.*– 2004.– №1.– сс. 44-67.
71. *Hopfield J.J. and Tank D.W.* Neural computation of decisions in optimization problems // *Biological Cybernetics.*– 1985.– Vol. 52.– pp. 144-152.
72. *Hopfield J.J. and Tank D.W.* Computing With Neural Circuits: A Model // *Science.*– 1986.– Vol. 233.– pp. 625-632.

73. *Reinelt G.* TSPLIB-A traveling salesman problem library // ORSA Journal on Computing.– 1991.– Vol. 3, № 4.– pp. 376-384.
74. <http://www.tsp.gatech.edu> (Traveling Salesman Problem)
75. *Wilson G.V., Pawley G.S.* On the stability of the traveling salesman problem algorithm of Hopfield and Tank // Biological Cybernetics.– 1988.– Vol. 58.– pp. 63-70.
76. *Brandt R.D., Wang Y., Laub A.J. and Mitra S.K.* Alternative networks for solving the traveling salesman problem and the list-matching problem // Proceedings of IEEE International Conference on Neural Networks, San Diego, CA.– 1988.– Vol. 2.– pp. 333-340.
77. *Aiyer S.V.B., Niranjan M. and Fallside F.* A theoretical investigation into the performance of the Hopfield model // IEEE Transactions on Neural Networks.– 1990.– Vol. 1.– pp. 204-215.
78. *Protzel P.W., Palumbo D.L. and Arras M.K.* Performance and fault-tolerance of neural networks for optimization // IEEE Transactions on Neural Networks.– 1993.– Vol. 4.– pp. 600-614.
79. *Крыжановский Б.В., Магомедов Б.М., Микаэлян А.Л.* Доменная модель нейронной сети // Доклады Академии Наук.– 2005.– Т.401, №4.– сс. 462-466.
80. *Крыжановский Б.В., Магомедов Б.М., Микаэлян А.Л.* Доменная модель как мультиагентная система // Искусственный интеллект.– 2005.– №4.– сс. 354-363.
81. *Kryzhanovsky B.V., Magomedov B.M.* Application of Domain Neural Network to Optimization Tasks // Proc. of the 15th International Conference On Artificial Neural Networks «ICANN-2005».– Warsaw, Poland.– Springer-Verlag.– ISSN: 0302-9743.– 2005.–Part. II.– pp. 397-403.
82. *Костенко В.А., Винокуров А.В.* Локально-оптимальные алгоритмы построения расписаний, основанные на использовании сетей Хопфилда // Программирование.– 2003.– № 4.

83. *Тарков М.С.* О вложении графов параллельных программ в графы распределенных вычислительных систем сетями Хопфилда // Сборник трудов XIII Всероссийской научно-технической конференции «Нейроинформатика-2011».– М.: МИФИ, 2011.– ч.3.– сс. 131-138.
84. *Feng G., Douligieris C.* The convergence and Parameter Relationship for Discrete-Time Continuous-State Hopfield networks // Proc. Of Intern. Joint Conference on neural networks.– 2001.– pp. 376-381.
85. *Крыжановский М.В., Мальсагов М.Ю.* Задача динамического распределения ресурсов в многопроцессорных системах // Сборник трудов IX Всероссийской научно-технической конференции «Нейроинформатика-2007».– М.: МИФИ, 2007.– сс. 229-234.
86. *Крыжановский М.В., Мальсагов М.Ю.* Нейросетевой подход к решению задачи распределения нагрузки в двухпроцессорной системе // Искусственный интеллект.– 2007.– №3.– сс. 547-555.
87. *Крыжановский М.В., Мальсагов М.Ю.* Задача распределения нагрузки в двухпроцессорной системе // Вестник компьютерных и информационных технологий.– 2008.– №3.– сс. 41-47.
88. *Kryzhanovsky M.V., Malsagov M.U.* Efficient multitasking in a dual-core CPU system // Optical Memory and Neural Networks (Information Optics).– 2008.– Vol. 17, №1.– pp. 84-91.
89. *Kryzhanovsky M.V., Malsagov M.U.* Distribution stream of tasks in dual-processor system // Proc. of the 18th International Conference on Artificial Neural Networks (ICANN-08).– Springer.– 2008.– LNCS-5164.– Part II.– pp. 150-158.
90. *Крыжановский М.В., Мальсагов М.Ю.* Решение задачи распределения нагрузки в двухпроцессорной системе на основе нейронных сетей // Искусственный интеллект.– 2008.– №3.– сс. 548-557.
91. *Kryzhanovsky M.V., Malsagov M.Yu.* Neuron Network Methods of Task Assignment in Multiprocessing System // Optical Memory and Neural Networks.– 2010.– Vol. 19, №3.– pp. 213–219.

92. *Карандашев Я.М., Крыжановский Б.В.* О трансформации энергетической поверхности в задаче бинарной оптимизации // Доклады академии наук.– 2009.– Т.429.– № 4.– сс. 465-469.
93. *Крыжановский Б.В., Литинский Л.Б.* Отыскание глобального минимума одного многоэкстремального функционала // Искусственный интеллект.– 2003.– №3.– сс. 116-120.
94. *Литинский Л.Б.* Нейросетевой подход к задаче минимизации квадратичного Функционала // Искусственный интеллект.– 2004.– №3.– сс. 542-549.
95. *Litinskii L., Magomedov B.* Global minimization of a quadratic functional: neural network approach // Pattern Recognition and Image Analysis.– 2004.–Vol. 15, №1.– pp. 80-82.
96. *Litinskii L.B.* Eigenvalue problem approach to discrete minimization. // Proc. of the 15th International Conference On Artificial Neural Networks «ICANN-2005».– Warsaw, Poland.– Springer-Verlag.– 2005.– pp. 405-410.
97. *Вылегжанин Д.В., Литинский Л.Б.* Компьютерная проверка нейросетевого алгоритма дискретной минимизации // Труды второй Всероссийской конференции МСО-2005.– М.: МГУ.– 2005.– сс. 590-596.
98. *Вылегжанин Д.В., Литинский Л.Б., Мурашкин А.А.* О выборе стартовых состояний при минимизации квадратичного дискретного функционала // Искусственный интеллект.– 2005.– №3.– сс. 334-342.
99. *Kirkpatrick S., Gelatt C.D., Vecchi M.P.* Optimization by simulated annealing // Science.– 1983.– Vol. 220.– pp. 671-680.
100. *Rutenbar R.A.* Simulated annealing algorithms: An overview // IEEE Circuits and Devices Magazine, 1989.
101. *Hinton G., Sejnowski T.* Learning and re-learning in boltzmann machines // In Parallel Distributed Processing: Explorations in The Microstructure of Cognition, MIT Press, Cambridge.– 1986.– chapter 7.– pp. 282-317.

102. *Tokuda I., Aihara K. and Nagashima T.* Adaptive annealing for chaotic optimization // *Physical Review E.*– 1998.– Vol. 58, №4.– pp. 5157-5160.
103. *Yamanaka K., Agu M., Miyajima T.* A continuous-time asynchronous Boltzman Machine // *Neural Networks.*– 1997.– Vol. 10, №6.– pp. 1103-1107.
104. *Glover F.* Tabu search: part I // *ORSA J. Comp.*– 1989.– Vol. 1.– pp. 190-206.
105. *Glover F.* Tabu search: part II // *ORSA J. Comp.*– 1990.– Vol. 2.– pp. 4-32.
106. *Glover F., Laguna. M.* Tabu search // Boston: Kluwer Acad. Publ., 1997.
107. *Glover F.(Ed.)* Tabu search methods for optimization // Feature Issue of *European J. Oper. Res.*– 1998.– Vol. 106, №2-3.
108. *Konishi J., Shimba S., Toyama J., Kudo M., Shimbo M.* Tabu search for solving optimization problems on Hopfield neural networks // *Proc. of Third International Conference "Knowledge-Based Intelligent Information Engineering Systems"*, 1999.
109. *Peng M., Gupta N.K., Armitage A.F.* An investigation into the improvement of local minima of the Hopfield network // *Neural Networks.*– 1996.– Vol. 9, №7.– pp. 1241-1253.
110. *Papageorgiou G., Likas A., Stafylopatis A.* Improved exploration in Hopfield network state-space through parameter perturbation driven by simulated annealing // *European Journal of Operational Research.*– 1998.– Vol. 108, №2.– pp. 283-292
111. *Stainbuch K.* Die Lernmatrix // *Kybernetik.*– 1961.– Vol. 1.– pp. 36-45.
112. *Willshaw D.J., Buneman O.P., Longuet-Higgins H.C.* Non-holographic associative memory // *Nature.*– 1969.– Vol. 222.– pp. 960-962.
113. *Palm G.* On the information storage capacity of local learning rules // *Neural Comp.*– 1992.– Vol. 4.– pp. 703-711.
114. *Widrow B. and Hoff Jr. M.E.* Adaptive switching circuits // *IRE Western Electric Show and Convention Record.*– 1960.– Part 4.– pp. 96-104.

115. *Sompolinsky H.* Neural Networks with nonlinear synapses and static noise // Phys. Rev. A.– 1986.– Vol. 34.– p. 2571.
116. *van Hemmen J.L.* Nonlinear neural networks near saturation // Phys. Rev. A.– 1987.– Vol. 36.– p. 1959.
117. *Kintzel W., Oppen M.* Models of Neural Networks I / Eds. E. Domany, J.L. van Hemmen, K. Schulten // Physics of Neural Networks, Springer.– 1995.– p. 170.
118. *Алиева Д.И., Крыжановский В.М.* Модель Хопфилда малых размеров с клипированными связями //Искусственный интеллект.– 2006.– №3.– сс. 240-248,.
119. *Крыжановский В.М., Симкина Д.И.* Клипирование модели Хопфилда малых размеров //Вестник компьютерных и информационных технологий.– 2007.– №10.– сс. 27-31,.
120. *Крыжановский Б.В., Крыжановский В.М., Микаэлян А.Л.* Применение процедуры клипирования в задачах бинарной минимизации квадратичного функционала //Доклады Академии Наук.– 2007.– Т.413, №6.– сс. 730-733,.
121. *Крыжановский В.М.* Исследование векторных нейронных сетей с бинаризованными синаптическими коэффициентами для задач обработки информации и принятия решения. // Диссертация на соискание ученой степени кандидата физико-математических наук, 2010.