

2. Посыпкин М.А. Архитектура и программная организация библиотеки для решения задач дискретной оптимизации методом ветвей и границ на многопроцессорных вычислительных комплексах: тр. ИСА РАН. 2006. Т. 25. С. 18–25.

3. Афанасьев А.П., Посыпкин М.А., Сигал И.Х. Проект BNB-Grid: решение задач глобальной оптимизации в распределенной среде: тр. Второй междунар. конф. // Системный анализ и информационные технологии (САИТ-2007). 2007. Т. 2. С. 177–181.

4. Evtushenko Y., Posypkin M., Sigal I. A framework for parallel large-scale global optimization // Computer Science – Research and Development. 2009. № 23 (3), pp. 211–215.

5. Посыпкин М.А. Методы и распределенная программная инфраструктура для численного решения задачи поиска молекулярных кластеров с минимальной энергией // Параллельные вычислительные технологии (ПаВТ'2009): тр. Междунар. науч. конф. (Н. Новгород, 30 марта – 3 апреля 2009 г.). Челябинск: Изд-во ЮУрГУ, 2009. С. 269–281.

СИНЕРГЕТИЧЕСКИЙ ПОДХОД К ОРГАНИЗАЦИИ СТРУКТУР ДАННЫХ

В.В. Дрождин, к.т.н.; А.М. Володин

(Пензенский государственный педагогический университет им. В.Г. Белинского, drozhdin@spu-penza.ru)

На основе синергетического подхода определены организация и поведение структур данных, определены вещественная, энергетическая и информационная составляющие структуры данных и процессов ее взаимодействия с внешней средой. Все структуры данных классифицированы на регулярные, квазирегулярные и нерегулярные. Указано, что самоорганизация наиболее ярко проявляется в нерегулярных структурах данных. Приведены условия возникновения самоорганизации и показан пример возможного пути эволюции структуры данных в поле структур данных.

Ключевые слова: самоорганизующаяся система, структура данных (СД), критерии оценки СД, виды СД (логическая, физическая, регулярная, квазирегулярная, нерегулярная), поле СД, структура-аттрактор, эволюция и популяция СД.

При разработке программного обеспечения большую роль играет выбор *структур данных* (СД). От этого зависят сложность реализации и эффективность выполнения программ. У различных видов СД свои особенности. Некоторые из них имеют узкую специализацию, то есть пригодны для решения только определенных задач.

СД – это система, состоящая из совокупности элементов данных и отношений между ними [1].

Элементами СД могут быть как простые, или атомарные данные, состоящие из одного элемента, так и сложные СД.

Структурные отношения задают устойчивые (структурные) взаимосвязи между элементами. В качестве таких взаимосвязей могут выступать отношения эквивалентности, порядка, иерархии, функционального соответствия и др. Отношения (связи) между элементами позволяют получить доступ от одних элементов к другим.

Выделяют логические и физические СД. *Логические СД* носят абстрактный характер (не учитывается их реализация на компьютере), хорошо формализованы и, как правило, имеют точное математическое описание. Они ориентированы на отражение объектов реального мира.

Физические СД предназначены для реализации логических СД и определяют, каким образом хранятся данные, как осуществляется доступ к ним и как реализуются операции их обработки. Физические СД ориентированы прежде всего на эффективную обработку данных.

Различают логическую и физическую организацию данных физической СД [2]. *Логическая организация данных* физической СД отражает мно-

жество элементов данных и отношения между ними, а также способ реализации операций обработки данных. *Физическая организация данных* физической СД определяет представление и размещение данных в памяти, метод доступа к ним и механизм их обработки. Кроме того, физическая организация данных может учитывать организацию и различные характеристики памяти, а также включать дополнительные (управляющие) подструктуры, обеспечивающие повышение эффективности обработки данных.

СД должны удовлетворять критериям корректности, надежности (устойчивости) и эффективности.

Корректность означает непротиворечивость и целостность данных, между которыми существуют сильные (логические) отношения. Для обеспечения корректности СД должна содержать методы валидации. Поэтому, если операция обработки данных может привести к нарушению корректности данных, то либо она запрещается, либо вызывается метод, восстанавливающий непротиворечивость и целостность данных.

Логическая и физическая организации данных физической СД могут подвергаться реорганизации, однако такие изменения не должны затрагивать логический уровень внешнего представления СД, на котором она должна оставаться неизменной. Разнообразие физической организации данных для одной логической организации данных не влияет на логику обработки физической СД, а изменяет только эффективность обработки данных.

Надежность – это способность СД выполнять необходимые операции обработки данных с тре-

буемым качеством в течение определенного времени.

Для обеспечения надежности СД должна быть устойчивой к сбоям оборудования и потере данных. При нарушении корректности данных вследствие их недостаточной надежности должна запускаться процедура восстановления данных. Для обеспечения такой возможности часто используется дублирование части данных.

Еще одним критерием выбора СД является эффективность выполнения операций обработки данных. Для этого наиболее часто используются поиск, добавление, удаление и модификация данных. Критерий эффективности начинает играть решающую роль в случае высокой интенсивности операций обработки данных или потребности в значительных затратах ресурсов (времени, памяти и др.).

При синергетическом подходе СД рассматривается как открытая автономная система, существующая в определенной среде и обменивающаяся с ней *веществом, энергией и информацией*. Таким образом, СД становится не только логической (формальной), но и конструктивной (реально существующей, функциональной) системой, приобретающей свойство активности, то есть функционирующей преимущественно вследствие внутренних потребностей, а не под влиянием внешних воздействий.

В качестве внешней среды выступает программа, в которой реализована и используется СД. От внешней среды к СД поступают запросы на обработку данных, которые СД получает, интерпретирует и исполняет с помощью соответствующих методов доступа и модификации данных. В результате обработки данных СД выдает ответ в виде потока данных или сообщения о результате выполнения запроса.

Вещество, из которого строится СД и которым она обменивается с внешней средой, – это последовательность байтов. Действительно, для представления СД в имеющихся технических средствах, а также запроса и ответа на него необходимо сформировать определенную структуру (последовательность байтов), которая размещается в памяти компьютера.

Энергия – это способность запроса, поступившего от внешней среды, произвести в СД некоторые изменения (например, модифицировать элементы данных) или заставить СД сформировать сообщение о своем состоянии, а также способность запросов, исходящих от СД, произвести некоторые изменения во внешней среде или получить информацию о ее состоянии.

Все запросы условно можно разделить на высокоэнергетические, требующие больших изменений и, следовательно, значительных затрат ресурсов, и низкоэнергетические, практически не требующие затрат ресурсов. Поэтому для обеспе-

чения высокой интенсивности обработки данных целесообразна такая организация СД, при которой все поступающие запросы воспринимаются ею как низкоэнергетические, а сложная обработка данных производится в свободное от работы время.

Информация рассматривается так, как обычно в физическом мире, и выполняет две функции: является отражением внешнего мира (выражает в запросе требование на обработку данных) или отражает внутреннюю структуру объекта (описывает структуру запроса).

Обмен информацией – это обмен данными, содержащимися в запросе (например, критерий поиска) и ответе (запрашиваемые данные).

Взаимодействие (обмен веществом, энергией и информацией) программных компонентов и СД часто носит случайный, нерегулярный характер. Поэтому в процессе длительного использования СД интенсивность различных операций обработки данных может существенно меняться.

Для адаптации к изменяющимся условиям и для повышения эффективности функционирования СД может содержать дополнительную (служебную или управляющую) информацию, которая не будет видна извне, однако может играть существенную роль при обработке запросов.

Дополнительная информация может включать признаки, информирующие одни элементы о свойствах других, и параметры, позволяющие настраивать СД на требуемое использование и обработку данных.

Признак – это некоторое свойство, которым обладают или не обладают структура, подструктура или элемент данных. Примерами таких свойств являются: наличие или отсутствие дубликатов среди элементов данных структуры, способ упорядочения элементов, количество элементов (кардинальность), диапазон значений, четность и нечетность значения элемента данных, соответствие элемента некоторой функции, символ завершения последовательности и др.

Признаки выполняют информирующую роль, то есть предоставляют информацию об определенных свойствах (аспектах) СД и ее элементов. Они позволяют повысить надежность и эффективность обработки СД.

Параметры – это изменяемые элементы структуры, выполняющие роль настроек и после назначения служащие ограничениями. Параметрами являются допустимое (максимальное и минимальное) количество элементов в структуре или подструктуре, максимальный и минимальный размер одного элемента данных, логическое ограничение (предикат, условие) на значения элементов, обязательность и необязательность наличия элементов, допустимость дубликатов и т.д.

В зависимости от наличия или отсутствия дополнительной информации все СД можно разде-

лить на структуры, не содержащие и содержащие дополнительную информацию.

Структуры, не содержащие дополнительную информацию, – это регулярные СД, которые строятся по определенным правилам. Все классические СД (множества, последовательности, деревья и др.) являются регулярными. Регулярные СД преимущественно ориентированы на равновероятный доступ к любому элементу данных. Однако доступ к различным элементам данных в СД часто происходит с разной интенсивностью. Не все данные одинаково востребованы и равнозначны, поэтому использование регулярных структур на практике не всегда целесообразно.

Структуры, содержащие дополнительную информацию, можно разделить на квазирегулярные и нерегулярные.

Квазирегулярные структуры – это СД, в которых дополнительная информация используется только в определенных ситуациях и на логику обработки структуры не оказывает существенного влияния. Пример квазирегулярных структур – СД, используемые системой управления БД для хранения данных в таблицах и индексах. Дополнительной информацией здесь являются кардинальность (количество элементов в таблице или индексе), количество занятых и свободных страниц (блоков), наибольшее и наименьшее значения, количество листовых страниц и уровней в индексе и т.д.

Нерегулярные структуры – это СД, содержащие дополнительную информацию, существенно влияющую на логику обработки структуры. Нерегулярные СД строятся из гетерогенных (неоднородных) подструктур и элементов с разной организацией и дисциплиной обслуживания, которые могут быть по-разному упорядочены (частично упорядочены или не упорядочены), содержать или не содержать дубликаты и т.д. Дополнительная информация в нерегулярных СД отражает неоднородность подструктур и элементов.

С точки зрения синергетики [3] регулярные СД можно отнести к максимально организованным системам с линейными законами функционирования. Такие структуры ненадежны и эффективны в очень узких пределах, а на их поддержание требуются огромные затраты ресурсов. Ненадежность регулярных СД можно продемонстрировать следующими примерами: при добавлении в бинарное дерево узла, содержащего три элемента, дерево перестает быть бинарным; если в упорядоченную последовательность добавить элемент, нарушающий порядок, последовательность перестанет быть упорядоченной.

Самоорганизация в регулярных СД невозможна. Предельная организованность подавляет эволюционные возможности СД и при быстром изменении условий среды приводит к их деградации и гибели.

Квазирегулярные СД близки по организации и функционированию к регулярным СД, поэтому их эволюция прямо пропорциональна времени [4].

Нерегулярные СД эквивалентны системам с нелинейными законами функционирования (системам с сильной нелинейностью), поэтому при определенных условиях в таких структурах может возникнуть самоорганизация. В результате самоорганизации будут порождаться СД, наиболее эффективные в конкретных ситуациях. Такие структуры достаточно быстро обрабатывают поступающие извне запросы в критических ситуациях (цейтнот времени и недостаточный объем ресурсов).

В силу недостаточности знаний и изменчивости среды, в которой будет использоваться СД, невозможно разработать структуру, эффективную во всех требуемых ситуациях. Поэтому свойства СД, ее логическая и физическая организации должны изменяться, чтобы максимально полно удовлетворять требованиям среды. Следовательно, СД должна эволюционировать.

Эволюция СД представляет собой необратимую последовательность допустимых изменений структуры во времени. Допустимость означает, что все изменения СД осуществляются в многомерном фазовом пространстве с учетом текущего состояния СД.

Фазовое пространство всех возможных структур данных будем называть *полем СД* (ПСД), на осях которого откладываются способы организации данных и методы их реализации, значения признаков и параметров СД, методы обработки данных и дисциплины обслуживания и т.д. ПСД является дискретной средой, каждая точка которой соответствует некоторой конкретной СД. Эволюция СД в ПСД представляется непрерывно идущей фазовой траекторией, начинающейся в точке, соответствующей начальному состоянию, и заканчивающейся в точке, соответствующей конечному состоянию СД. Последовательность точек в фазовой траектории отражает последовательные изменения СД в результате выполнения операций обработки данных.

С другой стороны, ПСД не является однородным, то есть СД, соответствующие различным точкам ПСД, неравнозначны. Неравнозначность проявляется в сложности и эффективности реализации и обработки СД в различных условиях.

В процессе эволюции возникают достаточно устойчивые структуры – *структуры-аттракторы* (СА), характеризующие развитые (установившиеся) стадии эволюции СД и «притягивающие» к себе траекторию изменения СД. СА – это оптимальные для определенных условий обработки данных СД, имеющие минимальную сложность, что позволяет системе быстро и с минимальными затратами обрабатывать поступающие извне запросы.

В качестве примера СА можно привести списковые структуры (последовательности), деревья, хешированные структуры и др. Список является эффективной структурой при небольшом объеме данных и преимущественном выполнении операции последовательного просмотра элементов, а иерархические структуры эффективно используются при достаточно больших объемах данных и одинаковой интенсивности их использования. При значительных объемах данных и необходимости произвольного доступа к ним (по ключу) оптимальными будут хеш-таблицы. Однако они, в отличие от деревьев, не позволяют эффективно просматривать данные в порядке их возрастания или убывания.

Можно утверждать, что СА являются все регулярные СД, так как каждая регулярная СД оптимальна для определенных условий обработки данных и имеет минимальную сложность вследствие регулярности построения и отсутствия дополнительной информации. Среди квазирегулярных и нерегулярных СД также могут возникать СА, однако эти аттракторы будут иметь более высокий порядок, то есть являться композицией аттракторов, соответствующих регулярным СД.

Каждая СА имеет область притяжения – набор СД, несущественно отличающихся от СА. Причем чем меньше СД отличается от СА, тем ближе она расположена в ПСД к СА и тем больше проявляет свойства СА.

Текущее состояние реальной СД может отличаться от СА, то есть реальная СД соответствует СД в ПСД, расположенной на некотором расстоянии от СА. Но, чем четче будут проявляться условия аттрактора, тем сильнее СД будет стремиться к СА, проходя определенную траекторию в ПСД от текущего состояния СД к СА, сопровождаемую упрощением СД и повышением эффективности реализации операций, оптимальных для СА. Таким образом, попав в область притяжения аттрактора, СД неизбежно будет эволюционировать к СА.

Находясь в области притяжения СА, поведение СД определяется законами функционирования, свойственными СА. В этой области СД способна гасить флуктуации – возмущающие воздействия внешней среды и внутренние отклонения в самой системе. Поэтому, хотя СД и фиксирует с помощью дополнительной информации различные особенности поступающих запросов и связанной с ними обработки данных, что незначительно модифицирует поведение СД в различных ситуациях, основное поведение СД остается постоянным. Это свидетельствует о нечувствительности СД к небольшим изменениям условий функционирования внутри области притяжения СА, особенно в центре (вблизи СА).

Однако флуктуации приводят к увеличению энтропии в СД, выражающейся в постепенном на-

коплении зафиксированных отклонений, противоречащих законам функционирования СА, в области притяжения которой находится СД. При этом некоторые операции обработки данных могут приводить к тому, что структурные отношения и логические ограничения на элементы структуры (например, условие сбалансированности деревьев) могут нарушаться, а выполнение процедур их восстановления во время обработки данных может оказаться невозможным вследствие высокой временной сложности.

В результате нарушается симметрия (законы функционирования), индуцируемая аттрактором. Например, красно-черные и AVL-деревья постепенно будут становиться несбалансированными, а в хеш-таблицах возрастает число коллизии, что приводит к увеличению длины цепочек и вырождению структуры в мультисписок.

Для адаптации СД к изменениям внешней среды без усложнения ее структуры целесообразно использовать модификаторы. Модификатор – это нечто, дополнительно включаемое в СД с целью изменения существующих и придания ей новых свойств. В качестве модификаторов могут выступать дополнительные элементы данных (ключевые элементы, дубликаты и др.), указатели на некоторые выделенные элементы (например, наиболее часто используемые элементы или элементы с определенными свойствами), признаки (наличие или отсутствие элементов с определенными свойствами, позиция первого и последнего элементов и др.) и параметры (число элементов в блоках в зависимости от интенсивности их использования, объем дополнительной памяти и т.д.), а также методы обработки данных, реализующие новые операции и дисциплины обслуживания. Модификаторы незначительно усложняют СД, но существенно повышают ее устойчивость, надежность и эффективность. Например, при удалении некоторого элемента он лишь помечается как удаленный, что позволяет избежать реорганизации СД в процессе выполнения операции удаления данных и существенно повысит ее эффективность.

Со временем наступает момент, когда отклонение СД от СА достигнет некоторого критического значения. При этом вследствие накопления большого количества отклонений (зафиксированных флуктуаций) или сильных возмущающих воздействий внешней среды (существенных изменений интенсивности и типов запросов, поступающих в СД) СД будет находиться на границе области притяжения СА и являться громоздкой и неэффективной. Это состояние неустойчиво, так как СД попадает под влияние нескольких СА (бифуркация). Например, может образоваться гибридная структура, содержащая линейные (на предшествующий и последующий элементы) и нелинейные (иерархические) указатели. Поэтому СД будет проявлять свойства последовательностей и де-

ревьев. Но в состоянии бифуркации СД не может находиться долго, так как любые флуктуации (преобладание запросов определенного типа или определенных операций обработки данных) запустят достаточно мощный процесс оптимизации СД, который автоматически перебросит ее в область притяжения другой СА и подчинит ее существенное поведение законам функционирования новой СА.

Формирование новой СД сопровождается качественным скачком (фазовым переходом), в результате которого резко изменяются конструктивные элементы СД, особенно модификаторы. Полученная СД будет обладать меньшей энтропией.

Таким образом, трансформация СД происходит по следующей схеме: функционирование в области притяжения исходной СА → отклонения (флуктуации) → бифуркация → реорганизация (фазовый переход) → функционирование в области притяжения новой СА.

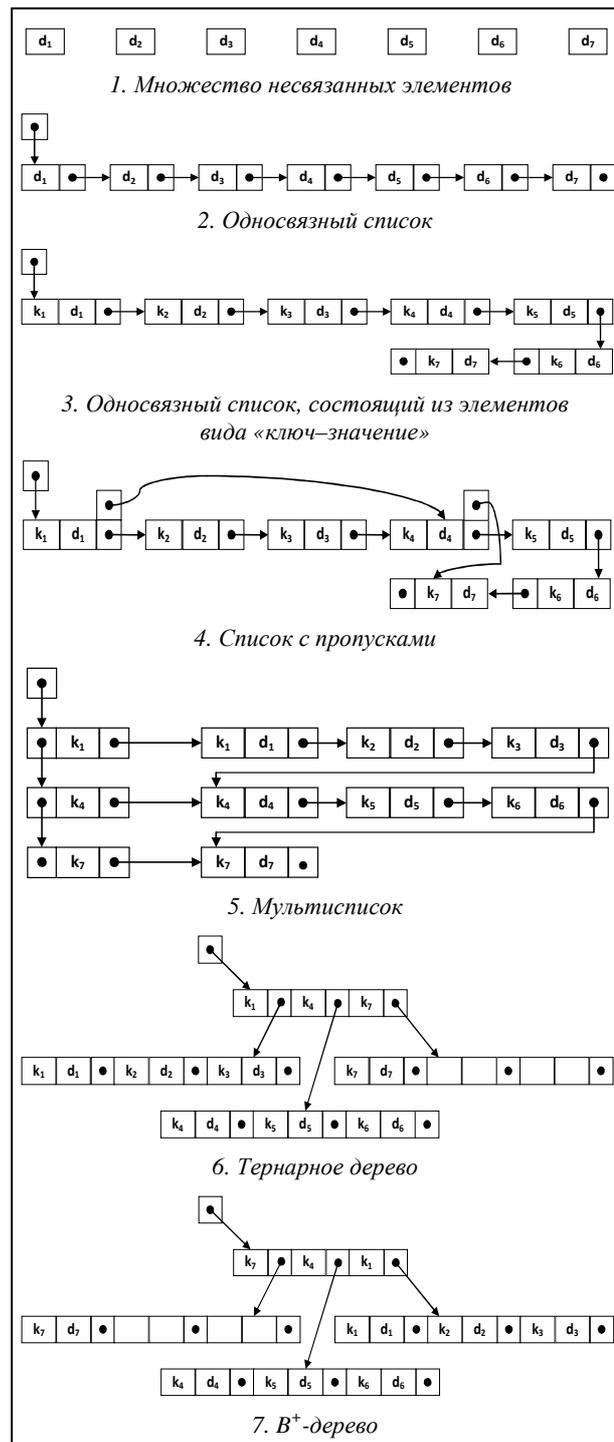
На рисунке приведена эволюция СД, сопровождающаяся изменением числа и типов связей между элементами: множество несвязанных элементов → односвязный список → односвязный список с доступом к элементам по ключу → список с пропусками → мультисписок → тернарное дерево → V^+ -дерево.

Сформировавшаяся в результате последовательности фазовых переходов СД является оптимальной в более широких пределах для операций прямого и последовательного доступа к элементам данных, чем исходная и промежуточные СД. При других условиях будет порождена другая последовательность структур, в результате которой сформируется оптимальная СД. Таким образом, из одной простейшей СД (множество элементов) может быть порожден широкий спектр СД, обеспечивающих эффективную обработку данных в различных условиях. Причем переход от одной СД к другой потребует минимальных затрат времени и ресурсов, что позволит системе эволюционировать в реальном времени и быть всегда готовой для обработки данных.

Однако в процессе эволюции СД может возникнуть ситуация, когда при реализации последовательности запросов потребуются многократное преобразование СД путем перехода от одной СА к другой и назад, то есть СД будет колебаться между двумя или небольшим числом СА. Это означает, что СД находится в неустойчивом состоянии из-за частого выполнения двух или нескольких наборов операций, сильно отличающихся способами обработки данных. Поэтому, кроме затрат ресурсов, необходимых для реализации операций обработки данных, будут требоваться большие затраты на эволюцию СД, что существенно снижает ее эффективность.

Для устранения такой ситуации система должна предпринять усилия по поиску и разра-

ботке новой СА, обеспечивающей эффективную реализацию всех необходимых операций обработки данных, а в случае отсутствия такой СА или ее высокой сложности перейти на новый уровень организации – уровень популяций изомерных СД, имеющих одинаковый состав, но различную структуру. Переход на уровень популяций СД позволяет сохранять их относительно невысокую сложность и обеспечивать очень высокую эффективность обработки данных в различных условиях.



Для управления эволюцией СД необходимо выделить параметры порядка (например, частоту и скорость выполнения операций обработки данных). Изменяя эти параметры, можно резонансно возбуждать СД, близкие к СА эволюции с требуемыми свойствами.

Таким образом, предложен подход к организации структур данных, позволяющий программным системам формировать СД, обеспечивающие эффективную обработку данных в различных условиях и требующие минимальных затрат на их создание и реорганизацию.

Литература

1. Дрождин В.В. Системный подход к построению модели данных эволюционных баз данных // Программные продукты и системы. 2007. № 3. С. 52–55.
2. Дрождин В.В., Володин А.М. Методы организации данных в современных системах управления данными // Изв. Пензенского гос. ун-та. 2008. № 8 (12). С. 103–106 (Физико-математические и технические науки).
3. Малинецкий Г.Г. Математические основы синергетики: хаос, структуры, вычислительный эксперимент. М.: Изд-во ЛКИ, 2007. 312 с.
4. Хакен Г. Информация и самоорганизация. Макроскопический подход к сложным системам. Сер.: Синергетика: от прошлого к будущему. М.: КомКнига, 2005. 248 с.

ЧИСЛЕННО-АНАЛИТИЧЕСКИЕ ИССЛЕДОВАНИЯ ЭВОЛЮЦИИ ДИСКРЕТНЫХ СТРУКТУР

А.Н. Белов; В.В. Масюков, к.ф.-м.н.
(Тверской государственный университет, abelov@tversu.ru)

Предложены и аналитически исследованы некоторые алгоритмы моделирования эволюции дискретных структур, на их основе разработана программа EVOLDIST. Результаты численного моделирования позволяют сделать вывод о перспективности использования развитых подходов для исследования сложных нелинейных природных явлений.

Ключевые слова: клеточные автоматы, дискретные структуры, нелинейные явления.

Известно, что при моделировании некоторых нелинейных явлений получить удовлетворительные результаты, хотя бы качественно адекватные наблюдаемым эффектам, часто удается только в рамках специфических подходов к численному моделированию, основанных на исследовании эволюции дискретных структур [1, 2]. Например, в рамках модели клеточного автомата удается получить решения типа спиральных волн для реакционно-диффузионных уравнений и другие нетривиальные нелинейные эффекты [2]. Поскольку вычислительная сложность сильно зависит от объема дискретной структуры (числа частиц), до настоящего времени в основном рассматривались системы с локальным характером взаимодействия (то есть в предположении о том, что изменение структуры в некоторой точке зависит только от соседей, находящихся в некоторой окрестности этой точки). Целью данной работы являются построение и изучение нелокальных алгоритмов моделирования эволюции дискретных структур и описание их реализации в программе EVOLDIST (*Evolution of Discrete Structures*).

Математический формализм и постановка задач

Дискретной структурой будем считать область $D \subset \mathbb{R}^n$, $n=1, 2, 3$, покрытую равномерной кубической сеткой с шагом h , состоящей из N узлов, в каждом узле r_i сетки находится шар радиусом $h/2$. Каждый шар характеризуется его принадлежно-

стью к одному из S сортов. В дальнейшем конфигурацией

$$\mathbf{X}^T = (\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_N) \quad (1)$$

будем называть один из вариантов взаимного расположения шаров, где \mathbf{q}_i – сорт шара в i -м узле, $i=1, \dots, N$. Полной энергией системы будем считать

$$\mathbf{E}(\mathbf{X}) = \mathbf{E}^{(in)} + \mathbf{E}^{(ex)} = \sum_{ij} \mathbf{E}_{ij} + \mathbf{E}^{(ex)}, \quad (2)$$

где $\mathbf{E}^{(in)}$ – потенциальная энергия взаимодействия между шарами; $\mathbf{E}_{ij} = \mathbf{E}(\mathbf{q}_i, \mathbf{q}_j, \mathbf{r}_{ij})$ – энергия шара в узле i , находящегося в поле шара узла j на расстоянии \mathbf{r}_{ij} ; $\mathbf{E}^{(ex)} = \sum_{i=1}^N \mathbf{E}_i^{(ex)} = \sum_{i=1}^N \Phi^{(ex)}(\mathbf{r}_i) \mathbf{q}_i$ – энергия системы во внешнем силовом поле. В дальнейшем будем рассматривать случай, когда энергия $\mathbf{E}^{(in)}$ является квадратичной формой \mathbf{q}_i . В матричном виде

$$\mathbf{E}(\mathbf{X}) = \mathbf{X}^T \boldsymbol{\varepsilon} \mathbf{X} + \mathbf{E}^{(ex)}, \quad (3)$$

где $\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}(\mathbf{r}_{ij})$ – симметричная матрица квадратичной формы, определяемая видом взаимодействия. При этом считаем, что $\mathbf{E}_{ii} = \mathbf{0}$, $\boldsymbol{\varepsilon}(\mathbf{r}_{ii}) = \mathbf{0}$. Заметим, если $\mathbf{E}_{ij}(\mathbf{r}) = \Theta(\mathbf{d} - \mathbf{r}_{ij}) \mathbf{f}(\mathbf{r}_{ij})$, где $\Theta(\mathbf{z})$ – функция Хевисайда, получается локальное взаимодействие с характерным расстоянием \mathbf{d} . При значениях \mathbf{d} порядка шага сетки h получаются модели клеточных автоматов.

Назовем единичной перестановкой обмен местами двух шаров разных сортов. Из конфигурации