

На правах рукописи



Фарков Михаил Александрович

Разработка алгоритмов выполнения молекулярного докинга с использованием графических процессоров

05.13.17 — Теоретические основы информатики

АВТОРЕФЕРАТ

диссертации на соискание учёной степени

кандидата технических наук

Красноярск 2017

Работа выполнена в Федеральном государственном автономном образовательном учреждении высшего образования «Сибирский федеральный университет»

Научный руководитель: доктор технических наук, профессор Легалов Александр Иванович

Официальные оппоненты: Кузнецов Александр Алексеевич, доктор физико-математических наук, доцент, ФГБОУ ВО «Сибирский государственный аэрокосмический университет имени академика М.Ф. Решетнева», кафедра прикладной математики, профессор;

Снытников Алексей Владимирович, кандидат физико-математических наук, Институт вычислительной математики и математической геофизики Сибирского отделения РАН, лаборатория параллельных алгоритмов решения больших задач, научный сотрудник

Ведущая организация: Федеральное государственное бюджетное научное учреждение "Федеральный исследовательский центр "Красноярский научный центр Сибирского отделения Российской академии наук", г. Красноярск

Защита состоится «07» апреля 2017 г. в 13.00 часов в ауд. УЛК 112 на заседании диссертационного совета Д 212.099.22 на базе Сибирского федерального университета по адресу: 660074, г. Красноярск, ул. Киренского, 26.

С диссертацией можно ознакомиться в библиотеке и на сайте Сибирского федерального университета по адресу <http://www.sfu-kras.ru>

Автореферат разослан «__» февраля 2017 г.

Учёный секретарь
диссертационного совета:

Покидышева Людмила Ивановна



ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность исследования.

Широкий спектр существующих прикладных задач требует применения сложных вычислительных алгоритмов и методов, предъявляющих высокие требования к вычислительным ресурсам. Для их эффективной реализации используются высокопроизводительные вычислительные системы, в том числе гетерогенные, включающие в себя специализированные процессоры, например, графические. Перенос существующих решений для центрального процессора на гетерогенные вычислительные системы без изменений невозможен в силу значительных различий в программно-аппаратной архитектуре. Поэтому актуальной является задача подбора подходящих алгоритмов и их преобразования с учётом особенностей информационных процессов, протекающих в высокопроизводительных вычислительных системах, и специфики прикладной задачи.

Одной из таких прикладных задач, имеющих важное народно-хозяйственное значение, является выполнение молекулярного докинга, основной областью применения которого является разработка лекарственных препаратов. Молекулярный докинг – это компьютерное моделирование взаимодействия молекул. Основной целью выполнения молекулярного докинга является поиск реалистичных конформаций молекулярных соединений на основе количественной оценки оптимальности их взаимного пространственного расположения, структурной комплементарности и энергии связывания. Подбор перспективных химических соединений (испытания *in vitro*) для проведения клинических испытаний (испытания *in vivo*) может занимать до трёх лет и является чрезвычайно ресурсоёмким этапом, так как требуется отобрать небольшое количество перспективных химических соединений (кандидатов в лекарства) из набора, состоящего из нескольких миллионов веществ, что требует существенных финансовых затрат на реагенты для проведения химических реакций, а также существенных временных затрат на проведение каждого теста. Именно для этого этапа перспективным является использование компьютерных моделей (*in silico*), что позволяет значительным образом снизить количество химических реакций *in vitro*, вследствие чего происходит снижение как финансовых, так и временных затрат. Внедрение перспективных параллельных архитектур позволит эффективнее и быстрее выполнять молекулярный докинг. Следует отметить, что разработка методов, алгоритмов и подходов для выполнения молекулярного докинга существенно отстаёт от развития современных перспективных параллельных архитектур. Так, по состоянию на 2016 год лишь несколько программ для молекулярного докинга используют возможности многоядерных центральных процессоров; единицы используют возможности кластеров и массивно параллельных систем (*MPP*), а те, что используют, зачастую реализованы на устаревших библиотеках. Возможности гетерогенных вычислительных систем применяются ещё реже. Существует всего лишь несколько реализаций молекулярного докинга для

графических процессоров, ориентированных на лиганд-белковое взаимодействие, которые либо не принесли существенного ускорения в процесс выполнения докинга, либо недостаточно полно используют ресурсы графических процессоров. Актуальной является задача разработки алгоритмов молекулярного докинга для высокопроизводительных вычислительных систем с использованием графических процессоров.

В основе подавляющего большинства методов лиганд-белкового докинга лежат методы численной оптимизации. Разработка новых методов, алгоритмов и программного обеспечения для численных методов оптимизации, ориентированных на использование ресурсов графических процессоров, позволит повысить эффективность выполнения лиганд-белкового докинга, а также решения широкого круга других прикладных задач фармакологии, биоинформатики, математической физики, экономики и ядерной физики, требующих интенсивного применения численных методов оптимизации в процессе вычислений.

Целью работы является разработка алгоритмов реализации численных методов оптимизации на базе гетерогенных вычислительных систем, использующих графические процессоры, и решение с их помощью задачи молекулярного лиганд-белкового докинга.

Для достижения поставленной цели необходимо решить следующие **задачи**:

1. Провести анализ существующих методов и алгоритмов решения задачи молекулярного докинга и возможностей эффективной реализации этих методов на современных высокопроизводительных гетерогенных системах.

2. На основе проведенного анализа предложить модели и алгоритмы, адаптированные для эффективной реализации выбранных методов на графических процессорах.

3. Используя предложенные модели и алгоритмы, разработать программы, обеспечивающие решение поставленной задачи молекулярного лиганд-белкового докинга.

4. Провести анализ эффективности разработанного программного обеспечения на примерах рассматриваемой предметной области.

Объектом исследования являются информационные процессы, протекающие в гетерогенных вычислительных системах, использующих графические процессоры, а также численные методы и алгоритмы решения задачи молекулярного докинга.

Предметом исследования является эффективная реализация численных методов оптимизации с учетом особенностей взаимодействия информационных процессов, протекающих в гетерогенных вычислительных системах, использующих графические процессоры.

Научные положения, выносимые на защиту.

1. Алгоритм реализации метода дифференциальной эволюции, учитывающий особенности графических процессоров.

2. Алгоритм расчёта силовых полей межмолекулярного взаимодействия на основе вычисления сеток, адаптированный для графических процессоров.

3. Алгоритм решения задачи молекулярного докинга с использованием графических процессоров.

Научная новизна представленных результатов.

1. Предложен алгоритм реализации метода дифференциальной эволюции на графических процессорах. В отличие от существующих подходов алгоритм использует одноблочную параллельную декомпозицию процедуры оптимизации и выполняет все основные этапы метода дифференциальной эволюции без управления со стороны центрального процессора. Это позволило увеличить количество одновременно выполняемых процедур оптимизации на графическом процессоре и автоматически управлять их выполнением за счёт учёта особенностей обрабатываемых данных, архитектуры графического процессора и метода вычислений.

2. Предложен алгоритм вычисления сеток межмолекулярного взаимодействия с использованием графического процессора. В отличие от существующих подходов алгоритм унифицировано выполняет вычисление для различных компонент энергии межмолекулярного взаимодействия и использует параллельное управление подготовкой данных для вычислений на графических процессорах. Это позволило равномерно распределять нагрузку по всем доступным ресурсам графического процессора и увеличить производительность по сравнению с существующими решениями.

3. Предложен алгоритм решения задачи молекулярного лиганд-белкового докинга на графических процессорах, использующих разработанный подход к выполнению метода дифференциальной эволюции. В отличие от существующих подходов алгоритм может выполнять молекулярный лиганд-белковый докинг для нескольких пар химических соединений одновременно на одном графическом процессоре, что позволяет повысить эффективность использования ресурсов графических процессоров и увеличить скорость обработки больших наборов лигандов.

Предложенные алгоритмы создают научную основу решения важных прикладных задач фармакологии и биоинформатики на базе гетерогенных вычислительных систем, использующих графические процессоры.

Практическая значимость полученных результатов. Разработанные в диссертации алгоритмы реализованы в виде комплекса программ. Применение разработанного комплекса программного обеспечения для решения задачи молекулярного лиганд-белкового докинга продемонстрировало существенное увеличение производительности по сравнению с аналогичными решениями. Алгоритм реализации метода дифференциальной эволюции на графических процессорах демонстрирует ускорение до 86 раз при полной загрузке графического процессора по сравнению с решениями, использующими только возможности центрального процессора. Предложенный алгоритм вычисления сеток межмолекулярного взаимодействия работает до 48 раз быстрее по сравнению с существующими методами решения. При выполнении

молекулярного докинга достигается ускорение до 17 раз при использовании одного графического процессора, и до 27 раз при использовании массива графических процессоров.

Апробация работы. Результаты исследования были представлены и обсуждены на следующих конференциях: XII Международная научная конференция «Интеллект и наука», Железногорск, 2012 год; X Международная научно-практическая конференция студентов, аспирантов и молодых учёных «Молодёжь и современные информационные технологии», Томск, 2012 год; Всероссийская научно-практическая конференция «Многоядерные процессоры, параллельное программирование, ПЛИС, системы обработки сигналов», Барнаул, 2013 год; XIII Международная научная конференция «Интеллект и наука», Железногорск, 2013 год.

Результаты диссертации опубликованы в 9 работах, из них в изданиях, рекомендованных Высшей Аттестационной Комиссией – 4.

Получено два свидетельства о регистрации программы для ЭВМ в Федеральном институте промышленной собственности (№2013660687; №2014619771).

Получен Акт о внедрении результатов диссертации в ООО «Мобилфон». Рекомендации, содержащиеся в диссертации, использовались для решения задач оптимизации планирования нагруженных вычислений в многосерверной среде. Получен Акт о внедрении результатов диссертации в ООО «Функциональные наносистемы». Рекомендации, содержащиеся в диссертации, использовались для решения задачи расчёта кинетики реакции и транспорта на границе раздела «металлический микросетчатый электрод – органическая электрохромная композиция».

Личный вклад автора. Результаты, представленные в диссертации, получены непосредственно автором: алгоритм реализации метода дифференциальной эволюции на графических процессорах; алгоритм вычисления сеток силовых полей межмолекулярного взаимодействия на графических процессорах; алгоритм выполнения молекулярного лиганд-белкового докинга на графических процессорах; комплекс программного обеспечения, реализующий указанные алгоритмы. Автором написаны и подготовлены к публикации все статьи из списка работ, представляющих результаты диссертации.

Структура и объём диссертации. Диссертация включает в себя введение, четыре главы, заключение, список использованных источников, список терминов и определений, приложение. Содержит 23 рисунка, 11 таблиц и 53 формулы. Список использованных источников содержит 135 наименований. Общий объём диссертации – 114 страниц.

СОДЕРЖАНИЕ РАБОТЫ

Во введении обоснована актуальность исследования, поставлена основная цель и определены задачи, которые необходимо решить в рамках проводимой работы.

В первой главе проблема молекулярного докинга сформулирована как многомерная оптимизационная задача, в которой требуется найти глобальный минимум следующей целевой функции:

$$E = E_{elec} + E_{vdw} = 18,22 \sum_{i < j} \frac{q_{target,i} q_j}{r_{ij}} + \sum_{i < j} \varepsilon_{ij} \left[\left(\frac{r_{0ij}}{r_{ij}} \right)^{12} - 2 \left(\frac{r_{0ij}}{r_{ij}} \right)^6 \right],$$

где E_{elec} – компонента, моделирующая электростатическое взаимодействие; E_{vdw} – компонента, моделирующая Ван-дер-Ваальсово взаимодействие; $q_{target,i}$ – заряд i -го атома биомишени; q_j – заряд атома лиганда. r_{ij} – расстояние между атомами; ε_{ij} – размерность энергии для пары атомов i и j ; r_{0ij} – эталонное Ван-Дер-Ваальсово расстояние между парами атомов i и j .

В главе исследованы существующие методы численной оптимизации. Подробно разобрано применение методов численной оптимизации для решения прикладных задач, а именно для выполнения молекулярного лиганд-белкового докинга.

В результате исследования особенностей информационных процессов, протекающих в гетерогенных вычислительных системах с графическими процессорами, сформулированы следующие специфичные требования, предъявляемые к методам многомерной численной оптимизации: высокая степень параллелизма по данным; максимальная линейность вычислительного процесса; отсутствие синхронизации или наличие только барьерной синхронизации; регулярная структура данных фиксированного объёма; минимальное количество входных параметров.

Метод дифференциальной эволюции в большей степени соответствует всем предъявленным выше требованиям, поэтому он выбран для исследования. Вместе с тем отдельно следует отметить высокую универсальность метода дифференциальной эволюции. Данный метод успешно применяется при решении оптимизационных задач, возникающих в различных прикладных областях: рентгенография и кристаллография, системы телекоммуникаций, анализ изображений, электромагнетизм, обработка сигналов, проектирование солнечных батарей.

Во второй главе проведён анализ существующих алгоритмов реализации метода дифференциальной эволюции с использованием графических процессоров. Выявлены недостатки существующих реализаций, которые ограничивают эффективность вычислений. Прежде всего, это наличие большого количества вызовов вычислительных ядер со стороны центрального процессора. Разделение алгоритма на несколько ядер является следствием другого недостатка – декомпозиции решаемой задачи с ориентацией на использование большого количества вычислительных блоков. CUDA (Compute Unified Device Architecture) не имеет явных средств выполнения синхронизации

вычислений между различными блоками. Такая синхронизация может быть выполнена либо при помощи атомарных операций над глобальной памятью, либо разбиением вычислений на несколько ядер и их синхронным запуском. Такой подход налагает ограничения на использования CUDA потоков и полноценную загрузку мультипроцессоров. Одним из ключевых недостатков существующих реализаций является неэффективное распределение вычислений для одной задачи оптимизации по всем доступным мультипроцессорам графического процессора.

Для преодоления существующих недостатков предложен алгоритм, который совершенствует существующие подходы и вносит следующие модификации:

1. Все этапы метода дифференциальной эволюции реализуются единым вызовом глобального вычислительного ядра, в пределах которого этапы метода дифференциальной эволюции максимально линеаризованы.

2. Все внутренние структуры спроектированы таким образом, чтобы обеспечивать линейный и последовательный доступ к памяти в пределах каждой группы нитей (варпа) с целью объединения запросов к глобальной памяти.

3. Вычислительный процесс организуется так, чтобы заменить мультиблочную вычислительную сетку и использовать возможности одного вычислительного блока для реализации одного запуска алгоритма так, чтобы удовлетворять требуемым размерам кластера векторов приближенных решений и спецификации CUDA. Это позволяет применять встроенные средства барьерной синхронизации и позволяет максимально эффективно использовать CUDA потоки для декомпозиции задачи. При этом декомпозиция задачи осуществляется таким образом, чтобы одновременно выполнялось несколько процедур оптимизации, число которых автоматически регулируется, базируясь на доступных ресурсах.

4. Исключение предварительной генерации случайных чисел. Предварительная генерация случайных чисел и вычисление их в отдельных CUDA потоках не оправданы, так как генерация случайных чисел для выполнения этапа мутации не является затратной процедурой в сравнении с вычислением целевой функции.

5. Выполнение первоначальной инициализации векторов на центральном процессоре. Начальная инициализация векторов кластера не является вычислительно затратной процедурой, поэтому следует оставить инициализацию на центральном процессоре. Это позволяет выполнять инициализацию более эффективно и, в случае необходимости, вводить дополнительные эмпирические соображения, обусловленные прикладной задачей.

Выполнения процедур оптимизации с использованием массива графических процессоров определяется следующей моделью:

$$M = (DE, GPU), DE = (de_1, de_2, \dots, de_n), GPU = (gpu_1, gpu_2, \dots, gpu_m),$$

где DE – множество всех процедур оптимизаций, которые необходимо выполнить; GPU – множество доступных графических процессоров. При этом:

$$\forall i \in [1, n], de_i = (D_i, F_i, CR_i),$$

где D_i – количество оптимизируемых параметров, F_i – константа мутации, CR_i – константа кроссинговера для i -ой процедуры оптимизации. Решение задачи оптимизации некоторой целевой функции определяется набором из NP_i векторов размерностью D_i . Каждый такой набор векторов называется кластером, или популяцией (V_i). Размерность популяции определяется согласно формуле:

$$|V_i| = NP_i \in [5D_i, 10D_i]$$

Графический процессор определяется как:

$$\forall j \in [1, m], gpu_j = (MX_j, STREAMS_j),$$

где MX_j – множество мультипроцессоров, $STREAMS_j$ – множество CUDA потоков, управляющих выполнением процедур оптимизации. Распределение процедур оптимизации для одного графического процессора, согласно предлагаемой модели, представлено как:

$$P_s \subset DE, |P_s| = |STREAMS_j|, P_1 \cup P_2 \cup \dots \cup P_r = DE, P_1 \cap P_2 \cap \dots \cap P_r = \emptyset, s = \overline{1, r}$$

$$f = P_s \rightarrow STREAMS_j$$

Предложенные изменения позволили сформировать новый алгоритм реализации метода дифференциальной эволюции на графических процессорах, который обеспечивает максимальную загрузку всех мультипроцессоров графического процессора, позволяет повысить производительность и сделать процесс вычислений максимально автономным для того, чтобы в значительной степени упростить диспетчеризацию выполнения нескольких процедур оптимизации на одном графическом процессоре.

В третьей главе предложен алгоритм выполнения молекулярного лиганд-белкового докинга, ускоряющий вычисления за счёт использования графических процессоров. В основу разработанного программного обеспечения был положен алгоритм реализации метода дифференциальной эволюции, предложенный во второй главе. Кроме того, для ускорения молекулярного лиганд-белкового докинга предложен и реализован алгоритм вычислений сеток силовых полей с использованием графических процессоров, который может использоваться как отдельный самостоятельный программный продукт.

Вычисление сеток силовых полей декомпозировано на два уровня: на уровне сеток и на уровне ячеек сеток. Первый уровень декомпозиции позволяет эффективно балансировать нагрузку при использовании нескольких графических процессоров. Второй уровень декомпозиции позволяет эффективно использовать ресурсы отдельного графического процессора и полностью задействовать доступные мультипроцессоры.

Для полной загрузки каждого мультипроцессора графического процессора необходимо обеспечивать для него достаточный объём заданий, подготовленный к вычислению, т.е. размещенный надлежащим образом в

памяти графического процессора. В данном случае ограничением является получение информации о биомолекулах и вывод вычисленных сеток силовых полей на диск. Для преодоления данной проблемы и сглаживания задержек, обусловленных вводом/выводом, предлагается специальный менеджер времени выполнения. Менеджер представляет собой разделяемый ресурс, который используется тремя потоками центрального процессора: ввод информации с диска, управление вычислениями на графическом процессоре, вывод результатов на диск. Менеджер в свою очередь поддерживает две очереди, а также методы для работы с ними: очередь для задач, готовых выполняться на графическом процессоре; очередь для задач, которые были посчитаны на графическом процессоре.

Разработанное решение, в первую очередь, предназначено для быстрой обработки большого набора лигандов при изучении некоторой определённой биомолекулы, так как это наиболее востребованная задача при поиске кандидатов в лекарства. Модель позволяет учесть использование всех доступных ресурсов графического процессора, равномерное распределение вычислительной нагрузки и использование нескольких графических процессоров системы. Программное обеспечение ориентировано на выполнение гибкого лиганд-белкового докинга с учётом подвижности лиганда по торсионным углам.

При выполнении лиганд-белкового докинга в качестве оптимизируемых параметров выступают 6 стандартных степеней свободы (смещение всего лиганда по координатным осям и его вращение вокруг координатных осей), а также, в случае необходимости, по одной переменной на каждый торсионный угол. Предложенный алгоритм параллельной реализации метода дифференциальной эволюции на графических процессорах позволяет обрабатывать несколько лигандов одновременно, причем каждый отдельный лиганд одним вычислительным CUDA блоком.

Специфичным для лиганд-белкового докинга является этап вычисления целевой функции, который разделён на две стадии. Прежде всего каждой нитью графического процессора выполняется трансформация лиганда согласно вектору приближенного решения, после чего выполняется вычисление целевой функции. Так как при выполнении итераций метода дифференциальной эволюции значения переменных приближенного решения меняются произвольным образом, трансформация лиганда может привести к формированию нефизической трёхмерной структуры при изменении некоторого торсионного угла таким образом, что часть лиганда будет повернута «в себя», либо может привести к выходу лиганда за пределы заданной сетки, ограничивающей сайт связывания биомолекулы. Поэтому предварительно до вычисления целевой функции выполняется проверка на расположение лиганда в пределах трёхмерной сетки, а также на столкновение с биомолекулой. Если в процессе проверки структура признана подходящей, производится вычисление целевой функции.

Для ускорения выполнения докинга используется сеточный подход. Так как сетки силового поля вычисляются с некоторым детерминированным шагом,

положения атомов лиганда после трансформации не может полностью соответствовать ячейкам сетки. Таким образом, для вычисления значения энергии атомов необходимо применять интерполяцию, в данном случае трилинейную.

Вычисление значения целевой функции производится на каждой итерации метода дифференциальной эволюции, а также при запуске метода для вычисления целевой функции для инициализированных значений векторов.

Процесс выполнения молекулярного лиганд-белкового докинга с учётом предложенного алгоритма реализации метода дифференциальной эволюции с использованием графических процессоров имеет вид:

$$D = (LIGANDS, TARGET, GPU),$$

$$LIGANDS = (ligand_1, ligand_2, \dots, ligand_n),$$

$$GPU = (gpu_1, gpu_2, \dots, gpu_m),$$

где $LIGANDS$ – множество всех лигандов, которые необходимо обработать, т.е. провести процедуру оптимизации; GPU – множество доступных графических процессоров. При этом:

$$\forall i \in [1, n], ligand_i = (F_i, CR_i, ATOMS_i, TORSIONS_i, clash_i),$$

где F_i и CR_i – параметры i -ой процедуры оптимизации, $ATOMS_i$ – множество всех атомов лиганда, $TORSIONS_i$ – множество всех торсионных углов лиганда, $clash_i$ – карта складывания лиганда.

$$|V_i| = [5(6 + |TORSIONS_i|), 10(6 + |TORSIONS_i|)]$$

Графический процессор представлен следующим образом:

$$\forall j \in [1, m], gpu_j = (MX_j, STREAMS_j, GRIDS_j),$$

где MX_j – множество мультипроцессоров графического процессора, $STREAMS_j$ – множество CUDA потоков, управляющих выполнением процедур оптимизации, $GRIDS_j$ – множество сеток силовых полей для биомишени, представленных на графическом процессоре.

Распределение вычислений для одного графического процессора представлено как:

$$R_s \subset LIGANDS, |R_s| = |STREAMS_j|, R_1 \cup R_2 \cup \dots \cup R_r = DE,$$

$$R_1 \cap R_2 \cap \dots \cap R_r = \emptyset, s = \overline{1, r}$$

$$f = R_s \rightarrow STREAMS_j$$

В четвёртой главе представлены результаты тестирования предложенных решений на различных графических процессорах в сравнении с альтернативным программным обеспечением.

Предложенный алгоритм реализации метода дифференциальной эволюции на графических процессорах был протестирован на сходимость на эталонных тестовых наборах. Результаты этого тестирования приведены в таблице 1.

Таблица 1 – Результаты тестирования сходимости алгоритма реализации метода дифференциальной эволюции на графических процессорах

Функция	D	NP	F	CR	VTR	IPR	NEF
Goldstein's function	1	10	0,5	0,1	7,1	[-10;10]	18
De Jong function 1	3	30	0,9	0,1	e-6	[-5,12;5,12]	67
Rosenbrock's saddle	2	20	0,9	0,9	e-6	[-2,048;2,048]	60
Penalized Shubert Function	1	10	0,5	0	-12,8	[-10;10]	26
2d Penalized Shubert Function	2	20	0,5	0	-186,73	[-10;10]	148
Hyper-Ellipsoid	30	30	0,5	0,1	e-10	[-1;1]	481
Hyper-Ellipsoid	100	1000	0,5	0,1	e-10	[-1;1]	3204
Griewangk's function	20	200	0,5	0,2	e-3	[-600;600]	705
Griewangk's function	100	1000	0,5	0,2	e-3	[-600;600]	2216
Rastrigin's Function	20	200	0,5	0	0,9	[-600;600]	329
Rastrigin's Function	100	1000	0,5	0	0,9	[-600;600]	3849
Ackley's function	30	300	0,5	0,1	e-3	[-30;30]	1580
Ackley's function	100	1000	0,5	0,1	e-3	[-30;30]	81154

Принятые обозначения:

D — количество переменных;

NP — количество векторов приближенных решений;

F,CR — параметры метода дифференциальной эволюции;

VTR — контрольное значение, по достижению которого считается, что найден глобальный экстремум;

IPR — область инициации параметров;

NEF — количество вычислений целевой функции, которое потребовалось для достижения глобального экстремума.

Из результатов тестирования следует: для всех функций, приведенных в первом столбце таблицы 1, предложенный алгоритм привел к глобальному экстремуму и для большинства функций показал требуемую скорость сходимости. Исключение составила лишь функция Экли для 100 оптимизируемых переменных, где скорость сходимости оказалась ниже (81154 NEF против заявленных 36801).

Для тестирования производительности предложенного решения была выбрана функция Экли как наиболее вычислительно затратная процедура из тестового набора. В качестве параметров для тестов были установлены: размер вектора параметров 100, размер кластера векторов 100. Параметры для метода дифференциальной эволюции были оставлены без изменения. Все тесты выполнялись для 10000 NEF с целью исключить искажения в оценке производительности. Результаты тестов представлены в таблице 2.

Таблица 2 – Результаты тестирования производительности алгоритма реализации метода дифференциальной эволюции на графических процессорах

Конфигурация	Время вычислений, сек	Ускорение	Ускорение в расчёте на общее количество MX
CPU 1 thread	951	6,75	86,46
CPU 12 threads	287	2,04	26,09
GPU 1 MX*	141	-	-

* предлагаемое решение

Тестирование показало, что предложенный алгоритм реализации метода дифференциальной эволюции работает в 2 раза быстрее по сравнению с 12-ти потоковой реализацией этого метода на центральном процессоре, кроме того

при полной загрузке графического процессора обеспечивается ускорение в 26 раз.

Предложенная реализация сравнивалась с возможностями программы AutoGrid 4. В качестве теста производилось вычисление 9 сеток силовых полей для одной биомешины (Таблица 3). Тестирование показало, что предложенный алгоритм вычисления сеток силовых полей демонстрирует значительное ускорение по сравнению с AutoGrid 4.

Таблица 3 – Среднее время вычисления 9 сеток силовых полей

Конфигурация	Затраченное время, мс	Ускорение
AutoGrid 4	74010	-
Tesla M2090*	1824,04	40,58
Tesla K20m*	1523,21	48,59
GeForce GT 440*	11370,01	6,51

* предлагаемое решение

Для тестирования предложенной программной реализации алгоритма молекулярного лиганд-белкового докинга использовались графические процессоры NVIDIA Tesla K20m и Tesla K40. Тесты проводились как для одиночных графических процессоров, так и для массива графических процессоров.

В качестве альтернативных решений, использующих возможности только центрального процессора, были выбраны наиболее популярные пакеты для выполнения молекулярного докинга: AutoDock4 (как наиболее точное существующее решение) и AutoDock Vina (как наиболее быстрое существующее решение).

Были оценены временные затраты работы программ в пересчёте на один лиганд, что необходимо для последующих тестов скрининга большой базы данных. AutoDock Vina тестировалась в трёх конфигурациях, так как использует возможности многоядерных систем. Для программы AutoDock4 время, затрачиваемое на подготовку и предвычисление сеток силовых полей, в результаты тестов не включалось. Для предлагаемой реализации производился запуск с копиями тестового лиганда в количестве, эквивалентном количеству мультипроцессоров графического процессора, после чего время делилось на это количество. Результаты тестов представлены в таблице 4.

Таблица 4 – Время выполнения молекулярного докинга в расчете на один лиганд

Программа	Время выполнения, с
AutoDock Vina (cpu=24)	16
AutoDock Vina (cpu=12)	15,9
AutoDock Vina (cpu=6)	19,5
AutoDock4	153,2
GDock NVIDIA Tesla K20m*	3,12
GDock NVIDIA Tesla K40m*	1,27

* предлагаемое решение

По результатам проведённого теста видно, что предлагаемое решение работает быстрее в пересчёте на один лиганд.

Несмотря на высокую точность работы, программа AutoDock4 была исключена из теста скрининга базы данных лигандов из-за чрезвычайно низкой скорости работы. Для выполнения скрининга была выбрана биомишень с PDB идентификатором 3PTB и лиганд с ZINC идентификатором 36634. К тестовому набору случайным образом было добавлено 99 лигандов, содержащихся в открытой базе данных химических соединений (лекарств, разрешённых к использованию). Предлагаемое решение дополнительно тестировалось на массиве из трёх графических процессоров. Результаты тестов представлены в таблице 5.

Таблица 5 – Время выполнения молекулярного докинга для набора из 100 лигандов

Программа	Время выполнения, с
AutoDock Vina (cpu=24)	2253
AutoDock Vina (cpu=12)	2239
AutoDock Vina (cpu=6)	2302
GDock NVIDIA Tesla K20m*	322
GDock NVIDIA Tesla K20m X3*	85
GDock NVIDIA Tesla K40*	131

* предлагаемое решение

Предлагаемое решение демонстрирует существенный прирост производительности до 17 раз при использовании одного графического процессора и до 28 раз при использовании массива графических процессоров даже по сравнению с многопоточными решениями.

Для тестирования точности вычисления был выполнен молекулярный докинг для ряда биомишеней с их лигандами, позиционирование которых в активном сайте было определено при помощи кристаллографии. По результатам проведения процедуры для лучшей структуры было посчитано среднеквадратичное отклонение (RMSD) предсказанного положения от истинного. Результаты тестов представлены в таблице 6.

Таблица 6 – Среднеквадратичное отклонение предсказанных конформаций молекулы биомишени и лиганда от истинного значения

Комплекс	RMSD, Ангстрем
PDB:1ULB	2,15
PDB:2MCP	1,79
PDB:3PTB	0,44

Таким образом, по результатам серии тестов можно сделать выводы, что разработанные алгоритмы и программы значительно быстрее существующих аналогов, использующих только возможности центрального процессора. Они также гарантируют приемлемую точность, несмотря на то, что в процессе выполнения докинга используется только глобальный поиск для поиска конформации.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ

1. Предложен алгоритм выполнения дифференциальной эволюции с использованием графических процессоров. Предлагаемая последовательность реализации этапов дифференциальной эволюции обеспечивает распределение отдельных процедур оптимизации по мультипроцессорам графического процессора, позволяя тем самым увеличить нагрузку на графический процессор и использовать встроенные средства синхронизации. Разработанный алгоритм хорошо подходит для решения задач, связанных с выполнением большого количества процедур оптимизации.

2. Разработан алгоритм расчёта силовых полей межмолекулярного взаимодействия с помощью сеток на графических процессорах. Предлагаемый алгоритм позволяет эффективно автоматически распределять вычисления различных типов сеток силовых полей для больших баз химических соединений между доступными ресурсами графических процессоров. Управление вычислениями организовано таким образом, чтобы обеспечить максимальную загрузку графических процессоров и скрыть задержки работы с дисковой памятью. Алгоритм демонстрирует значительное ускорение вычислений по сравнению с аналогами.

3. Предложен алгоритм выполнения молекулярного лиганд-белкового докинга с использованием графических процессоров. Алгоритм может выполнять молекулярный лиганд-белковый докинг для нескольких пар химических соединений одновременно на одном графическом процессоре. Алгоритм ориентирован на автоматическую обработку больших баз лигандов и позволяет прозрачно для пользователя распределять нагрузку по всем доступным графическим процессорам. При выполнении вычислительных экспериментов алгоритм показал существенное ускорение процесса молекулярного моделирования и требуемую точность вычислений при обработке реальных химических соединений.

4. Создан комплекс программ, реализующий разработанные алгоритмы и позволяющий провести вычислительные эксперименты для решения задач вычисления сеток силовых полей и молекулярного лиганд-белкового докинга с использованием графических процессоров, что повысило эффективность проводимых вычислений.

СПИСОК РАБОТ ОПУБЛИКОВАННЫХ ПО ТЕМЕ ДИССЕРТАЦИИ

В изданиях, рекомендованных ВАК:

1. Farkov, M. Calculation of force field grids for molecular docking using GPU / M. Farkov // Journal of Siberian Federal University. Biology. – 2014. – Т.7, № 1. – С. 4-13.

2. Фарков, М.А. Улучшение реализации метода дифференциальной эволюции на графических процессорах / М.А. Фарков, А.И. Легалов // Вестник

Сибирского государственного аэрокосмического университета имени академика М.Ф. Решетнева. – 2014. – Т. 3, № 55. – С. 157-161.

3. Фарков, М.А. Применение методов оптимизации для выполнения молекулярного докинга на графических процессорах / М.А. Фарков, А.И. Легалов // Моделирование и анализ информационных систем. – 2014. – Т. 21, № 5. – С. 93-101.

4. Farkov, M.A. Application of Numerical Optimization Methods to Molecular Docking on Graphics Processing Units / M.A. Farkov, A.I. Legalov // Automatic Control and Computer Sciences. – 2016. – Т.50, №7. – С. 471-476.

В трудах конференций:

5. Фарков, М.А. Выполнение молекулярного докинга с использованием графических ускорителей / М.А. Фарков // Сборник трудов XII Международной молодёжной научной конференции «Интеллект и наука труды». – Красноярск. – 2012. – С. 181-182.

6. Фарков, М.А. Специфика предобработки биоинформационных данных для графических процессоров / М.А. Фарков // Сборник трудов всероссийской научно-практической конференции «Многоядерные процессоры, параллельное программирование, ПЛИС, системы обработки сигналов». – Барнаул. – 2013. – С. 76-79.

7. Фарков, М.А. Сеточный подход к выполнению молекулярного докинга. Его достоинства и недостатки. / М.А. Фарков // Сборник трудов X Международной научно-практической конференции студентов, аспирантов и молодых ученых «Молодежь и современные информационные технологии». – Томск. – 2012. – С. 186-188.

8. Фарков, М.А. Ускорение вычисления сеток силовых полей при помощи графических процессоров / М.А. Фарков // Сборник трудов XIII Международной молодёжной научной конференции «Интеллект и наука труды». – Железногорск. – 2013. – С. 129-131.

9. Фарков, М.А. Вычисление сеток взаимодействия молекул с использованием графических процессоров / М.А. Фарков // Исследования наукограда. – 2014. – Т. 21, № 3. – С. 49-52.

Авторские свидетельства на программы:

1. Фарков М.А. Программа для ЭВМ GPU-CGFF // Свидетельство о государственной регистрации программы для ЭВМ №2013660687 от 14го ноября 2013.

2. Фарков М.А. Программа для ЭВМ GDock // Свидетельство о государственной регистрации программы для ЭВМ № 2014619771 от 22го сентября 2014.