На правах рукописи

Селиверстов Евгений Юрьевич

Структурно-параметрическое согласование метаэвристических алгоритмов глобальной оптимизации с архитектурой графических процессорных устройств

Специальность 2.3.5. Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей (технические науки)

АВТОРЕФЕРАТ
диссертации на соискание ученой степени
кандидата технических наук

Москва – 2023

Работа выполнена на кафедре систем автоматизированного проектирования в Федеральном государственном бюджетном образовательном учреждении высшего образования «Московский государственный технический университет им. Н.Э. Баумана (национальный исследовательский университет)» (МГТУ им. Н.Э. Баумана).

Научный руководитель	**Карпенко Анатолий Павлович** доктор физико-математических наук, профессор
Официальные оппоненты	**Посыпкин Михаил Анатольевич** доктор физико-математических наук, доцент, член-корреспондент РАН, Федеральное государственное учреждение «Федеральный исследовательский центр „Информатика и управление“ Российской Академии Наук», заместитель директора по научной работе
	Мунерман Виктор Иосифович кандидат технических наук, доцент, Федеральное государственное бюджетное образовательное учреждение высшего образования «Смоленский государственный университет», доцент
Ведущая организация	Федеральное государственное бюджетное образовательное учреждение науки «Институт проблем управления имени В.А. Трапезникова» Российской Академии Наук

Защита состоится 15 февраля 2024 года в 15 час. 00 мин. на заседании диссертационного совета *24.2.331.19* на базе МГТУ им. Н.Э. Баумана по адресу: 105005, Москва, 2-я Бауманская ул., д.5, стр.1, зал Ученого совета ГУК.

С диссертацией можно ознакомиться в библиотеке *МГТУ им. Н.Э. Баумана* и на сайте https://bmstu.ru.

Автореферат разослан «____» _____ 2023 г.

Отзывы и замечания по автореферату в двух экземплярах, заверенные печатью, просьба высылать по адресу: 105005, Москва, 2-я Бауманская ул., д.5, стр.1, кафедра ИУ 3, на имя ученого секретаря диссертационного совета.

Ученый секретарь
кандидат технических наук,
доцент *С.А. Сакулин*

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы исследования. Задачи глобальной оптимизации возникают во многих областях науки и техники. Важным классом алгоритмов оптимизации для решения этих задач являются *метаэвристические алгоритмы* глобальной оптимизации (МЭГО-алгоритмы), например, алгоритм роя частиц. В силу высокой вычислительной сложности современных задач глобальной оптимизации актуальным является применение параллельных вычислительных систем для решения этих задач. Перспективным классом таких систем выступают графические процессорные устройства (ГПУ), отличительной особенностью которых является иерархическая структура вычислительных устройств и памяти. Следствием этого является актуальность разработки математического и программного обеспечения для эффективного распараллеливания МЭГО-алгоритмов на ГПУ.

Отличительной особенностью МЭГО-алгоритмов является наличие в них свободных параметров, оказывающих существенное влияние на их эффективность, например, на скорость сходимости и вероятность локализации глобального экстремума. Задача отыскания оптимальных значений структурных и свободных параметров параллельного МЭГО-алгоритма называется задачей структурно-параметрического согласования этого алгоритма с архитектурой параллельной вычислительной системы (ПВС). Для решения данной задачи необходимо решать задачу «оптимального» отображения параллельного МЭГО-алгоритма на архитектуру ПВС для каждой из рассматриваемых структур и значений его свободных параметров.

Исследования разных аспектов структурно-параметрического согласования МЭГО-алгоритмов с архитектурой ПВС проводят специалисты K. De Jong, D. Fogel, P. Angeline, R. Smith, T. Back, J. Dongarra, F. Song, S. Chen, В. Воеводин, А. Карпенко, М. Посыпкин. Существующие методы решения задачи согласования ориентированы в основном на классические архитектуры ПВС. Для ГПУ слабо развиты методы автоматического синтеза программы оптимизации, учитывающие особенности архитектуры ГПУ и обеспечивающие высокую параллельную эффективность оптимизации.

Объектом исследования является системное программное обеспечение ГПУ.

Предметом исследования является задача структурно-параметрического согласования параллельных МЭГО-алгоритмов с архитектурой ГПУ.

Целью диссертационной работы является разработка и исследование эффективности математических моделей, методов и алгоритмов структурно-параметрического согласования параллельных МЭГО-алгоритмов с архитектурой ГПУ.

Задачи исследования.

1) Обзор методов согласования МЭГО-алгоритмов с архитектурой ГПУ.

2) Разработка математических моделей ГПУ.

3) Разработка методов и алгоритмов решения задачи структурно-параметрического согласования МЭГО-алгоритмов с архитектурой ГПУ.

4) Разработка экспериментальной программной системы, реализующей предложенные математические модели, методы и алгоритмы.

5) Исследование эффективности разработанного математического и программного обеспечения при решении тестовых и практически значимых задач.

Научная новизна.

1) Предложен параллельный метаэвристический алгоритм глобальной оптимизации (МЭГО-алгоритм), отличающийся использованием графового представления, позволяющего формализовать операции алгоритма, коммуникации между ними и выявить информационную структуру алгоритма.

2) Предложена математическая модель графического процессорного устройства (ГПУ), состоящая из графовой структурной модели, графовой коммуникационной модели и модели памяти ГПУ. Отличительной особенностью модели является детализация особенностей структуры ГПУ, коммуникаций и ограничений доступа к памяти, что позволяет синтезировать программы оптимизации с высокой параллельной эффективностью.

3) Формализована задача структурно-параметрического согласования МЭГО-алгоритма с архитектурой ГПУ, отличающаяся использованием графового отображения этого алгоритма на модель ГПУ и ограничивающих функций отображения, что позволяет сформировать множество допустимых отображений МЭГО-алгоритма на архитектуру ГПУ и повысить эффективность решения задачи согласования.

4) Предложены иерархический метод и алгоритм согласования МЭГО-алгоритма с архитектурой ГПУ для решения задачи структурно-параметрической оптимизации по векторному критерию оптимальности. Отличительной особенностью метода и алгоритма является совместное выполнение программы базового МЭГО-алгоритма и программы, реализующей предложенные алгоритмы структурного и параметрического согласования алгоритма с архитектурой ГПУ, что позволяет осуществлять динамическую адаптацию структурных и свободных параметров для МЭГО-алгоритма и поиск оптимального отображения этого алгоритма на архитектуру ГПУ.

5) Предложена структура программной системы глобальной оптимизации, реализующей предложенное математическое обеспечение и осуществляющей решение базовой задачи оптимизации при помощи расширения языка программирования C++. Предложен алгоритм автоматического синтеза программы оптимизации, отличительной особенностью которого является построение программы оптимизации на основе этого языка и графов МЭГО-алгоритма.

Теоретическая и практическая значимость. Разработаны математические модели, методы и алгоритмы согласования МЭГО-алгоритма с архитектурой ГПУ, позволяющие повысить эффективность решения задачи глобальной оптимизации.

Разработана программная система, реализующая предложенные математические модели, методы и алгоритмы. Основой системы является подсистема синтеза и исполнения параллельной программы, позволяющая синтезировать параллельную программу, оптимальную для данного ГПУ, а также решить с помощью этой программы задачу глобальной оптимизации. Использование программной системы позволило повысить точность и скорость решения тестовых задач глобальной оптимизации.

С использованием разработанного математического и программного обеспечения решена практическая обратная параметрическая задача химической кинетики, основанная на результатах экспериментальных исследований реакции каталитического гидроалюминирования олефинов Институтом нефтехимии и катализа (ИНК) РАН.

Методы исследования. При разработке математических моделей, методов, алгоритмов и программного обеспечения использовались методы теории графов, теории вероятностей, теории параллельных вычислений и языков программирования, методы нелинейного программирования, многокритериальной оптимизации, дискретной оптимизации. Использованы численные методы решения систем обыкновенных дифференциальных уравнений и систем линейных уравнений.

Основные положения, выносимые на защиту.

1) Графовое представление параллельного МЭГО-алгоритма и математическая модель ГПУ, учитывающие особенности рассматриваемого класса параллельных алгоритмов и архитектуры ГПУ.

2) Иерархический метод и алгоритм согласования МЭГО-алгоритма с архитектурой ГПУ для решения задачи структурно-параметрической оптимизации по векторному критерию оптимальности, отличающиеся совместным выполнением программы базового МЭГО-алгоритма и программы, реализующей предложенные алгоритмы, что позволяет осуществлять динамическую адаптацию структурных параметров, свободных параметров и отображения для МЭГО-алгоритма.

3) Структура программной системы, реализующей предложенные математические модели, методы и алгоритмы, и осуществляющей автоматический синтез параллельной программы оптимизации на основании графовых моделей МЭГО-алгоритма и ГПУ.

Соответствие паспорту научной специальности. Содержание работы соответствует паспорту научной специальности 2.3.5 Математическое и программное обеспечение вычислительных машин, комплексов и компьютер-

ных сетей (технические науки): п.2 «Языки программирования и системы программирования, семантика программ», п.3 «Модели, методы, архитектуры, алгоритмы, языки и программные инструменты организации взаимодействия программ и программных систем», п.8 «Модели и методы создания программ и программных систем для параллельной и распределенной обработки данных, языки и инструментальные средства параллельного программирования».

Достоверность полученных в диссертации научных результатов обеспечивается корректностью используемого математического аппарата и подтверждается вычислительными экспериментами на широком классе тестовых задач.

Апробация работы. Основные результаты и положения диссертационной работы представлены на международном симпозиуме «Интеллектуальные системы» (Санкт-Петербург, 2018); международной научной конференции «Параллельные Вычислительные Технологии» (Челябинск, 2013); всероссийских суперкомпьютерных конференциях «Научный сервис в сети Интернет» (Абрау-Дюрсо, 2009, 2011—2013); международной молодежной конференции «Наукоемкие технологии и интеллектуальные системы» (Москва, 2011, 2012); международной научной конференции «Системы компьютерной математики и их приложения» (Смоленск, 2017).

Внедрение результатов работы. Результаты диссертационной работы использованы при решении задач химической кинетики, исследуемых Лабораторией математической химии Института нефтехимии и катализа РАН, а также в учебном процессе кафедры САПР МГТУ им. Н.Э. Баумана.

Публикации. Основные результаты диссертации опубликованы в 16 научных работах в журналах и сборниках трудов конференций, из них 3 работы — в рецензируемых журналах, рекомендованных ВАК РФ, 2 работы — в сборниках, индексируемых в Scopus и Web of Science.

Личный вклад автора. Лично автором разработаны предложенные математические модели, методы, алгоритмы и программное обеспечение. Содержание диссертации и основные положения, выносимые на защиту, отражают персональный вклад автора в опубликованные работы.

Объем и структура диссертации. Диссертация состоит из введения, четырех глав, основных выводов по работе, одного приложения, заключения, библиографии и списка литературы из 250 наименований. Объем работы составляет 220 страниц, включая 38 рисунков и 6 таблиц.

СОДЕРЖАНИЕ РАБОТЫ

Во введении обоснована актуальность диссертационной работы, сформулированы цель и научная новизна результатов исследований, показана практическая значимость полученных результатов, представлены выносимые на защиту научные положения.

В первой главе рассмотрена *базовая задача* глобальной оптимизации

$$\min_{X \in D_X} \Phi(X) = \Phi(X^*)\,, \tag{1}$$

где X — вектор *варьируемых параметров*; X^* — искомый оптимальный вектор; $D_X \subset \mathbb{R}^d$ — область допустимых значений X; $\Phi(X)$ — целевая функция. МЭГО-алгоритм, используемый для решения базовой задачи, называется *базовым алгоритмом*, а задача отыскания оптимальных в некотором смысле значений свободных параметров этого алгоритма — *задачей установки*, решаемой методами статической настройки (до выполнения алгоритма) или методами динамической адаптации (во время выполнения алгоритма). Программа, реализующая решение базовой задачи, называется *базовой программой*.

В главе приведен обзор основных классов последовательных и параллельных МЭГО-алгоритмов, моделей параллелизма, методов установки свободных параметров этих алгоритмов. Выполнен обзор методов согласования параллельных МЭГО-алгоритмов с архитектурой ГПУ. Рассмотрены особенности ГПУ, важные с точки зрения решения задачи согласования. Сформулированы основные требования к методам и алгоритмам решения этой задачи.

Во второй главе предложены математические модели ГПУ и параллельного метаэвристического алгоритма.

Математическая модель ГПУ предложена в виде набора графовых моделей: структурная модель; коммуникационная модель; модель памяти. *Структурная модель* $GS = (VS, ES)$ отражает иерархическую структуру ГПУ. Вершины VS представляют вычислительные устройства уровней хост-системы (h), ГПУ (g), мультипроцессора (mp), скалярного процессора (sp), а ребра ES — отношение включения этих устройств. На Рис. 2а приведен пример графа GS, где этим уровням соответствуют множества вершин (h), (g_1), (mp_{11}, mp_{12}) и $(sp_{111}, sp_{112}, sp_{121}, sp_{122})$ соответственно. В графовой *коммуникационной модели* $GC = (VS, EC)$ ребра EC представляют допустимые коммуникационные связи между указанными вычислительными устройствами. *Модель памяти* ГПУ $GM = (VM, EM)$, где $VM = VS \cup VMM$, определяет ограничения доступа, заданные ребрами EM, к различным видам памяти ГПУ (множество вершин VMM) с его вычислительных устройств, представленных вершинами VS. Совокупность указанных моделей формализует структурные и коммуникационные особенности ГПУ, необходимые для согласования МЭГО-алгоритмов с архитектурой ГПУ. Преимущество предложенных моделей ГПУ перед другими известными моделями состоит в явном представлении структуры ГПУ, коммуникаций и ограничений доступа к памяти ГПУ.

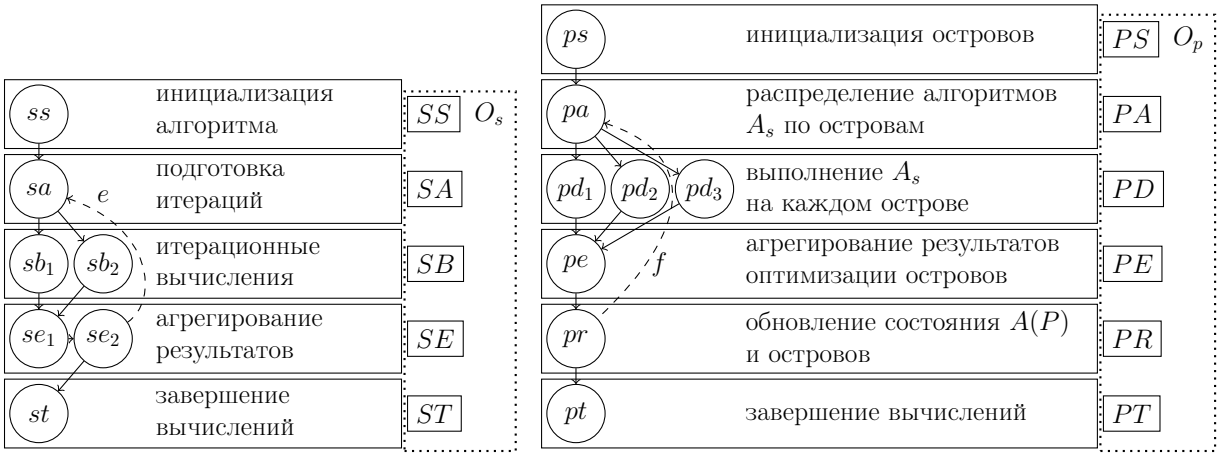
Математическая модель базового алгоритма. Рассматривается класс $\mathbb{MA}$ последовательных базовых МЭГО-алгоритмов A_s, которые имеют популяционную структуру, где каждый агент популяции представляет значение вектора варьируемых параметров X (например, алгоритмы эволюционной стратегии, дифференциальной эволюции). В популяционном алгоритме роя частиц (PSO) каждый агент (частица) из популяции (роя) вычисляет значение вектора X на основании своего состояния, а на каждой итерации происходит выбор лучшей частицы из роя и расчет новых состояний частиц.

Для синтеза параллельного МЭГО-алгоритма (*синтезируемого алгоритма*) используется островная модель параллелизма. Популяция I разделена на непересекающиеся «острова» (субпопуляции) I_i. Каждому из островов I_i поставлен в соответствие один и тот же последовательный базовый алгоритм $A_s(P_s)$, который выполняет оптимизацию векторов X этого острова в *фазе вычислений*. В *фазе синхронизации* и *фазе коммуникации* параллельный алгоритм осуществляет обмен агентами популяции между островами, что повышает разнообразие общей популяции и позволяет предотвратить стагнацию процесса оптимизации. Здесь $P_s \in D_{P_s}$ — вектор свободных параметров алгоритма $A_s(P_s)$; D_{P_s} — множество допустимых значений этого вектора. В алгоритме PSO, например, свободными параметрами являются число частиц в рое, коэффициент инерции и топология соседства частиц в рое. Синтезируемый алгоритм обозначен $A(P)$, где вектор $P = (P_s, P_p)$ свободных параметров включает вектор P_s и вектор $P_p \in D_{P_p}$ структурных параметров параллельного алгоритма; D_{P_p} — область допустимых значений компонентов этого вектора.

В качестве математической модели алгоритма $A(P)$ предложена иерархическая графовая модель $GA(P)$, состоящая на *нижнем уровне* из модели $GA_s(P_s)$ алгоритма $A_s(P_s)$, а на *верхнем уровне* из модели $GA_p(P)$ собственно алгоритма $A(P)$. Графовые модели $GA_s(P_s)$, $GA_p(P)$ представлены в виде невзвешенных ориентированных графов вида $G = (V, E)$, где множество V вершин графа обозначает *операции алгоритма*, а множество E ребер — зависимости по данным между этими операциями.

На основании классификации операций последовательных МЭГО-алгоритмов (De Jong, Bäck) в диссертации предложено разбиение O_s множества операций алгоритма A_s на *классы операций* SS, SA, SB, SE, ST. Аналогично для синтезируемого алгоритма $A(P)$ предложено разбиение O_p множества операций на классы операций PS, PA, PD, PE, PR, PT.

На Рис. 1,а приведен пример графа $GA_s(P_s)$ и разбиения O_s. Для последовательного алгоритма PSO классу SB соответствуют операции вычисления координат и скоростей частиц, классу SE — отыскание лучшей частицы в рое, классу ST — определение искомого оптимального вектора X^*. На Рис. 1,б приведен пример графа $GA_p(P)$, в котором вершинам pd_1, pd_2, pd_3 соответствует класс операций PD выполнения алгоритма A_s, а вершине $pr \in PR$ — операции



а) Граф GA_s б) Граф GA_p

Рис. 1.

Примеры графов GA_s и GA_p алгоритмов $A_s(P_s)$, $A(P)$ и их классы операций синхронизации алгоритма $A(P)$.

Третья глава посвящена разработке методов и алгоритмов согласования базового МЭГО-алгоритма с архитектурой ГПУ. В главе представлена формализация задачи структурно-параметрического согласования, изложены предложенные в диссертации методы и алгоритмы решения этой задачи.

Формализация задачи структурно-параметрического согласования. Предлагается *T-отображение* $T(GA(P), \mathcal{M})$ графа $GA(P)$ на графы $\mathcal{M} = (GC, GS, GM)$ ГПУ как кортеж, состоящий из вершинных и реберных отображений графа $GA(P)$ на эти графы. *Вершинное отображение* $\lambda(G, H)$ графа G на граф H представляется в виде вектора $(\lambda_1, \ldots, \lambda_k)$ *компонент отображения* $\lambda_k = \{g_k, h_k\}$, где g_k, h_k — подмножества вершин графов G, H соответственно, и задает соответствие подмножеств вершин $\{g_k\}$ и $\{h_k\}$. *Реберное отображение* $\eta(G, H) = \{(e_g, e_h)\}$ задает соответствие ребер e_g,e_h графов G, H. Множество D_T допустимых T-отображений сформировано с помощью *ограничивающей функции*, задающей ограничения на вершинные и реберные отображения исходя из особенностей архитектуры ГПУ и МЭГО-алгоритма.

На Рис. 2,б приведен пример вершинного отображения $\lambda(GA_p, GS) = (\lambda_1 \ldots \lambda_4)$ графа $GA_p(P)$ на структурный граф ГПУ GS (Рис. 2,а).

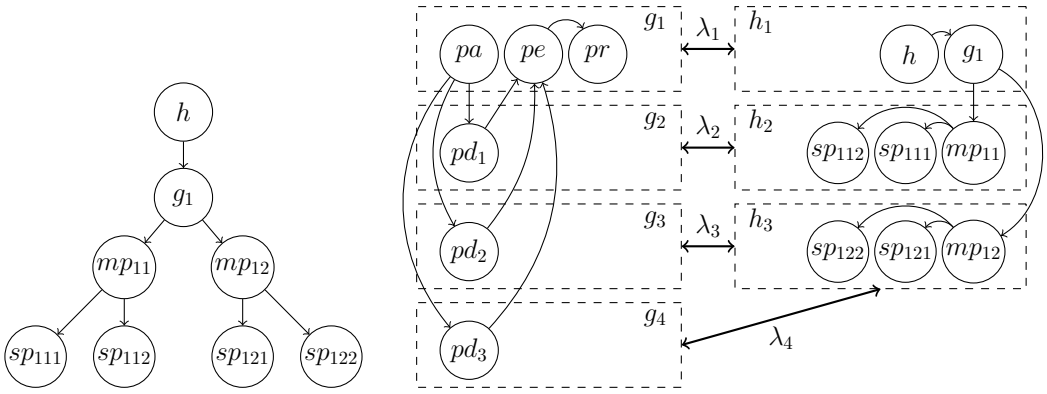
Постановка задачи структурно-параметрического согласования. Для фиксированной модели $\mathcal{M}$ ГПУ поставлена задача согласования как задача многокритериальной оптимизации

$$\min_{P \in D_P, T \in D_T} \Psi(P, T) = \Psi(P^*, T^*)\,, \tag{2}$$

где Ψ — векторный критерий оптимальности согласования; P^*, T^* — искомые оптимальные вектор свободных параметров P и отображение соответственно; $\min^M$ — символ многокритериальной оптимизации. Минимизация по P_s реализует параметрическое согласование, а по P_p — структурное согласование.

Предложен векторный критерий оптимальности согласования

$$\Psi(P, T) = (E_A, \Phi_A, r_A)\,,$$



а) GS б) $\lambda(GA_p, GS)$

Рис. 2.
Пример структурного графа GS и вершинного отображения $\lambda(GA_p, GS)$

где компонента $E_A(P,T)$ оценивает параллельную эффективность реализации алгоритма $A(P)$; $\Phi_A(\cdot)$ — качество решения базовой задачи (1); $r_A(\cdot)$ — скорость сходимости алгоритма $A(P)$. При решении задачи параметрического согласования использованы компоненты Φ_A, r_A, а при решении задачи структурного согласования — E_A. В вычислительных экспериментах в качестве критериев $E_A(\cdot), \Phi_A(\cdot), r_A(\cdot)$ использованы время выполнения синтезированного алгоритма $A(P)$, достигнутое значение минимума целевой функции базовой задачи оптимизации, скорость сходимости алгоритма $A(P)$ соответственно.

Иерархический метод и алгоритм решения задачи согласования. В диссертации предложен *иерархический метод* MO решения задачи согласования (2) как совокупность трех методов:

• $\mathsf{MO_S}$ — параметрического согласования последовательного алгоритма $A_s(P)$ с архитектурой ГПУ $\mathcal{M}$;

• $\mathsf{MO_P}$ — структурного согласования параллельного алгоритма $A(P)$ с этой архитектурой;

• $\mathsf{MO_T}$ — поиска оптимального отображения алгоритма $A(P)$ на эту архитектуру.

Отличительной особенностью метода MO является декомпозиция задачи (2) на внешнюю задачу структурного согласования, решаемую методом $\mathsf{MO_P}$, и вложенную задачу поиска отображения, решаемую методом $\mathsf{MO_T}$ на каждой итерации внешней задачи. Это позволяет решать задачу согласования совместно с решением базовой задачи (1). Метод $\mathsf{MO_S}$ выполняется до решения базовой задачи оптимизации алгоритмом $A(P)$. Метод MO позволяет а) синтезировать оптимальный алгоритм $A(P^*)$, б) определить оптимальное отображение T^* этого алгоритма на ГПУ с архитектурой $\mathcal{M}$, в) решить базовую задачу оптимизации алгоритмом $A(P^*)$ на этом ГПУ.

В диссертации предложен *иерархический алгоритм* AO, реализующий иерархический метод MO. Алгоритм AO включает в себя алгоритмы $\mathsf{AO_S}$, $\mathsf{AO_P}$, $\mathsf{AO_T}$, соответствующие методам $\mathsf{MO_S}$, $\mathsf{MO_P}$, $\mathsf{MO_T}$, а также метод $\mathsf{AO_C}$ синтеза

программы оптимизации по заданному отображению.

До начала выполнения алгоритма $A(P)$ алгоритм $\mathsf{AO_S}$ определяет значение вектора свободных параметров P_s с помощью метода статической установки параметров. Затем осуществляется итерационное решение базовой задачи алгоритмом $A(P)$. На каждой итерации t значение вектора свободных параметров P предоставляется алгоритмом $\mathsf{AO_P}$, осуществляющим динамическую адаптацию вектора этих параметров на основании текущих значений критериев оптимальности Φ_A, r_A. На каждой итерации t алгоритм $\mathsf{AO_T}$ выполняет поиск оптимального отображения T алгоритма $A(P)$ на архитектуру $\mathcal{M}$ ГПУ при фиксированном значении P. На основании полученного отображения T синтезируется базовая программа оптимизации при помощи алгоритма $\mathsf{AO_C}$.

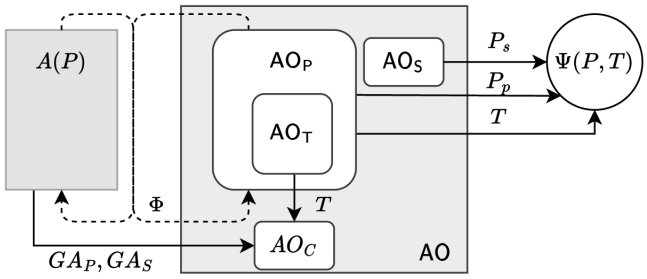


Рис. 3.
Схема совместного выполнения алгоритма оптимизации $A(P)$ и иерархического алгоритма согласования AO

На Рис. 3 изображена схема совместного выполнения указанных алгоритмов, где прерывистыми стрелками обозначены совместные итерации алгоритмов, а сплошными стрелками — источники значений компонентов P_s, P_p, T для вычисления значений векторного критерия оптимальности $\Psi(P, T)$.

Метод $\mathsf{MO_S}$ *и алгоритм* $\mathsf{AO_S}$. Назначение метода $\mathsf{MO_S}$ состоит в поиске оптимального значения P_s^* базового МЭГО-алгоритма $A_s(P_s)$ путем настройки свободных параметров. Метод выполняется однократно до начала выполнения алгоритма оптимизации $A(P)$.

В диссертации предложен алгоритм $\mathsf{AO_S}$, реализующий метод $\mathsf{MO_S}$ и основанный на решении задачи однокритериальной оптимизации

$$\min_{P_s \in D_{P_s}} \Phi_S(P_s) = \Phi_S(P_s^*)\,,$$

где скалярный критерий $\Phi_S(P_s)$ получен сверткой критериев Φ_A, r_A. Значения этих критериев можно вычислить до выполнения алгоритма $A(P)$, что позволяет исполнить алгоритм $\mathsf{AO_S}$ до начала выполнения алгоритма $A(P)$.

Метод $\mathsf{MO_P}$ *и алгоритм* $\mathsf{AO_P}$. Метод $\mathsf{MO_P}$ предназначен для адаптации вектора P_p структурных параметров алгоритма $A(P)$ в процессе решения базовой задачи этим алгоритмом. Адаптация осуществляется в процессе итераций алгоритма $A(P)$ на основе оценок текущих значений критериев Φ_A, r_A, что позволяет получить для алгоритма $A(P)$ оптимальную динамическую стратегию. Метод использует фиксированные вектор P_s и отображение T.

В алгоритме $\mathsf{AO_P}$ для компоненты $p_{p,i}$ вектора P_p процесс динамической

адаптации определяет формула

$$p_{p,i}^{t+1} = F_i(p_{p,i}^t, t, \Phi_A(P^t, T^t), r_A(P^t, T^t)), \quad (3)$$

где *функция адаптации* F_i задает правила перехода к значению $p_{p,i}^{t+1}$, исходя из номера итерации t, текущего значения $p_{p,i}^t$ и текущих значений критериев Φ_A, r_A. В диссертации предложена функция адаптации вида

$$F_i(p_{p,i}^t, t, \Phi_A(\cdot), r_A(\cdot)) = B_i(t) + \sum_{k=0}^{N_\tau} R_i(k, t, p_{p,i}^t, \Phi_A(\cdot), r_A(\cdot)), \quad (4)$$

где B_i — монотонная составляющая; R_i — реактивные составляющие. Составляющая B_i в формуле (4) обеспечивает монотонное изменение значений параметра $p_{p,i}$ в зависимости от t. Например, с ростом t эта составляющая реализует уменьшение числа N_S островов и увеличение длительности N_T фазы вычислений, что позволяет улучшить диверсификационные свойства алгоритма $A(P)$ и сократить число испытаний целевой функции базовой задачи.

Реактивные составляющие в формуле (4) обеспечивают адаптацию параметра $p_{p,i}$ на основании изменения значений величины $r_A(\cdot)$. Число реактивных составляющих N_τ в формуле (4) определено числом *узловых точек реакции* — номеров итераций, на которых к функциям адаптации добавляются затухающие составляющие, позволяющие предотвращать стагнацию оптимизации.

Функции адаптации (3) в общем случае различны для каждого из параметров $p_{p,i} \in P_p$. На Рис. 4 приведен пример функции адаптации F_1 и ее составляющих для параметра $p_{p,1} = N_S$.

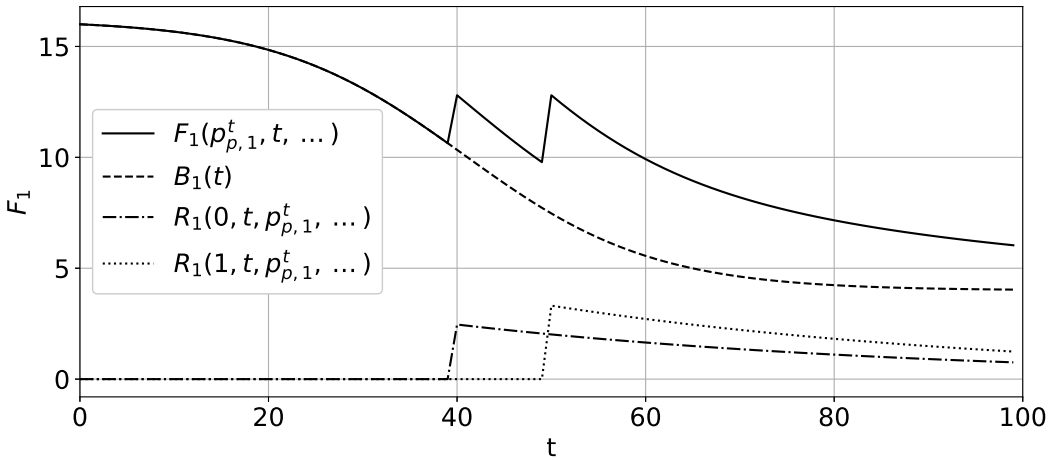


Рис. 4.

Пример функции адаптации F_1 и ее составляющих B_1, R_1

Схема итерации t алгоритма $\mathsf{AO_P}$ имеет следующий вид.

1) На основе графовых моделей $GA(P)$, $\mathcal{M}$, отображения T^t и вектора свободных параметров P^t синтезируется базовая программа оптимизации алгоритмом $\mathsf{AO_C}$.

2) Выполняется итерация алгоритма $A(P)$, включающая фазу выполнения алгоритма A_s на всех островах, фазы коммуникации и синхронизации между островами. Определяются значения критериев оптимальности Φ_A, r_A.

3) Алгоритмом $\mathsf{AO_P}$ решается задача структурного согласования алгоритма $A(P)$ с архитектурой $\mathcal{M}$ ГПУ при фиксированном отображении T^t —

определяются значения компонент вектора P^{t+1}.

4) Вложенным алгоритмом $\mathsf{AO_T}$ отыскивается оптимальное отображение T^{t+1} алгоритма на ГПУ для данного значения параметров P^{t+1}.

Метод $\mathsf{MO_T}$ *и алгоритм* $\mathsf{AO_T}$. Суть метода $\mathsf{MO_T}$ состоит в поиске оптимального отображения T^* алгоритма $A(P)$ на архитектуру ГПУ $\mathcal{M}$, минимизирующего время выполнения алгоритма на этом ГПУ при фиксированном векторе P. Метод основан на решении задачи дискретной оптимизации

$$\min_{T \in D_T} F_T(P, T) = F_T(P, T^*)\,, \tag{5}$$

где критерий оптимальности $F_T(P, T)$ формализует оценку параллельной эффективности отображения T. Для решения задачи (5) в диссертации предложен алгоритм $\mathsf{AO_T}$, включающий в себя следующие алгоритмы (Рис. 5).

- $\mathsf{AO_{TI}}$ — отыскание множества D_T отображений;
- $\mathsf{AO_{TF}}$ — расчет значений критерия оптимальности $F_T(T)$;
- $\mathsf{AO_{TNEXT}}$ — переход к следующему отображению из множества D_T.

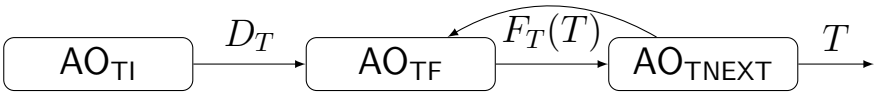


Рис. 5.
Схема алгоритма $\mathsf{AO_T}$

Алгоритм $\mathsf{AO_{TI}}$ отыскивает множество подграфов $\{\acute{H}\}$ графа H, изоморфных графу алгоритма GA, и основан на алгоритме Мессмера (Messmer, Bunke, 2000) поиска изоморфных подграфов, в котором по графу H строится дерево принятия решений τ_H, а для каждого входного графа GA осуществляется поиск по этому дереву. Каждому такому подграфу $\acute{H}$, определенному перестановкой PV вершин графа GA, соответствует совокупность T-отображений. Предложенный в диссертации алгоритм позволяет снизить сложность обхода дерева τ_H путем проверки в каждом его узле значения ограничивающей функции S и сократить размер дерева τ_H путем исключения графов, изоморфных с H, и отыскания неподвижных точек перестановки PV на основании особенностей архитектуры ГПУ, формализованных графом $\mathcal{M}$.

Алгоритм $\mathsf{AO_{TF}}$ реализует вычисление значений критерия оптимальности $F_T(T) = F_E(T) + \alpha J(\acute{T}, T)$, исходя из значения *масштабированного времени выполнения* $F_E(\cdot)$ алгоритма $A(P)$ на ГПУ, значения штрафной функции $J(\cdot)$ с коэффициентом штрафа α. Значение функции $J(T) = d_T^2(\acute{T}, T)$ вычисляется на основании *функции инерционности* d_T, реализующей оценку вычислительной сложности перехода от отображения $\acute{T}$ к отображению T. Алгоритм использует *историю отображений* $\mathcal{H}$ из экспериментальных оценок величины E_A. Время $F_E(\cdot)$ для отображений, не содержащихся в $\mathcal{H}$, вычисляется по аналитической оценке критерия E_A и коэффициенту масштабирования, учитывающего различие в экспериментальной и аналитической оценках величины E_A. Аналитическая оценку величины E_A определяется как вес гамильтонова пути S_T во взвешенном графе GA^*, построенном на основе гра-

фов $GA_s(P_s), GA_p(P)$ путем замены вершин $pd_i \in PD$ на графы $GA_s(P_s)$ и раскрытия N_T итераций алгоритма $A(P)$. Веса вершин определяет вычислительная сложность операций алгоритма, а веса ребер — отображение T.

Алгоритм $\mathsf{AO}_{\mathsf{TNEXT}}$ реализует выбор отображения T с вероятностью $\tilde{p}(T)$ при выполнении условия $F_T(T) < F_T(T^*)$, где T^* — лучшее из найденных отображений в истории отображений $\mathcal{H}$.

В четвертой главе приводится структура программной системы, реализующей предложенные и разработанные в диссертации математические модели, методы и алгоритмы. Представлены результаты исследования эффективности системы, а также результаты ее применения для решения тестовых и практических задач глобальной оптимизации.

Программная система реализована на языке программирования C++17 с применением библиотеки Nvidia CUDA и имеет объем около 18 тыс. строк. Система использует специализированный синтаксический анализатор на основе LLVM/Clang. Структура системы представлена на Рис. 6.

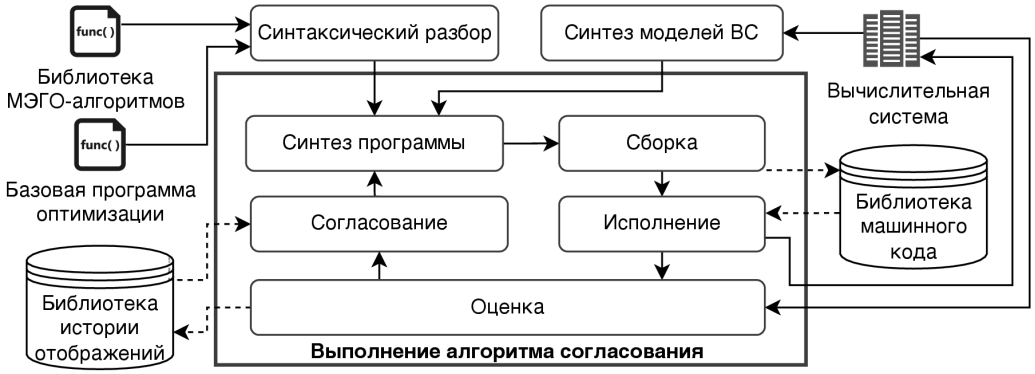


Рис. 6.
Структура программной системы

Программная система содержит библиотеку базовых МЭГО-алгоритмов на *МЭГО-языке* — расширении языка программирования C++, позволяющем связать графовые модели параллельного МЭГО-алгоритма $A(P)$ с аннотированными *программными функциями* этого языка, реализующими все операции алгоритмов $A(P)$, $A_s(P)$ и коммуникации между ними. Для каждой операции алгоритма $A(P)$ задаются независимые программные реализации на МЭГО-языке для каждого элемента множества $(O_s \cup O_p) \times TD \times TS \times TM$, где множество TS показывает вид коммуникации между операциями алгоритма (например, коммуникация между хост-системой и ГПУ); TD — размещение функции на ГПУ или хост-системе; TM — способ обращения к памяти из функции (например, обращение к глобальной памяти). Все варианты программных реализаций имеют аннотации элементов этого множества, а конкретный вариант выбирается алгоритмом AO_{C} исходя из отображения T.

Модуль «Синтаксический разбор» осуществляет построение и анализ абстрактного синтаксического дерева программ на МЭГО-языке, а также разбор кода графовых моделей алгоритмов на языке DOT. Модуль «Синтез моделей

вычислительной системы» синтезирует графовые модели заданной системы. Модуль «Синтез программы» синтезирует программу оптимизации алгоритмом $\mathsf{AO_C}$, основанным на обходе графа $GA_p^{(N_T)}$. Для каждой вершины пути $v_k \in S_T$ автоматически синтезируется вызов функции, реализующей операцию алгоритма v_k на МЭГО-языке и имеющей определяемые из T-отображения значения аннотаций. Для каждого ребра графа синтезируется вызов функции коммуникации между операциями v_k, v_{k+1}. Модуль «Сборка» осуществляет трансляцию синтезированной программы в динамическую библиотеку, сохраняемую в *библиотеку машинного кода*, чтобы избежать повторного синтеза и сборки программы. Модуль «Исполнение» реализует загрузку и исполнение синтезированной программы на хост-системе и ГПУ. Модуль «Согласование» осуществляет решение задачи структурно-параметрического согласования алгоритмом AO с использованием *библиотеки истории отображений* $\mathcal{H}$. Модуль «Оценка» определяет значение компонентов критерия оптимальности $\Psi(P,T)$ по результатам выполнения синтезированной программы.

Исследование эффективности разработанного математического и программного обеспечения выполнено на стандартных тестовых функциях глобальной оптимизации с использованием хост-системы с процессором Intel Xeon Broadwell E5-2686 и ГПУ Nvidia Tesla K80. Сравнивается эффективность

• предложенной программной системы, реализующей синтезированный МЭГО-алгоритм (обозначается далее как PAGOS-алгоритм) и использующей алгоритмы структурно-параметрического согласования МЭГО-алгоритма с архитектурой вычислительной системы;

• классической программной реализации параллельного МЭГО-алгоритма, обозначаемого как CLASSIC-алгоритм и отличающегося от PAGOS-алгоритма фиксированными структурой и отображением.

В качестве базового МЭГО-алгоритма оптимизации для обеих систем используется канонический алгоритм PSO. Применяется метод мультистарта с числом стартов, равным 30, и равномерно случайным выбором начальных векторов X базовой задачи, полагается размерность $d = \{20, 50\}$.

Использованы индикаторы эффективности задачи согласования:

• $\tilde{\Phi}_A^*$ — средний по мультистарту достигаемый минимум Φ_A базовой задачи оптимизации;

• $\tilde{E}_A^*$ — среднее по мультистарту время E_A решения базовой задачи;

• $\tilde{k}_\Phi$ — *коэффициент улучшения* (отношение средних по мультистарту значений минимумов в PAGOS- и CLASSIC-алгоритмах);

• $\tilde{k}_C$ — *коэффициент испытаний* (отношение средних значений чисел испытаний целевой функции базовой задачи в этих алгоритмах);

• S — ускорение (отношение времени решения базовой задачи в этих алгоритмах);

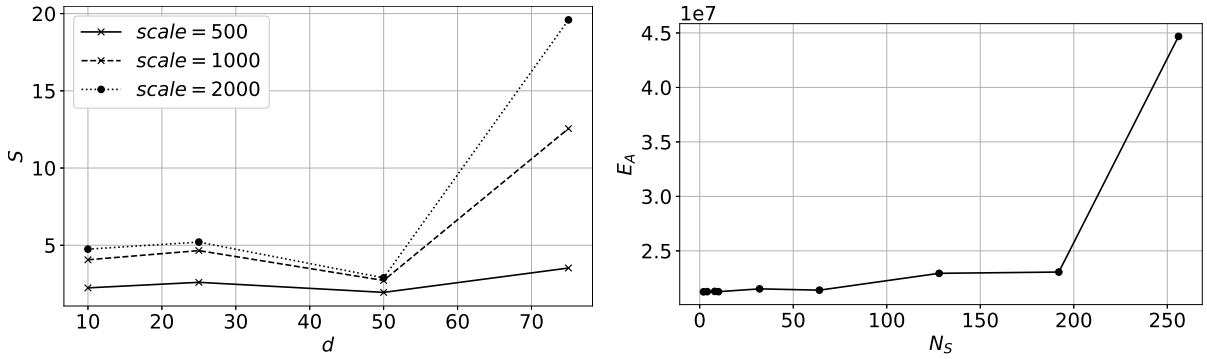
• $\hat{p}$ — оценка вероятности локализации Φ_A^* с заданной точностью.

В Таблице 1 приведены результаты вычислительных экспериментов для некоторых тестовых функций и $d = 50$. На Рис. 7,а приведена в качестве примера зависимость ускорения S от размерности d базовой задачи при различных значениях вычислительной сложности *scale* целевой функции Розенброка, на Рис. 7,б — зависимость значений критерия E_A от числа островов N_S, демонстрирующая масштабируемость PAGOS-алгоритма.

Таблица 1.

Результаты исследования эффективности для тестовых задач

Тестовая функция	Система	$\tilde{\Phi}^*_A$	c_{v,Φ_A}	$\tilde{E}^*_A$	c_{v,E_A}	$\hat{p}$	$\tilde{k}_C$	$\tilde{k}_\Phi$
Экли	CLASSIC	$4{,}82\cdot10^{-5}$	0,30	12,1	0,04	1,0	—	—
	PAGOS	$7{,}90\cdot10^{-6}$	0,18	5,1	0,06	1,0	3,9	6,1
Розенброка	CLASSIC	40,34	0,26	69,4	0,33	1,0	—	—
	PAGOS	0,93	1,81	61,4	0,26	1,0	1,1	43,4
Сфера	CLASSIC	$1{,}09\cdot10^{-5}$	0,34	11,1	0,06	1,0	—	—
	PAGOS	$5{,}75\cdot10^{-17}$	0,86	5,1	0,05	1,0	2,9	$1{,}90\cdot10^{11}$
Растригина	CLASSIC	396,36	0,07	4,7	0,14	1,0	—	—
	PAGOS	85,77	0,24	5,0	0,04	1,0	1,1	4,6
Швефеля	CLASSIC	40254,44	0,16	5,1	0,21	0,1	—	—
	PAGOS	$3{,}80\cdot10^{-5}$	0,40	34,3	0,09	1,0	3,0	$1{,}06\cdot10^{9}$
Гриванка	CLASSIC	0,02	5,38	10,9	0,08	1,0	—	—
	PAGOS	0,00	1,47	5,1	0,05	1,0	3,2	7,5



а) Зависимость ускорения S от размерности d

б) Зависимость E_A от числа островов N_S

Рис. 7.

Результаты вычислительных экспериментов по исследованию эффективности

На основании результатов экспериментов с тестовыми функциями высокой размерности ($d = 50$) сделаны следующие выводы.

1) Применение предложенной программной системы позволяет сократить общее время решения базовой задачи по сравнению с классической системой от 3 до 8 раз. Для различных тестовых функций ускорение S растет приблизительно линейно при увеличении вычислительной сложности *scale* целевой функции базовой задачи в диапазоне $[500; 2000]$.

2) PAGOS-алгоритм показывает на рассмотренных тестовых функциях

лучшую скорость сходимости, чем CLASSIC-алгоритм, со средним значением коэффициента улучшения $\tilde{k}_\Phi = 3{,}0$.

3) Значения коэффициента улучшения $\tilde{k}_\Phi$ находятся в диапазоне $[1, 44]$ и практически не зависят от размерности задачи. Для тестовой функции Швефеля CLASSIC-алгоритм неспособен предотвратить стагнацию оптимизации, тогда как PAGOS-алгоритм успешно локализует минимум этой функции.

4) Оценка вероятности $\hat{p}$ составляет не менее 1,0 только при использовании PAGOS-алгоритма.

5) Среднее по всем тестовым функциям значение коэффициента испытаний $\tilde{k}_C$ составляет 2,3 для размерности задачи $d = 20$ и 2,7 для $d = 50$.

6) PAGOS-алгоритм имеет значения индикатора эффективности $\tilde{E}_A^*$, меньшие до 2,2 раз по сравнению с CLASSIC-алгоритмом.

Практическая задача химической кинетики основана на результатах экспериментальных исследований реакции каталитического гидроалюминирования олефинов, выполненных Институтом нефтехимии и катализа (ИНК) РАН. Обратная параметрическая задача химической кинетики состоит в определении значений вектора кинетических констант X скоростей реакций веществ с концентрациями C на основе экспериментальных значений $\tilde{C}(t)$ концентраций в моменты времени t:

$$\min_{X \in D_X} \Phi(C, X) = \Phi(C, X^*)\,, \tag{6}$$

где $\Phi(C, X)$ — евклидова мера соответствия решений $C(t)$ значениям $\tilde{C}(t_k)$ в K экспериментальных точках $\{t_k\}$; D_X — область допустимых значений кинетических констант. На каждой итерации задача (6) для каждого значения вектора варьируемых параметров X требует решения прямой задачи химической кинетики, имеющей вид системы обыкновенных дифференциальных уравнений в нормальной форме Коши

$$\frac{dC}{dt} = \sum_{j=1}^{N_W} v_{ij} F_W(X_j, C)\,, t \in [0; T_R]\,, C(0) = C^0\,, \tag{7}$$

где T_R — общее время реакции; v — коэффициенты реакции; $F_W(X, C)$ — функция скорости реакции, определяемая исходя из схемы химических реакций.

Задача (7) является вычислительно сложной, поскольку требует многократного решения жесткой системы ОДУ (6). В вычислительных экспериментах принято $d = 10$, $K = 5$, $N_W = 9, |C| = 13$. В Таблице 2 приведены результаты вычислительных экспериментов для PAGOS- и CLASSIC-алгоритма.

Таблица 2.
Результаты вычислительного эксперимента для задачи химической кинетики

Алгоритм	$\tilde{\Phi}_A^*$	$\tilde{E}_A^*$	$\tilde{t}$	$\tilde{k}_C$	$\tilde{k}_\Phi$	$\tilde{S}$
CLASSIC	3,0	1032	4789	—	—	—
PAGOS	3,0	1590	1627	10,8	1,0	4,4

Результаты вычислительных экспериментов показывают, что программная система завершает процесс оптимизации при числе итераций в 2,9 раз меньшем,

чем классическая реализация, в 10,8 раз меньшем по числу оценок целевой функции базовой задачи и в 4 раза меньшем общем времени решения задачи. Среднее значение коэффициента испытаний $\tilde{k}_C$ составляет 10,8; ускорение $\tilde{S}$ — 4,4. Найдено решение со значением индикатора $\tilde{\Phi}_A^* = 3{,}0$, на 3% лучшим по сравнению с наилучшим известным решением.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ И ВЫВОДЫ

1) Предложены математическая модель параллельного МЭГО-алгоритма и математическая модель ГПУ, учитывающие особенности рассматриваемого класса параллельных алгоритмов и архитектуры ГПУ.

2) Формализована задача структурно-параметрического согласования МЭГО-алгоритма с архитектурой ГПУ как задача многокритериальной оптимизации на основе отображений групп вершин графов МЭГО-алгоритма и ГПУ, ограничений этих отображений и векторного критерия оптимальности задачи согласования.

3) Предложены иерархический метод и алгоритм структурно-параметрического согласования, отличающиеся совместным выполнением базового МЭГО-алгоритма, алгоритма динамической адаптации значений свободных параметров этого алгоритма и алгоритма его структурного согласования с архитектурой ГПУ, что позволяет одновременно синтезировать оптимальный МЭГО-алгоритм и определить его оптимальное отображение на ГПУ.

4) Предложен алгоритм поиска оптимального отображения МЭГО-алгоритма на архитектуру ГПУ, основанный на алгоритме поиска вершинных отображений произвольных рангов и алгоритме отыскания подграфовых изоморфизмов при помощи дерева решений.

5) Предложена структура программной системы, реализующей предложенные математические модели, методы и алгоритмы. Система осуществляет автоматический синтез алгоритма и программы оптимизации, а также исполнение динамических библиотек с этой программой на ГПУ.

6) Проведено исследование эффективности разработанного математического и программного обеспечения. На наборе стандартных тестовых функций показано, что синтезированный параллельный алгоритм превосходит базовый МЭГО-алгоритм по значению достигаемого экстремума целевой функции базовой задачи до 50 раз, по числу испытаний целевой функции — в 2,3 раза, по ускорению — от 3 до 8 раз.

7) Разработанное математическое и программное обеспечение подтвердило свою эффективность при решении практической задачи химической кинетики для реакции каталитического гидроалюминирования олефинов триэтилалюминием. Общее время решения базовой задачи сокращено более чем в 4 раза. Получено решение, на 3% лучшее по сравнению с наилучшим известным решением по критерию невязки экспериментальных и модельных значений скоростей реакции.

ПУБЛИКАЦИИ АВТОРА ПО ТЕМЕ ДИССЕРТАЦИИ

1. Селиверстов Е. Ю. Структурное согласование алгоритмов глобальной оптимизации с архитектурой графических процессоров // Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение. 2022. № 2. С. 42–59. (1,1 п.л.)
2. Воробьева Е. Ю., Карпенко А. П., Селиверстов Е. Ю. Ко-гибридизация алгоритмов роя частиц // Наука и образование: Электронное научное издание. 2012. № 4. С. 1–20. http://technomag.edu.ru/doc/355729.html (дата обращения: 16.12.15), (1,3 п.л./0,6 п.л.)
3. Карпенко А. П., Селиверстов Е. Ю. Глобальная оптимизация методом роя частиц. Обзор // Информационные технологии. 2010. № 2. С. 25–34. (0,63 п.л./0,4 п.л.)
4. Seliverstov E. A Hierarchical Method of Parameter Setting for Population-Based Metaheuristic Optimization Algorithms // Journal of Applied and Industrial Mathematics. 2022. Vol. 16, N. 4. P. 776–788. (0,9 п.л.)
5. Seliverstov E., Karpenko A. Hierarchical Model of Parallel Metaheuristic Optimization Algorithms // Procedia Computer Science. 2019. Vol. 150. С. 441–449. Proceedings of the 13th International Symposium Intelligent Systems 2018. (0,6 п.л./0,4 п.л.)
6. Карпенко А. П., Селиверстов Е. Ю. Глобальная оптимизация методом роя частиц на графических процессорах // Научный сервис в сети Интернет: масштабируемость, параллельность, эффективность: Труды Всероссийской суперкомпьютерной конференции. Издательство МГУ. 2009. С. 188–191. (0,25 п.л./0,13 п.л.)
7. Параллелизм в структурной и параметрической идентификации кинетических моделей химических реакций / Селиверстов Е. Ю. [и др.] // Научный сервис в сети Интернет: все грани параллелизма: Труды Всероссийской суперкомпьютерной конференции. Издательство МГУ. 2013. С. 268–273. (0,38 п.л./0,13 п.л.)
8. Селиверстов Е. Ю. Обзор методов решения задачи планирования параллельных алгоритмов // Инженерный вестник. Электронный научно-технический журнал. 2014. Т. 12. С. 541–555. http://ainjournal.ru/doc/746179.html (дата обращения: 12.06.15), (0,9 п.л.)
9. Селиверстов Е. Ю. Графовые модели графического процессора // Системы компьютерной математики и их приложения: Материалы XVIII Международной научной конференции. Издательство СмолГУ. 2017. С. 117–119. (0,1 п.л.)
10. Карпенко А. П., Селиверстов Е. Ю. Глобальная безусловная оптимизация роем частиц на графических процессорах архитектуры CUDA // Наука и образование: Электронное научное издание. 2010. № 4. С. 1–18. http://technomag.edu.ru/doc/142202.html (дата обращения: 01.03.11), (1,12 п.л./0,8 п.л.)