УДК 004.023

КВАНТОВО-ИНСПИРИРОВАННЫЙ ГЕНЕТИЧЕСКИЙ АЛГОРИТМ ВЕЩЕСТВЕННОЙ ОДНОКРИТЕРИАЛЬНОЙ ОПТИМИЗАЦИИ С АДАПТАЦИЕЙ ЗНАЧЕНИЙ ПАРАМЕТРОВ НА ОСНОВЕ ИСТОРИИ УСПЕХА

Масленников В.В.

Федеральное государственное бюджетное образовательное учреждение высшего образования «МИРЭА – Российский технологический университет», e-mail: maslennikov@mirea.ru

В работе предложен алгоритм вещественной однокритериальной оптимизации, объединяющий потенциал квантово-инспирированных вычислений с механизмом контекстно-зависимой самонастройки. В основе нового алгоритма лежит модифицированный квантово-инспирированный генетический алгоритм, использующий многоуровневые квантовые системы и физически обоснованную модель декогеренции, имитирующую условия эпохи NISQ. В отличие от существующих квантово-инспирированных алгоритмов оптимизации, основанных на ручной калибровке или фиксированных эвристиках, предложенный алгоритм автоматически регулирует ключевые управляющие параметры, такие как угол поворота в квантовых вентилях и вероятность мутации, на основе анализа собственного опыта с использованием истории успеха и взвешенного среднего Лемера второго порядка. Это позволяет динамически адаптировать баланс между глобальным исследованием и локальной эксплуатацией в соответствии с особенностями ландшафта целевой функции. Комплексная оценка на основе набора эталонных функций из бенчмарка по эволюционным вычислениям показала, что предложенный алгоритм обеспечивает высокую надёжность и устойчивость на мультимодальных, а также сложных гибридных и композитных функциях. Полученные результаты подчёркивают перспективность синтеза квантово-инспирированных оптимизационных моделей и адаптивных стратегий управления для создания робастных инструментов решения задач оптимизации в условиях «чёрного ящика».

Ключевые слова: квантово-инспирированный алгоритм, генетический алгоритм, однокритериальная оптимизация, кудит, адаптация значений параметров, самонастройка, история успеха.

QUANTUM-INSPIRED GENETIC ALGORITHM FOR REAL-VALUED SINGLE-OBJECTIVE OPTIMIZATION WITH PARAMETER ADAPTATION BASED ON SUCCESS HISTORY

Maslennikov V.V.

Federal State Budget Educational Institution of Higher Education «MIREA – Russian Technological University», e-mail: maslennikov@mirea.ru

This paper proposes a real-valued single-objective optimization algorithm that combines the potential of quantum-inspired computation with a context-aware self-tuning mechanism. The core of the new algorithm is a modified quantum-inspired genetic algorithm employing multi-level quantum systems and a physically grounded decoherence model that emulates the noisy intermediate-scale quantum era. In contrast to existing quantum-inspired optimization algorithms that rely on manual calibration or fixed heuristics, the proposed algorithm automatically adjusts key control parameters, such as the rotation angle in quantum gates and the mutation probability, by analyzing its own performance history using success-history adaptation combined with a second-order Lehmer weighted mean. This enables dynamic balancing between global exploration and local exploitation tailored to the characteristics of the objective function landscape. Comprehensive evaluation on a suite of benchmark functions from the evolutionary computation benchmark set demonstrates that the proposed algorithm achieves high reliability and robustness on multimodal, as well as complex hybrid and composite functions. The results highlight the promise of integrating quantum-inspired optimization models with adaptive control strategies to develop robust black-box optimization tools.

Keywords: quantum-inspired algorithm, genetic algorithm, single-objective optimization, qudit, parameter adaptation, self-adaptation, success history.

Введение

Эволюционные алгоритмы уже несколько десятилетий остаются одними из самых мощных и универсальных инструментов для решения сложных задач оптимизации, особенно в условиях «чёрного ящика», когда аналитические свойства целевой функции неизвестны. Среди эволюционных алгоритмов широкое распространение получили генетические алгоритмы, что связано с их простотой, гибкостью и способностью эффективно исследовать сложные, многомерные и мультимодальные пространства решений. Однако производительность данных алгоритмов критически зависит от правильного выбора и настройки генетических операторов и их численных параметров [1, 2]. Ввиду отсутствия универсальных настроек, применимых ко всем задачам, что формально подтверждается теоремой «бесплатных завтраков не бывает» [3], значительное внимание исследователей в последние годы уделяется разработке специальных механизмов самонастройки и самоконфигурирования [4, 5], способствующих динамической адаптации параметров алгоритма или структуры операторов алгоритма в процессе поиска оптимальных решений. Среди таких подходов в научном сообществе особый акцент делается на механизмах, основанных на истории успеха (Success History Adaptation, далее – SHA), изначально предложенных для алгоритма дифференциальной эволюции и позже успешно перенесённых на генетические алгоритмы [6, 7]. Механизм SHA отслеживает, какие значения параметров привели к улучшению решений, и использует эту информацию для формирования новых значений, что обеспечивает гибкую и контекстно-зависимую настройку без необходимости вручную задавать сложные правила адаптации.

Параллельно подходам к динамической адаптации активно развивается направление квантово-инспирированных эволюционных алгоритмов (Quantum-Inspired Evolutionary Algorithms, далее – QIEA) [8-12], сочетающих в себе принципы квантовой механики – суперпозицию, запутанность и декогеренцию – с классическими эволюционными схемами. Такие алгоритмы демонстрируют повышенную способность к исследованию пространства решений и устойчивость к застреванию в локальных оптимумах [11], поскольку квантовое представление агентов позволяет одновременно кодировать и оценивать множество потенциальных решений. Однако их эффективность, как и у классических эволюционных алгоритмов, сильно зависит от корректности настройки внутренних параметров, таких как угол поворота в квантовых вентилях, частота и вероятность мутации, вероятность скрещивания и уровень декогеренции [11]. В большинстве существующих алгоритмов QIEA [8-12] эти параметры либо фиксируются априори, либо настраиваются с помощью простых эвристик, что ограничивает их универсальность и устойчивость при решении задач оптимизации с разной структурой и свойствами. Несмотря на очевидный синергетический потенциал, интеграция современных механизмов самонастройки, в частности SHA, в квантово-инспирированные алгоритмы является малоизученной областью, что создаёт значительный пробел между теоретическими возможностями алгоритмов QIEA и их практической эффективностью.

В настоящем исследовании предлагается квантово-инспирированный генетический алгоритм вещественной однокритериальной оптимизации, в котором впервые интегрирован механизм адаптации на основе истории успеха SHA для динамической настройки ключевых параметров квантовой эволюции. В отличие от существующих алгоритмов QIEA [8-12], в разработанном подходе угол поворота в квантовых вентилях и вероятность мутации адаптируются в реальном времени на основе эффективности предыдущих изменений, оцениваемой через улучшение значений целевой функции. Для повышения устойчивости адаптации используется среднее значение Лемера, учитывающее как успешность, так и уровень улучшения решений. Предложенный алгоритм сохраняет преимущества квантового представления – параллельную оценку множества решений и устойчивость к преждевременной сходимости – и одновременно получает способность к самоорганизации, что делает его применимым к широкому спектру задач без предварительной глобальной настройки. Экспериментальная оценка на тестовых задачах подтверждает, что новый алгоритм обеспечивает высокую надёжность и скорость сходимости, демонстрируя потенциал для практического применения в задачах численной оптимизации «чёрного ящика».

Текущие исследования

В контексте квантово-инспирированного моделирования для алгоритмов однокритериальной оптимизации в [12] предложен Quantum Snowflake Algorithm (далее – QSA) – гибридный метаэвристический алгоритм, сочетающий квантово-инспирированный туннельный эффект, суперпозицию частичных решений, репульсивные «столкновения» агентов и локальный поиск. QSA решает проблему сохранения разнообразия популяции за счёт применения сил отталкивания между решениями, а также обеспечивает выход из локальных минимумов благодаря механизму, имитирующему квантовое туннелирование. Вместе с тем следует отметить, что алгоритм не использует суперпозицию на уровне амплитуд вероятностей существования квантовых

состояний, матрицы плотности или квантовые вентили, а лишь заимствует отдельные концепции из теории квантовой механики. Кроме того, такие параметры алгоритма как температура, ширина барьера и коэффициенты отталкивания, требуют ручной настройки, что ограничивает его робастность при решении задач с сильно различающимися ландшафтами.

В работе [13] рассматривается Adaptive Hybrid Quantum-Inspired Optimization Algorithm (далее – AHQSOA). Данный алгоритм объединяет глобальный поиск на основе алгоритма Quantum Particle Swarm Optimization (далее – QPSO) [14] и локальную эксплуатацию с использованием операторов алгоритма Quantum Evolutionary Algorithm (далее – QEA) [15]. Ключевым преимуществом AHQSOA является встроенный механизм адаптивной настройки значений параметров, реализованный на основе обучения с подкреплением, который динамически балансирует между исследованием и эксплуатацией в зависимости от текущей стадии оптимизационного процесса. Однако алгоритм AHQSOA обладает значительным недостатком, который заключается в необходимости одновременного управления операторами из квантовых структур алгоритмов QPSO и QEA, а также в потребности использовать внешний агент обучения с подкреплением, что увеличивает вычислительные затраты и затрудняет анализ сходимости.

Исследование [16] предлагает квантово-инспирированный оптимизационный фреймворк для динамического распределения ресурсов в системах автомобильных граничных вычислений (Vehicular Edge Computing). Авторы моделируют задачу распределения ресурсов, таких как вычислительная мощность, пропускная способность и объём памяти, в виде задачи многокритериальной оптимизации, где целевая функция объединяет минимизацию задержки, энергопотребления и максимизацию справедливости распределения с использованием индекса Джини [17]. Ключевая особенность подхода, применяемого в исследовании, заключается в представлении множества возможных решений в виде квантового состояния суперпозиции, где каждое базисное состояние соответствует конкретной стратегии распределения, а амплитуда базисного состояния определяет вероятность выбора. Однако проведение экспериментов показало, что при использовании такого подхода происходит резкое снижение возможности масштабирования при увеличении числа автомобилей свыше 50, что ставит под сомнение применимость подхода в крупномасштабных городских сценариях. Также необходимо отметить, что обновление амплитуд осуществляется детерминировано на основе значений функции приспособленности, без учёта динамической настройки параметров, вследствие чего снижается способность оперативно реагировать на изменения в распределении ресурсов и требованиях задач в условиях сильной динамики автомобильных сетей.

В работах [4, 5] описываются методы самоконфигурирования эволюционных алгоритмов Self-Configuring Evolutionary Algorithm (далее – SelfCEA) и Population-Level Dynamic Probabilities Evolutionary Algorithm (далее – PDP), а также в соответствии методами реализуются самоконфигурируемые генетические алгоритмы с адаптацией на основе истории успеха Self-Configuring Success-History based Adaptive Genetic Programming (далее – SelfCSHAGP) и Population-Level Dynamic Probabilities Success-History based Adaptive Genetic Programming (далее – PDPSHAGP), изначально разработанные для задач генетического программирования и символьной регрессии, но обладающие высокой степенью обобщаемости и применимые к широкому классу задач численной оптимизации. Главной особенностью рассматриваемого в [4, 5] подхода является глубокая интеграция двух уровней самонастройки – адаптации значений численных параметров посредством механизма Success-History based Parameter Adaptation, заимствованного из дифференциальной эволюции, и динамического самоконфигурирования типов генетических операторов (селекции, скрещивания и мутации) с использованием методов SelfCEA и PDP. Особое внимание уделяется модификации оператора скрещивания: он поддерживает многородительскую рекомбинацию, селективное давление на этапе выбора генов и адаптивную интенсивность, что обеспечивает гибкий баланс между исследованием и эксплуатацией. Эксперименты, проведённые на базе набора из 120 физических уравнений, показали, что оба алгоритма превосходят классические методы самонастройки генетического программирования более чем в 90% задач. При этом алгоритмы демонстрируют высокую эффективность в решении инженерных задач, связанных с проектированием систем управления летательными аппаратами и автоматизацией машинного обучения, где критически важны надёжность и способность находить точные аналитические зависимости в сложных многомерных пространствах. Однако такая гибкость достигается значительной архитектурной сложностью: алгоритмы поддерживают до 160 конфигураций операторов, что усложняет их анализ, настройку значений параметров и воспроизводимость результатов.

Описание алгоритма

В настоящем исследовании в основе предлагаемого подхода лежит модифицированный квантово-инспирированный генетический алгоритм вещественной однокритериальной оптимизации (Modified Quantum-

Inspired Genetic Algorithm, далее – MQIGA), впервые представленный в работе [11]. Исходная версия MQIGA реализует однокритериальную оптимизацию на основе квантово-инспирированного моделирования с использованием многоуровневых квантовых систем, называемых кудитами, и имитацией процесса квантовой декогеренции. Алгоритм сохраняет ключевые преимущества квантовых вычислений, такие как суперпозиция состояний и параллельная оценка множества решений, и при этом полностью реализуется на классическом вычислительном оборудовании. В данной работе к базовой архитектуре MQIGA добавляется механизм адаптации значений параметров на основе истории успеха SHA, позволяющий динамически настраивать угол поворота в квантовых вентилях и вероятность мутации в зависимости от эффективности предыдущих изменений, тем самым повышая робастность и скорость сходимости алгоритма без потребности в ручной настройке.

Алгоритм MQIGA реализует эволюционный процесс в квантово-инспирированном пространстве, где каждый агент популяции представлен не классической бинарной или вещественной хромосомой, а квантовой хромосомой, состоящей из последовательности кудитов – многоуровневых квантовых систем с размерностью $d \geq 3$:

$$|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle + \alpha_2|2\rangle + \alpha_3|3\rangle + \cdots + \alpha_{d-1}|d-1\rangle = \sum_{j=0}^{d-1} \alpha_j|j\rangle = (\alpha_0, \alpha_1, \alpha_2, \alpha_3, \dots, \alpha_{d-1})^T \in \mathbb{C}^d, \quad (1)$$

где $|\psi\rangle$ – квантовая система, находящаяся одновременно в более, чем двух состояниях; $|j\rangle$ – ортонормированный базисный вектор; $\alpha_0 \dots \alpha_{d-1}$ – амплитуды вероятностей при $\alpha_0 \dots \alpha_{d-1} \in \mathbb{C}^d$ и $|\alpha_0|^2 + |\alpha_1|^2 + |\alpha_2|^2 + |\alpha_3|^2 + \cdots + |\alpha_{d-1}|^2 = 1$.

В отличие от традиционных кубитов ($d = 2$), кудиты позволяют кодировать суперпозицию d базисных состояний одновременно, что значительно повышает параллелизм поиска потенциальных решений. Каждый кудит в хромосоме описывается комплексным вектором состояния или, что более корректно в условиях декогеренции, матрицей плотности, формируемой на основе операторов группы Гейзенберга-Вейля. Эволюция популяции осуществляется через последовательное применение специализированных операторов: инициализации с использованием обобщённого вентиля Адамара, скрещивания, реализующего когерентное смешивание состояний лучшего агента и текущего кандидата, и мутации, вводящей случайные возмущения через применение случайных операторов Гейзенберга-Вейля и генерацию новых квантовых состояний. Ключевой особенностью MQIGA является процедура измерения с учётом шума, имитирующая влияние декогеренции в эпоху Noisy Intermediate-Scale Quantum (далее – NISQ) [18]: перед измерением состояние каждого кудита преобразуется в матрицу плотности, к которой применяется стохастический оператор Гейзенберга-Вейля с заданным уровнем ошибки. Это позволяет моделировать реалистичное взаимодействие квантовой системы с окружением, что способствует поддержанию разнообразия популяции и снижает риск преждевременной сходимости. После измерения квантовая популяция преобразуется в классическую форму, оценивается с помощью функции приспособленности, а затем на основе полученных значений обновляются квантовые состояния с учётом элитизма и генетических операторов.

Архитектура алгоритма MQIGA реализует эволюционный цикл через последовательность строго формализованных этапов.

На первом этапе инициализация популяции осуществляется применением обобщённого вентиля Адамара к состоянию $|0\rangle$ каждого кудита, что создаёт равномерную суперпозицию всех возможных базисных состояний. Пусть $|0\rangle$ – основное состояние кудита размерности d, представленное вектором:

$$|0\rangle = (1, 0, 0, \dots, 0)^T \in \mathbb{C}^d. \quad (2)$$

Обобщённый вентиль Адамара для кудита размерности d определяется следующим образом:

$$H_d = \frac{1}{\sqrt{d}} \sum_{j=0}^{d-1} \sum_{k=0}^{d-1} e^{2\pi i jk/d} |j\rangle\langle k|, \quad (3)$$

где $j, k = \overline{0, d-1}$.

Тогда инициализированное квантовое состояние каждого гена (кудита) получается как:

$$|\psi\rangle_{init} = H_d|0\rangle = \frac{1}{\sqrt{d}} \sum_{k=0}^{d-1} |k\rangle. \quad (4)$$

Измерение с учётом декогеренции моделируется через переход к матрице плотности и её возмущение случайным оператором из группы Гейзенберга-Вейля, что имитирует влияние шума в эпоху NISQ. Пусть чистое квантовое состояние кудита размерности d задано вектором:

$$|\psi\rangle \in \mathbb{C}^d, \quad \||\psi\rangle\| = 1. \tag{5}$$

Матрица плотности, соответствующая состоянию (5), вычисляется как:

$$\boldsymbol{\rho} = |\psi\rangle\langle\psi|. \tag{6}$$

Тогда при воздействии возмущения окружающей среды на кудит матрица плотности изменяется с помощью случайно выбранного оператора из группы Гейзенберга-Вейля $\{\mathbf{W}_{m,n}\}_{m,n=0}^{d-1}$:

$$\boldsymbol{\rho}_{noisy} = (1-\varepsilon)\boldsymbol{\rho} + \varepsilon \mathbf{W}_{m,n}\boldsymbol{\rho}\mathbf{W}_{m,n}^{\dagger}, \tag{7}$$

где $\mathbf{W}_{m,n} = X^m Z^n$ – элемент группы Гейзенберга-Вейля, X и Z – обобщённые операторы Паули для кудита размерности d; $\mathbf{W}_{m,n}^{\dagger}$ – эрмитово-сопряжённый оператор; $\varepsilon \in [0,1]$ – уровень ошибки.

Вероятности исходов измерения определяются диагональными элементами возмущённой матрицы плотности:

$$p_k = \langle k|\boldsymbol{\rho}_{noisy}|k\rangle = \left(\boldsymbol{\rho}_{noisy}\right)_{kk}, \tag{8}$$

где $k = \overline{0, d-1}$.

После этого выполняется преобразование квантовой хромосомы в классическое решение посредством выбора индекса k с вероятностью p_k, пропорциональной квадрату модуля амплитуды в соответствии с правилом Борна [19], с последующим отображением в вещественное пространство через линейную нормализацию. Для описания данного процесса определяется, что квантовая хромосома агента состоит из L кудитов размерности d каждый, и в результате измерения кудитов с учётом декогеренции получается последовательность индексов:

$$\mathbf{c} = (c_1, c_2, c_3, \dots, c_L), \tag{9}$$

где $c_j = \{0, 1, 2, \dots, d-1\}$ при $j = \overline{1, L}$ – индекс j-го гена в классической хромосоме, то есть результат измерения j-го кудита.

Последовательность (9) интерпретируется как число в позиционной системе счисления по основанию d, что даёт десятичное значение:

$$x_{dec} = \sum_{j=1}^{L} c_j d^{L-j}, \tag{10}$$

где $x_{dec} \in [0, d^L - 1]$.

Далее осуществляется линейная нормализация значения (10) в заданный для конкретной функции приспособленности диапазон поиска $[x_{min}, x_{max}]$:

$$x_{norm} = x_{min} + \frac{x_{dec}}{d^L - 1}(x_{max} - x_{min}). \tag{11}$$

Значение x_{norm} интерпретируется как кандидатное решение задачи однокритериальной оптимизации и подаётся на вход функции приспособленности. Такой подход обеспечивает биективное отображение между пространством квантовых состояний и непрерывным интервалом поиска, сохраняя при этом разрешающую способность, определяемую длиной хромосомы L и размерностью кудита d.

Оператор скрещивания в алгоритме MQIGA реализует когерентное смешивание квантовых состояний лучшего агента популяции и текущего кандидата с целью передачи перспективных решений с сохранением квантовой суперпозиции. Пусть $\boldsymbol{\rho}_i^{(j)}$ – матрица плотности, описывающая состояние $|\psi_i^{(j)}\rangle$ j-го кудита i-го агента, а $|\psi_{best}^{(j)}\rangle$ – состояние глобально лучшего агента в текущем поколении. Для выполнения когерентного смешивания сначала из матрицы плотности извлекается чистое состояние, после чего формируется новое состояние:

$$|\psi_i^{(j)\prime}\rangle = \cos\left(\frac{\theta_{shift}^{(j)}}{2}\right)|\psi_{best}^{(j)}\rangle + \sin\left(\frac{\theta_{shift}^{(j)}}{2}\right)|\psi_i^{(j)}\rangle, \tag{12}$$

где $\theta_{shift}^{(j)} \in [0, 2\pi]$ – параметр, задающий угол поворота на обобщённой сфере Блоха [11] для j-го кудита.

После нормализации квантовое состояние обновляется и преобразуется обратно в матрицу плотности в соответствии с уравнением (6):

$$\left|\psi_i^{(j)'}\right\rangle \leftarrow \frac{\left|\psi_i^{(j)'}\right\rangle}{\left\|\left|\psi_i^{(j)'}\right\rangle\right\|}, \tag{13}$$

$$\boldsymbol{\rho}_i^{(j)'} = \left|\psi_i^{(j)'}\right\rangle\left\langle\psi_i^{(j)'}\right|. \tag{14}$$

Затем к получившейся матрице плотности применяется модель декогеренции в соответствии с уравнением (7):

$$\boldsymbol{\rho}_i^{(j)}{}_{cross} = (1-\varepsilon)\boldsymbol{\rho}_i^{(j)'} + \varepsilon \mathbf{W}_{m,n}\boldsymbol{\rho}_i^{(j)'}\mathbf{W}_{m,n}^{\dagger}. \tag{15}$$

Такая форма скрещивания обеспечивает плавный переход между состояниями родителей и позволяет управлять степенью влияния лучшего агента на потомка: при $\theta_{shift}^{(j)} = 0$ потомок наследует состояние лучшего агента, а при $\theta_{shift}^{(j)} = \pi$ – сохраняет своё исходное состояние.

Оператор мутации вводит стохастические возмущения, способствующие исследованию новых областей пространства решений и предотвращающие преждевременную сходимость. С этой целью к промежуточному состоянию кудита $\left|\psi_i^{(j)'}\right\rangle$ добавляется случайное квантовое состояние $\left|\phi^{(j)}\right\rangle$, сгенерированное с равномерным распределением на единичной сфере в $\mathbb{C}^d$. Результирующее состояние после мутации определяется как:

$$\left|\psi_i^{(j)''}\right\rangle = \cos\left(\frac{\theta_{rot}^{(j)}}{2}\right)\left|\psi_i^{(j)'}\right\rangle + \sin\left(\frac{\theta_{rot}^{(j)}}{2}\right)\left|\phi^{(j)}\right\rangle, \tag{16}$$

где $\theta_{rot}^{(j)} \in [0, 2\pi]$ – параметр, регулирующий интенсивность мутации.

Далее выполняется нормализация состояния $\left|\psi_i^{(j)''}\right\rangle$ по аналогии с (13), и формируется соответствующая матрица плотности по аналогии с (14):

$$\left|\psi_i^{(j)''}\right\rangle \leftarrow \frac{\left|\psi_i^{(j)''}\right\rangle}{\left\|\left|\psi_i^{(j)''}\right\rangle\right\|}, \tag{17}$$

$$\boldsymbol{\rho}_i^{(j)''} = \left|\psi_i^{(j)''}\right\rangle\left\langle\psi_i^{(j)''}\right|. \tag{18}$$

Затем аналогично уравнению (15) формируется новая матрица плотности, определяемая воздействием декогеренции:

$$\boldsymbol{\rho}_i^{(j)}{}_{mut} = (1-\varepsilon)\boldsymbol{\rho}_i^{(j)''} + \varepsilon \mathbf{W}_{m,n}\boldsymbol{\rho}_i^{(j)''}\mathbf{W}_{m,n}^{\dagger}. \tag{19}$$

Таким образом, мутация сохраняет когерентную структуру квантового пространства за счёт унитарных преобразований, что повышает разнообразие популяции и устойчивость к локальным минимумам.

Общий псевдокод алгоритма MQIGA приведён в алгоритме 1.

Алгоритм 1. MQIGA.

Вход: $N \in \mathbb{N}$ ▷ размер популяции.
$L \in \mathbb{N}$ ▷ длина квантовой хромосомы (количество кудитов).
$d \in \mathbb{N}$ ▷ размерность кудита.
$T_{max} \in \mathbb{N}$ ▷ максимальное количество поколений.
$\varepsilon \in [0, 1]$ ▷ уровень ошибки (шума) для модели декогеренции.
$P_{mut} \in [0, 1]$ ▷ вероятность мутации агента.
$f: \mathbb{R}^v \to \mathbb{R}$ ▷ целевая функция приспособленности, где $v \in \mathbb{N}$ – размерность пространства поиска.
$[x_{min}, x_{max}] \subset \mathbb{R}$, $x_{min} < x_{max}$ ▷ диапазон значений для поиска оптимального решения целевой функции.
$\wp_{fit} = +\infty$ ▷ переменная, хранящая лучшее значение функции приспособленности.
$\wp_{best} = \emptyset$ ▷ переменная, хранящая лучшее решение функции приспособленности.

1. **Для каждого** агента $i = \overline{1, N}$ **выполнять:**
2. **Для каждого** кудита $j = \overline{1, L}$ **выполнять:**
3. $|0\rangle = (1, 0, 0, \ldots, 0)^T \in \mathbb{C}^d$ ▷ определить начальное квантовое состояние.
4. $\left|\psi_i^{(j)}\right\rangle_{init} = H_d|0\rangle$ ▷ инициализировать состояние суперпозиции.
5. $\boldsymbol{\rho}_i^{(j)} = \left|\psi_i^{(j)}\right\rangle_{init}\left\langle\psi_i^{(j)}\right|_{init}$ ▷ создать матрицу плотности.

Завершить цикл.
Завершить цикл.
Установить номер поколения $t = 0$.
Пока $t < T_{max}$ **выполнять:**
Для каждого агента $i = \overline{1, N}$ **выполнять:**
Для каждого кудита $j = \overline{1, L}$ **выполнять:**
$\boldsymbol{\rho}_i^{(j)}{}_{noisy} = (1 - \varepsilon)\boldsymbol{\rho}_i^{(j)} + \varepsilon \mathbf{W}_{m,n}\boldsymbol{\rho}_i^{(j)}\mathbf{W}_{m,n}^{\dagger}$ ▷ изменить матрицу плотности воздействием оператора
из группы Гейзенберга-Вейля $\{\mathbf{W}_{m,n}\}_{m,n=0}^{d-1}$.
$p_k = \left\langle k \middle| \boldsymbol{\rho}_i^{(j)}{}_{noisy} \middle| k \right\rangle = \left(\boldsymbol{\rho}_i^{(j)}{}_{noisy}\right)_{kk}$ ▷ вычислить вероятность исхода измерения,
где $k = \overline{0, d-1}$.
Выбрать индекс $c_j = k$ с вероятностью p_k.
Завершить цикл.
$\mathbf{c} = (c_1, c_2, c_3, \dots, c_L)$ ▷ последовательность индексов кудитов i-го агента.
$x_{dec} = \sum_{j=1}^{L} c_j d^{L-j}$ ▷ вычислить десятичное значение на основе $c_j \in \mathbf{c}$.
$x_{norm} = x_{min} + \frac{x_{dec}}{d^L - 1}(x_{max} - x_{min})$ ▷ выполнить линейную нормализацию.
$f_i = f(x_{norm})$ ▷ оценить функцию приспособленности.
Если $f_i < \wp_{fit}$ **выполнить:**
$\wp_{fit} \leftarrow f_i$ ▷ обновить значение переменной.
$\wp_{best} \leftarrow \mathbf{c}$ ▷ обновить значение переменной.
Завершить цикл.
Для каждого агента $i = \overline{1, N}$ **выполнять:**
Если $i \in \wp_{best}$, то **пропустить** поколение.
Для каждого кудита $j = \overline{1, L}$ **выполнять:**
Извлечь квантовое состояние $|\psi_i^{(j)}\rangle$ из $\boldsymbol{\rho}_i^{(j)}$.
$|\psi_i^{(j)\prime}\rangle = \cos\left(\frac{\theta_{shift}^{(j)}}{2}\right)|\psi_{best}^{(j)}\rangle + \sin\left(\frac{\theta_{shift}^{(j)}}{2}\right)|\psi_i^{(j)}\rangle$ ▷ выполнить когерентное смешивание, где
$\theta_{shift}^{(j)} \in [0, 2\pi]$, $|\psi_{best}^{(j)}\rangle$ – квантовое состояние глобально лучшего агента.
$|\psi_i^{(j)\prime}\rangle \leftarrow \frac{|\psi_i^{(j)\prime}\rangle}{\||\psi_i^{(j)\prime}\rangle\|}$ ▷ выполнить нормализацию квантового состояния.
$\boldsymbol{\rho}_i^{(j)\prime} = |\psi_i^{(j)\prime}\rangle\langle\psi_i^{(j)\prime}|$ ▷ преобразовать квантовое состояние в матрицу плотности.
$\boldsymbol{\rho}_i^{(j)}{}_{cross} = (1 - \varepsilon)\boldsymbol{\rho}_i^{(j)\prime} + \varepsilon \mathbf{W}_{m,n}\boldsymbol{\rho}_i^{(j)\prime}\mathbf{W}_{m,n}^{\dagger}$ ▷ применить модель декогеренции.
Завершить цикл.
Завершить цикл.
Для каждого агента $i = \overline{1, N}$ **выполнять:**
Если $i \in \wp_{best}$, то **пропустить** поколение.
Для каждого агента **генерировать** случайное значение $u_i \sim \mathcal{U}(0, 1)$.
Если $u_i \leq P_{mut}$ **выполнять:**
Для каждого кудита $j = \overline{1, L}$ **выполнять:**
Сгенерировать случайное состояние $|\phi^{(j)}\rangle$ на единичной сфере в $\mathbb{C}^d$.
$|\psi_i^{(j)\prime\prime}\rangle = \cos\left(\frac{\theta_{rot}^{(j)}}{2}\right)|\psi_i^{(j)\prime}\rangle + \sin\left(\frac{\theta_{rot}^{(j)}}{2}\right)|\phi^{(j)}\rangle$ ▷ вычислить состояние после мутации, где
$\theta_{rot}^{(j)} \in [0, 2\pi]$.
$|\psi_i^{(j)\prime\prime}\rangle \leftarrow \frac{|\psi_i^{(j)\prime\prime}\rangle}{\||\psi_i^{(j)\prime\prime}\rangle\|}$ ▷ выполнить нормализацию квантового состояния.
$\boldsymbol{\rho}_i^{(j)\prime\prime} = |\psi_i^{(j)\prime\prime}\rangle\langle\psi_i^{(j)\prime\prime}|$ ▷ преобразовать квантовое состояние в матрицу плотности.
$\boldsymbol{\rho}_i^{(j)}{}_{mut} = (1 - \varepsilon)\boldsymbol{\rho}_i^{(j)\prime\prime} + \varepsilon \mathbf{W}_{m,n}\boldsymbol{\rho}_i^{(j)\prime\prime}\mathbf{W}_{m,n}^{\dagger}$ ▷ применить модель декогеренции.
$\boldsymbol{\rho}_i^{(j)} \leftarrow \boldsymbol{\rho}_i^{(j)}{}_{mut}$ ▷ обновить матрицу плотности кудита.
Завершить цикл.
Иначе выполнить:
Для каждого кудита $j = \overline{1, L}$ **выполнять:**
$\boldsymbol{\rho}_i^{(j)} \leftarrow \boldsymbol{\rho}_i^{(j)}{}_{cross}$ ▷ обновить матрицу плотности кудита.

53. **Завершить цикл.**
54. **Завершить цикл.**
55. **Если** $\wp_{best} \neq \emptyset$ **выполнять:**
56. **Записать** $\wp_{best}$ в популяцию.
57. $t \leftarrow t + 1$ ▷ увеличить счётчик поколений.
58. **Завершить цикл.**
59. **Возвратить** $\wp_{best}$ ▷ лучшее решение функции приспособленности.

В обновлённой версии алгоритма MQIGA-SHA реализован механизм динамической настройки ключевых параметров – угла поворота θ_{shift}, управляющего интенсивностью скрещивания, и вероятности мутации P_{mut}. Адаптация основана на анализе истории успеха SHA: в каждом поколении алгоритм отслеживает, привело ли применение текущих значений параметров к улучшению значения функции приспособленности по сравнению с предыдущим поколением. Успешные значения параметров θ_{shift} и P_{mut} сохраняются и используются для обновления соответствующих историй фиксированного размера. Обновление выполняется с помощью взвешенного среднего Лемера второго порядка [20], которое учитывает не только факт успеха, но и степень улучшения. Полученное среднее комбинируется с предыдущим значением в истории с заданным весовым коэффициентом, после чего параметр ограничивается нижним пороговым значением для предотвращения преждевременного вырождения.

Для формального описания механизма динамической настройки значений параметров необходимо ввести следующие обозначения. Пусть в t-м поколении алгоритма для каждого i-го агента известны значения параметров $\theta_{shift}^{(j)}$ для j-го кудита и P_{mut}, а также значение функции приспособленности до и после применения генетических операторов: f_i^{old} и f_i^{new}. Тогда для задачи минимизации маска успеха определяется как:

$$s_i = \begin{cases} 1, & \text{если } f_i^{new} < f_i^{old} \\ 0, & \text{иначе} \end{cases}, \tag{20}$$

где $s_i = 1$ означает, что для i-го агента применение текущих значений параметров привело к улучшению решения, и данные значения параметров подлежат учёту при адаптации.

Для параметра θ_{shift} формируется вектор успешных значений:

$$\mathbf{\Theta}_{succ} = \left\{\theta_{shift}^{(j)} : s_i = 1, i = \overline{1,N}, j = \overline{1,L}\right\}, \tag{21}$$

где $\theta_{shift}^{(j)}$ – значение угла поворота, использованное для j-го кудита i-го агента.

Аналогично, для вероятности мутации P_{mut} формируется вектор успешных значений:

$$\mathbf{P}_{succ} = \left\{P_{mut} : s_i = 1, i = \overline{1,N}\right\}, \tag{22}$$

где P_{mut} – значение вероятности мутации, использованное для всего i-го агента.

Для каждого вектора из (21) и (22) вычисляется взвешенное среднее Лемера второго порядка:

$$\overline{\theta}_{Lem} = \frac{\sum_{\text{n}} w_{\text{n}} \theta_{\text{n}}^2}{\sum_{\text{n}} w_{\text{n}} \theta_{\text{n}}}, \quad \overline{P}_{Lem} = \frac{\sum_{\text{n}} w_{\text{n}} p_{\text{n}}^2}{\sum_{\text{n}} w_{\text{n}} p_{\text{n}}}, \tag{23}$$

где θ_{n} и p_{n} – n-е ($\text{n} = \overline{1,R}$, R – общее количество успешных случаев в текущем поколении, то есть число агентов, для которых $s_i = 1$) успешные значения из множеств (21) и (22) соответственно. Весовые коэффициенты w_{n} пропорциональны величине улучшения качества решения:

$$w_{\text{n}} = \frac{\Delta f_{\text{n}}}{\sum_{\text{n}} \Delta f_{\text{n}}}, \tag{24}$$

где $\Delta f_{\text{n}} = f_i^{old} - f_i^{new}$.

Таким образом, значительные улучшения Δf_{n} вносят больший вклад в вычисление нового среднего, что делает адаптацию чувствительной к степени успеха.

Полученные средние значения используются для обновления последнего элемента в соответствующих историях:

$$H_\theta^{new}[H_{size}] = (1 - K) \cdot H_\theta^{old}[H_{size}] + K \cdot \overline{\theta}_{Lem}, \tag{25}$$

$$H_p^{new}[H_{size}] = (1 - K) \cdot H_p^{old}[H_{size}] + K \cdot \overline{P}_{Lem}, \tag{26}$$

где $H_{size} \in \mathbb{N}$ – размер истории; форма записи вида $H_\theta[H_{size}]$ и $H_p[H_{size}]$ означает последний элемент в

массивах соответствующих историй H_θ и H_p; $H_\theta^{old}[H_{size}]$ и $H_p^{old}[H_{size}]$ – значения последних элементов в массивах соответствующих историй H_θ и H_p до обновления; $K \in [0,1]$ – коэффициент адаптации, управляющий скоростью обновления значений параметров (в случае $K = 1$ новое значение полностью заменяет старое, а при $K \to 0$ обновление осуществляется медленно).

После обновления истории сдвигаются циклически, а новые значения параметров для следующего поколения устанавливаются как среднее по всей истории:

$$\theta_{shift}^{(j)} \leftarrow \frac{1}{H_{size}} \sum_{r=1}^{H_{size}} H_\theta[r], \tag{27}$$

$$P_{mut} \leftarrow \frac{1}{H_{size}} \sum_{r=1}^{H_{size}} H_p[r]. \tag{28}$$

С целью предотвращения вырождения значений параметров, например, когда вероятность мутации стремится к нулю и алгоритм теряет способность к исследованию, значения параметров ограничиваются нижним пороговым значением:

$$\theta_{shift}^{(j)} \leftarrow \max\left(\theta_{shift}^{(j)}, \text{ɣ}\right), \quad P_{mut} \leftarrow \max(P_{mut}, \text{ɣ}). \tag{29}$$

где max – функция поиска максимального значения; $\text{ɣ} \in (0,1]$ – нижнее пороговое значение.

Полный псевдокод алгоритма MQIGA-SHA приведён в алгоритме 2.

Алгоритм 2. MQIGA-SHA.

Вход: $N \in \mathbb{N}$ ▷ размер популяции.
$L \in \mathbb{N}$ ▷ длина квантовой хромосомы (количество кудитов).
$d \in \mathbb{N}$ ▷ размерность кудита.
$T_{max} \in \mathbb{N}$ ▷ максимальное количество поколений.
$\varepsilon \in [0,1]$ ▷ уровень ошибки (шума) для модели декогеренции.
$P_{mut} \in [0,1]$ ▷ вероятность мутации агента.
$f: \mathbb{R}^v \to \mathbb{R}$ ▷ целевая функция приспособленности, где $v \in \mathbb{N}$ – размерность пространства поиска.
$[x_{min}, x_{max}] \subset \mathbb{R}$, $x_{min} < x_{max}$ ▷ диапазон значений для поиска оптимального решения целевой функции.
$\wp_{fit} = +\infty$ ▷ переменная, хранящая лучшее значение функции приспособленности.
$\wp_{best} = \emptyset$ ▷ переменная, хранящая лучшее решение функции приспособленности.
$H_{size} \in \mathbb{N}$ ▷ размер истории успеха.
$H_f = \emptyset$ ▷ массив для хранения значений функции приспособленности.
$H_\theta = \emptyset$ ▷ массив для хранения значений угла поворота θ_{shift}.
$H_p = \emptyset$ ▷ массив для хранения значений вероятности мутации P_{mut}.
$H_w = \emptyset$ ▷ массив для хранения значений весовых коэффициентов.
$K \in [0,1]$ ▷ коэффициент адаптации.
$\text{ɣ} \in (0,1]$ ▷ минимальное пороговое значение для настраиваемых параметров.

1. **Для каждого** агента $i = \overline{1,N}$ **выполнять:**
2. **Для каждого** кудита $j = \overline{1,L}$ **выполнять:**
3. $|0\rangle = (1,0,0,\dots,0)^T \in \mathbb{C}^d$ ▷ определить начальное квантовое состояние.
4. $|\psi_i^{(j)}\rangle_{init} = H_d|0\rangle$ ▷ инициализировать состояние суперпозиции.
5. $\boldsymbol{\rho}_i^{(j)} = |\psi_i^{(j)}\rangle_{init}\langle\psi_i^{(j)}|_{init}$ ▷ создать матрицу плотности.
6. **Завершить цикл.**
7. **Завершить цикл.**
8. **Установить** номер поколения $t = 0$.
9. **Пока** $t < T_{max}$ **выполнять:**
10. **Для каждого** агента $i = \overline{1,N}$ **выполнять:**
11. **Для каждого** кудита $j = \overline{1,L}$ **выполнять:**
12. $\boldsymbol{\rho}_i^{(j)}{}_{noisy} = (1-\varepsilon)\boldsymbol{\rho}_i^{(j)} + \varepsilon \mathbf{W}_{m,n} \boldsymbol{\rho}_i^{(j)} \mathbf{W}_{m,n}^\dagger$ ▷ изменить матрицу плотности воздействием оператора
13. из группы Гейзенберга-Вейля $\{\mathbf{W}_{m,n}\}_{m,n=0}^{d-1}$.
14. $p_k = \left\langle k \middle| \boldsymbol{\rho}_i^{(j)}{}_{noisy} \middle| k \right\rangle = \left(\boldsymbol{\rho}_i^{(j)}{}_{noisy}\right)_{kk}$ ▷ вычислить вероятность исхода измерения,
15. где $k = \overline{0,d-1}$.
16. **Выбрать** индекс $c_j = k$ с вероятностью p_k.
17. **Завершить цикл.**

18. $\mathbf{c} = (c_1, c_2, c_3, \dots, c_L)$ ▷ последовательность индексов кудитов i-го агента.
19. $x_{dec} = \sum_{j=1}^{L} c_j d^{L-j}$ ▷ вычислить десятичное значение на основе $c_j \in \mathbf{c}$.
20. $x_{norm} = x_{min} + \frac{x_{dec}}{d^L - 1}(x_{max} - x_{min})$ ▷ выполнить линейную нормализацию.
21. $f_i = f(x_{norm})$ ▷ оценить функцию приспособленности.
22. **Если** $f_i < \wp_{fit}$ **выполнить:**
23. $\wp_{fit} \leftarrow f_i$ ▷ обновить значение переменной.
24. $\wp_{best} \leftarrow \mathbf{c}$ ▷ обновить значение переменной.
25. $H_f \leftarrow H_f \cup \{f_i\}$ ▷ сохранить значение функции приспособленности.
26. **Завершить цикл.**
27. **Для каждого** агента $i = \overline{1, N}$ **выполнять:**
28. **Если** $i \in \wp_{best}$, то **пропустить** поколение.
29. **Для каждого** кудита $j = \overline{1, L}$ **выполнять:**
30. **Извлечь** квантовое состояние $|\psi_i^{(j)}\rangle$ из $\boldsymbol{\rho}_i^{(j)}$.
31. $|\psi_i^{(j)\prime}\rangle = \cos\left(\frac{\theta_{shift}^{(j)}}{2}\right)|\psi_{best}^{(j)}\rangle + \sin\left(\frac{\theta_{shift}^{(j)}}{2}\right)|\psi_i^{(j)}\rangle$ ▷ выполнить когерентное смешивание, где
32. $\theta_{shift}^{(j)} \in [0, 2\pi]$, $|\psi_{best}^{(j)}\rangle$ – квантовое состояние глобально лучшего агента.
33. $|\psi_i^{(j)\prime}\rangle \leftarrow \frac{|\psi_i^{(j)\prime}\rangle}{\||\psi_i^{(j)\prime}\rangle\|}$ ▷ выполнить нормализацию квантового состояния.
34. $\boldsymbol{\rho}_i^{(j)\prime} = |\psi_i^{(j)\prime}\rangle\langle\psi_i^{(j)\prime}|$ ▷ преобразовать квантовое состояние в матрицу плотности.
35. $\boldsymbol{\rho}_{i\ cross}^{(j)} = (1-\varepsilon)\boldsymbol{\rho}_i^{(j)\prime} + \varepsilon \mathbf{W}_{m,n}\boldsymbol{\rho}_i^{(j)\prime}\mathbf{W}_{m,n}^{\dagger}$ ▷ применить модель декогеренции.
36. **Завершить цикл.**
37. **Завершить цикл.**
38. **Для каждого** агента $i = \overline{1, N}$ **выполнять:**
39. **Если** $i \in \wp_{best}$, то **пропустить** поколение.
40. **Для каждого** агента **генерировать** случайное значение $u_i \sim \mathcal{U}(0, 1)$.
41. **Если** $u_i \leq P_{mut}$ **выполнять:**
42. **Для каждого** кудита $j = \overline{1, L}$ **выполнять:**
43. **Сгенерировать** случайное состояние $|\phi^{(j)}\rangle$ на единичной сфере в $\mathbb{C}^d$.
44. $|\psi_i^{(j)\prime\prime}\rangle = \cos\left(\frac{\theta_{rot}^{(j)}}{2}\right)|\psi_i^{(j)\prime}\rangle + \sin\left(\frac{\theta_{rot}^{(j)}}{2}\right)|\phi^{(j)}\rangle$ ▷ вычислить состояние после мутации, где
45. $\theta_{rot}^{(j)} \in [0, 2\pi]$.
46. $|\psi_i^{(j)\prime\prime}\rangle \leftarrow \frac{|\psi_i^{(j)\prime\prime}\rangle}{\||\psi_i^{(j)\prime\prime}\rangle\|}$ ▷ выполнить нормализацию квантового состояния.
47. $\boldsymbol{\rho}_i^{(j)\prime\prime} = |\psi_i^{(j)\prime\prime}\rangle\langle\psi_i^{(j)\prime\prime}|$ ▷ преобразовать квантовое состояние в матрицу плотности.
48. $\boldsymbol{\rho}_{i\ mut}^{(j)} = (1-\varepsilon)\boldsymbol{\rho}_i^{(j)\prime\prime} + \varepsilon \mathbf{W}_{m,n}\boldsymbol{\rho}_i^{(j)\prime\prime}\mathbf{W}_{m,n}^{\dagger}$ ▷ применить модель декогеренции.
49. $\boldsymbol{\rho}_i^{(j)} \leftarrow \boldsymbol{\rho}_{i\ mut}^{(j)}$ ▷ обновить матрицу плотности кудита.
50. **Завершить цикл.**
51. **Иначе выполнить:**
52. **Для каждого** кудита $j = \overline{1, L}$ **выполнять:**
53. $\boldsymbol{\rho}_i^{(j)} \leftarrow \boldsymbol{\rho}_{i\ cross}^{(j)}$ ▷ обновить матрицу плотности кудита.
54. **Завершить цикл.**
55. **Завершить цикл.**
56. **Если** $\wp_{best} \neq \emptyset$ **выполнять:**
57. **Записать** $\wp_{best}$ в популяцию.
58. **Если** $t > 0$ **выполнять:**
59. **Для каждого** агента $i = \overline{1, N}$ **выполнять:**
60. $s_i = \begin{cases} 1, & \text{если } f_i^{new} < f_i^{old} \\ 0, & \text{иначе} \end{cases}$ ▷ определить маску успеха при $f_i^{old}, f_i^{new} \in H_f$.
61. **Если** $s_i = 1$ **выполнять:**
62. **Для каждого** кудита $j = \overline{1, L}$ **выполнять:**
63. $H_\theta \leftarrow H_\theta \cup \left\{\theta_{shift}^{(j)}\right\}$ ▷ сохранить успешное значение угла поворота θ_{shift} j-го кудита.
64. **Завершить цикл.**

$H_p \leftarrow H_p \cup \{P_{mut}\}$ ▷ сохранить успешное значение вероятности мутации P_{mut} i-го агента.
$\Delta f_{\text{n}} = f_i^{old} - f_i^{new}$ ▷ вычислить значение степени улучшения агента.
$w_{\text{n}} = \frac{\Delta f_{\text{n}}}{\sum_{\text{n}} \Delta f_{\text{n}}}$ ▷ вычислить значение весового коэффициента.
$H_w \leftarrow H_w \cup \{w_{\text{n}}\}$ ▷ сохранить значение весового коэффициента.
Завершить цикл.
$\boldsymbol{\Theta}_{succ} = \left\{\theta_{shift}^{(j)} : s_i = 1, i = \overline{1,N}, j = \overline{1,L}\right\}$ ▷ сформировать вектор успешных значений θ_{shift}.
$\mathbf{P}_{succ} = \left\{P_{mut} : s_i = 1, i = \overline{1,N}\right\}$ ▷ сформировать вектор успешных значений P_{mut}.
$\overline{\theta}_{Lem} = \frac{\sum_{\text{n}} w_{\text{n}} \theta_{\text{n}}^2}{\sum_{\text{n}} w_{\text{n}} \theta_{\text{n}}}$ ▷ вычислить взвешенное среднее Лемера при каждом $w_{\text{n}} \in H_w$, $\theta_{\text{n}} \in \boldsymbol{\Theta}_{succ}$,
где $\text{n} = \overline{1,R}$, R – общее количество успешных агентов поколения.
$\overline{P}_{Lem} = \frac{\sum_{\text{n}} w_{\text{n}} p_{\text{n}}^2}{\sum_{\text{n}} w_{\text{n}} p_{\text{n}}}$ ▷ вычислить взвешенное среднее Лемера при каждом $w_{\text{n}} \in H_w$, $p_{\text{n}} \in \mathbf{P}_{succ}$,
где $\text{n} = \overline{1,R}$, R – общее количество успешных агентов поколения.
$H_\theta^{new}[H_{size}] = (1-K) \cdot H_\theta^{old}[H_{size}] + K \cdot \overline{\theta}_{Lem}$ ▷ обновить значение последнего элемента в H_θ,
где $H_\theta^{old}[H_{size}]$ – значение последнего элемента в H_θ до обновления.
$H_\theta^{new}[H_{size}] \leftarrow \max(H_\theta^{new}[H_{size}], \text{x})$ ▷ применить пороговое ограничение.
$H_p^{new}[H_{size}] = (1-K) \cdot H_p^{old}[H_{size}] + K \cdot \overline{P}_{Lem}$ ▷ обновить значение последнего элемента в H_p,
где $H_p^{old}[H_{size}]$ – значение последнего элемента в H_p до обновления.
$H_p^{new}[H_{size}] \leftarrow \max\left(H_p^{new}[H_{size}], \text{x}\right)$ ▷ применить пороговое ограничение.
Сдвинуть истории H_θ и H_p циклически.
$\theta_{shift}^{(j)} \leftarrow \frac{1}{H_{size}} \sum_{r=1}^{H_{size}} H_\theta[r]$ ▷ обновить углы поворота θ_{shift} для $j = \overline{1,L}$ в соответствии с H_θ.
$P_{mut} \leftarrow \frac{1}{H_{size}} \sum_{r=1}^{H_{size}} H_p[r]$ ▷ обновить вероятность мутации P_{mut}.
$t \leftarrow t + 1$ ▷ увеличить счётчик поколений.
Завершить цикл.
Возвратить $\wp_{best}$ ▷ лучшее решение функции приспособленности.

В совокупности описанные действия формируют саморегулирующийся оптимизационный алгоритм, способную автономно подстраивать свою внутреннюю динамику под меняющийся ландшафт целевой функции. В отличие от подходов с фиксированными или эвристически задаваемыми значениями параметров, MQIGA-SHA не требует предварительного анализа задачи или ручной настройки, что значительно расширяет его применимость в условиях «чёрного ящика». Вместе с тем интеграция квантово-инспирированной модели с адаптацией на основе истории успеха даёт синергетический эффект: квантовое представление обеспечивает разнообразие пространства поиска, а механизм SHA направляет этот поиск, концентрируя вычислительные ресурсы на наиболее перспективных стратегиях. Эти особенности делают алгоритм MQIGA-SHA не просто более эффективным, а «интеллектуальным» инструментом, способным к обучению на собственном опыте и к принятию обоснованных решений о том, как именно следует искать оптимум в текущий момент времени.

Экспериментальная часть

Для оценки эффективности предложенного алгоритма MQIGA-SHA проведена серия вычислительных экспериментов на основе набора эталонных функций из бенчмарка по эволюционным вычислениям [21]. Данный бенчмарк включает функции, сгруппированные по типам сложности: унимодальные, мультимодальные, гибридные и композитные. Такая структура позволяет всесторонне протестировать способность алгоритма MQIGA-SHA к глобальному исследованию пространства решений, устойчивости к локальным оптимумам, адаптации к несепарабельным и многомасштабным ландшафтам, а также к функциям с различными типами смещения, вращения и композиции. Все функции определяются как задачи минимизации на гиперкубе $[-100, 100]^D$ (где D – размерность пространства) и обладают сдвинутым глобальным оптимумом, расположенным случайным образом внутри области поиска, что исключает возможность «настройки» алгоритма под известное положение минимума и обеспечивает соответствие условиям «чёрного ящика».

В тестировании использованы 4 мультимодальные функции, 3 гибридные функции и 3 композитные функции. Для всех тестовых задач оптимизации вводятся обозначения: **M** – матрица поворота, которая генерируется из стандартных нормально распределённых элементов на основе ортогонализации Грама-Шмидта с числом обусловленности, равным 1 или 2; **o** – смещённый глобальный оптимум, который случайным образом распределяется в диапазоне $[-80, 80]^D$.

Рассмотрены следующие функции:

1. T_1 – сдвинутая и повёрнутая расширенная функция Шаффера F6:

$$F_1(\mathbf{x}) = f_1\left(\mathbf{M}\left(\frac{0.5(\mathbf{x}-\mathbf{o}_1)}{100}\right)\right) + 600,$$
$$f_1(\mathbf{x}) = g(x_1, x_2) + g(x_2, x_3) + \cdots + g(x_{D-1}, x_D) + g(x_D, x_1), \tag{30}$$

где $g(x,y) = 0.5 + \frac{\left(\sin^2\left(\sqrt{x^2+y^2}\right)-0.5\right)}{\left(1+0.001(x^2+y^2)\right)^2}$.

2. T_2 – сдвинутая и повёрнутая функция Луначека №2:

$$F_2(\mathbf{x}) = f_2\left(\mathbf{M}\left(\frac{600(\mathbf{x}-\mathbf{o}_2)}{100}\right)\right) + 700,$$
$$f_2(x) = \min\left(\sum_{i=1}^{D}(\hat{x}_i - \mu_0)^2, dD + s\sum_{i=1}^{D}(\hat{x}_i - \mu_1)^2\right) + 10\left(D - \sum_{i=1}^{D}\cos(2\pi\hat{z}_i)\right), \tag{31}$$

где $\mu_0 = 2.5$; $\mu_1 = -\sqrt{\frac{\mu_0^2 - d}{s}}$; $s = 1 - \frac{1}{2\sqrt{D+20}-8.2}$; $d = 1$; $y = \frac{10(\mathbf{x}-\mathbf{o})}{100}$; $\hat{x}_i = 2\,\mathrm{sign}(x_i^*)\,y_i + \mu_0$ при $i = \overline{1,D}$; $z = \Lambda^{100}(\hat{\mathbf{x}} - \mu_0)$, Λ^α – диагональная матрица с диагональными элементами $\lambda_{ii} = \alpha^{\frac{i-1}{2(D-1)}}$ при $i = \overline{1,D}$.

3. T_3 – разрывная повёрнутая функция Растригина:

$$F_3(\mathbf{x}) = f_3\left(\frac{5.12(\mathbf{x}-\mathbf{o}_3)}{100}\right) + 800,$$
$$f_3(x) = \sum_{i=1}^{D}(z_i^2 - 10\cos(2\pi z_i) + 10), \tag{32}$$

где $z = \mathbf{M}_1\Lambda^{10}\mathbf{M}_2 T_{asy}^{0.2}\left(T_{osz}(y)\right)$, Λ^α – диагональная матрица с диагональными элементами $\lambda_{ii} = \alpha^{\frac{i-1}{2(D-1)}}$ при $i = \overline{1,D}$; $y_i = \begin{cases}\hat{x}_i, & \text{если } |\hat{x}_i| \le 0.5\\ \mathrm{round}(2\hat{x}_i)/2, & \text{если } |\hat{x}_i| > 0.5\end{cases}$; $\hat{x} = \mathbf{M}_1\frac{5.12(\mathbf{x}-\mathbf{o})}{100}$, T_{asy}^β определяется как $x_i = x_i^{1+\beta\frac{i-1}{D-1}\sqrt{x_i}}$ при $x_i > 0$ и $i = \overline{1,D}$, T_{osz} определяется как $x_i = \mathrm{sign}(x_i)\exp\left(\hat{x}_i + 0.049(\sin(c_1\hat{x}_i) + \sin(c_2\hat{x}_i))\right)$ при $i = \overline{1,D}$, где $\hat{x}_i = \begin{cases}\log(|x_i|), & \text{если } x_i \ne 0\\ 0, & \text{иначе}\end{cases}$, $\mathrm{sign}(x_i) = \begin{cases}-1, & \text{если } x_i < 0\\ 0, & \text{если } x_i = 0\\ 1, & \text{иначе}\end{cases}$, $c_1 = \begin{cases}10, & \text{если } x_i > 0\\ 5.5, & \text{иначе}\end{cases}$, $c_2 = \begin{cases}7.9, & \text{если } x_i > 0\\ 3.1, & \text{иначе}\end{cases}$.

4. T_4 – сдвинутая и повёрнутая модифицированная функция Швефеля:

$$F_4(\mathbf{x}) = f_4\left(\frac{1000(\mathbf{x}-\mathbf{o}_4)}{100}\right) + 1000,$$
$$f_4(\mathbf{x}) = 418.9829 \cdot D - \sum_{i=1}^{D} g(z_i), \tag{33}$$

где $z_i = x_i + 4.209687462275036\mathrm{e}+002$;

$$g(z_i) = \begin{cases} z_i\sin\left(|z_i|^{1/2}\right), & \text{если } |z_i| \le 500\\ (500 - \mathrm{mod}(z_i, 500))\sin\left(\sqrt{|500 - \mathrm{mod}(z_i, 500)|}\right) - \frac{(z_i-500)^2}{10000D}, & \text{если } z_i > 500\\ (\mathrm{mod}(|z_i|, 500) - 500)\sin\left(\sqrt{|\mathrm{mod}(|z_i|, 500) - 500|}\right) - \frac{(z_i+500)^2}{10000D}, & \text{если } z_i < -500\end{cases}.$$

5. T_5 – гибридная функция №1:

$$F_5(\mathbf{x}) = g_1(\mathbf{M}_1\mathbf{z}_1) + g_2(\mathbf{M}_2\mathbf{z}_2) + g_3(\mathbf{M}_3\mathbf{z}_3) + g_4(\mathbf{M}_4\mathbf{z}_4) + 1400, \tag{34}$$

где:

$\mathbf{z}_1 = \left[\mathbf{y}_{S_1}, \mathbf{y}_{S_2}, \ldots, \mathbf{y}_{S_{n_1}}\right]$,

$\mathbf{z}_2 = \left[\mathbf{y}_{S_{n_1+1}}, \mathbf{y}_{S_{n_1+2}}, \ldots, \mathbf{y}_{S_{n_1+n_2}}\right]$,

$\mathbf{z}_3 = \left[\mathbf{y}_{S_{n_1+n_2+2}}, \mathbf{y}_{S_{n_1+n_2+4}}, \ldots, \mathbf{y}_{S_{n_1+n_2+n_3}}\right]$,

$\mathbf{z}_4 = \left[\mathbf{y}_{S_{n_1+n_2+n_3+3}}, \mathbf{y}_{S_{n_1+n_2+n_3+6}}, \ldots, \mathbf{y}_{S_{n_1+n_2+n_3+n_4}}\right]$;

$S \sim \mathcal{U}(1, \Xi_D)$, Ξ_D – множество всех перестановок множества $\{1, 2, \ldots, D\}$; $\mathbf{y} = \mathbf{x} - \mathbf{o}_i$; $n_1 = \lceil 0.2D \rceil$, $n_2 = \lceil 0.2D \rceil$, $n_3 = \lceil 0.2D \rceil$, $n_4 = \lceil 0.4D \rceil$; при следующих базовых функциях:

$$g_1(\mathbf{x}) = \sum_{j=1}^{D}(10^6)^{\frac{j-1}{D-1}}x_j^2, \tag{35}$$

$$g_2(\mathbf{x}) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{j=1}^{D}x_j^2}\right) - \exp\left(\frac{1}{D}\sum_{j=1}^{D}\cos\left(2\pi x_j\right)\right) + 20 + \mathrm{e}, \tag{36}$$

$$g_3(\mathbf{x}) = \left[\frac{1}{D-1}\sum_{j=1}^{D-1}\left(\sqrt{\sqrt{x_j^2 + x_{j+1}^2}}\cdot\left(\sin\left(50\sqrt{x_j^2 + x_{j+1}^2}^{0.2}\right)+1\right)\right)\right]^2, \quad (37)$$

$$g_4(\mathbf{x}) = \sum_{j=1}^{D}\left(x_j^2 - 10\cos\left(2\pi x_j\right) + 10\right). \quad (38)$$

6. T_6 – гибридная функция №2:

$$F_6(\mathbf{x}) = g_1(\mathbf{M}_1\mathbf{z}_1) + g_2(\mathbf{M}_2\mathbf{z}_2) + g_3(\mathbf{M}_3\mathbf{z}_3) + g_4(\mathbf{M}_4\mathbf{z}_4) + g_5(\mathbf{M}_5\mathbf{z}_5) + 1700, \quad (39)$$

где:

$\mathbf{z}_1 = \left[\mathbf{y}_{S_1}, \mathbf{y}_{S_2}, \dots, \mathbf{y}_{S_{n_1}}\right]$,

$\mathbf{z}_2 = \left[\mathbf{y}_{S_{n_1+1}}, \mathbf{y}_{S_{n_1+2}}, \dots, \mathbf{y}_{S_{n_1+n_2}}\right]$,

$\mathbf{z}_3 = \left[\mathbf{y}_{S_{n_1+n_2+2}}, \mathbf{y}_{S_{n_1+n_2+4}}, \dots, \mathbf{y}_{S_{n_1+n_2+n_3}}\right]$,

$\mathbf{z}_4 = \left[\mathbf{y}_{S_{n_1+n_2+n_3+3}}, \mathbf{y}_{S_{n_1+n_2+n_3+6}}, \dots, \mathbf{y}_{S_{n_1+n_2+n_3+n_4}}\right]$,

$\mathbf{z}_5 = \left[\mathbf{y}_{S_{n_1+n_2+n_3+n_4+4}}, \mathbf{y}_{S_{n_1+n_2+n_3+n_4+8}}, \dots, \mathbf{y}_{S_{n_1+n_2+n_3+n_4+n_5}}\right]$;

$S\sim\mathcal{U}(1, \Xi_D)$, Ξ_D – множество всех перестановок множества $\{1, 2, \dots, D\}$; $\mathbf{y} = \mathbf{x} - \mathbf{o}_i$; $n_1 = \lceil 0.1D\rceil$, $n_2 = \lceil 0.2D\rceil$, $n_3 = \lceil 0.2D\rceil$, $n_4 = \lceil 0.2D\rceil$, $n_5 = \lceil 0.3D\rceil$; при следующих базовых функциях:

$$g_1(\mathbf{x}) = \frac{10}{D^2}\prod_{j=1}^{D}\left(1 + j\sum_{k=1}^{32}\frac{\left|2^k x_j - \mathrm{round}\left(2^k x_j\right)\right|}{2^k}\right)^{\frac{10}{D^{1.2}}} - \frac{10}{D^2}, \quad (40)$$

$$g_2(\mathbf{x}) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{j=1}^{D}x_j^2}\right) - \exp\left(\frac{1}{D}\sum_{j=1}^{D}\cos\left(2\pi x_j\right)\right) + 20 + \mathrm{e}, \quad (41)$$

$$\begin{gathered} g_3(\mathbf{x}) = \varpi_1\left(\varpi_2(x_1, x_2)\right) + \varpi_1\left(\varpi_2(x_2, x_3)\right) + \dots + \varpi_1\left(\varpi_2(x_{D-1}, x_D)\right) + \varpi_1\left(\varpi_2(x_D, x_1)\right), \\ \varpi_1(x) \Leftrightarrow f_2(x) \quad (30), \\ \varpi_2(\mathbf{x}) = \sum_{j=1}^{D-1}\left(100\left(x_j^2 - x_{j+1}\right)^2 + \left(x_j - 1\right)^2\right), \end{gathered} \quad (42)$$

$$g_4(\mathbf{x}) = 418.9829\cdot D - \sum_{j=1}^{D}\xi\left(\chi_j\right),$$

$$\chi_j = x_j + 4.209687462275036\mathrm{e}+002,$$

$$\xi(\chi_j) = \begin{cases} \chi_j\sin\left(\left|\chi_j\right|^{1/2}\right), & \text{если } \left|\chi_j\right| \le 500 \\ \left(500 - \mathrm{mod}\left(\chi_j, 500\right)\right)\sin\left(\sqrt{\left|500 - \mathrm{mod}\left(\chi_j, 500\right)\right|}\right) - \frac{\left(\chi_j - 500\right)^2}{10000D}, & \text{если } \chi_j > 500 \\ \left(\mathrm{mod}\left(\left|\chi_j\right|, 500\right) - 500\right)\sin\left(\sqrt{\left|\mathrm{mod}\left(\left|\chi_j\right|, 500\right) - 500\right|}\right) - \frac{\left(\chi_j + 500\right)^2}{10000D}, & \text{если } \chi_j < -500 \end{cases}, \quad (43)$$

$$g_5(\mathbf{x}) = \sum_{j=1}^{D}\left(x_j^2 - 10\cos\left(2\pi x_j\right) + 10\right). \quad (44)$$

7. T_7 – гибридная функция №3:

$$F_7(\mathbf{x}) = g_1(\mathbf{M}_1\mathbf{z}_1) + g_2(\mathbf{M}_2\mathbf{z}_2) + g_3(\mathbf{M}_3\mathbf{z}_3) + g_4(\mathbf{M}_4\mathbf{z}_4) + g_5(\mathbf{M}_5\mathbf{z}_5) + g_6(\mathbf{M}_6\mathbf{z}_6) + 2000, \quad (45)$$

где:

$\mathbf{z}_1 = \left[\mathbf{y}_{S_1}, \mathbf{y}_{S_2}, \dots, \mathbf{y}_{S_{n_1}}\right]$,

$\mathbf{z}_2 = \left[\mathbf{y}_{S_{n_1+1}}, \mathbf{y}_{S_{n_1+2}}, \dots, \mathbf{y}_{S_{n_1+n_2}}\right]$,

$\mathbf{z}_3 = \left[\mathbf{y}_{S_{n_1+n_2+2}}, \mathbf{y}_{S_{n_1+n_2+4}}, \dots, \mathbf{y}_{S_{n_1+n_2+n_3}}\right]$,

$\mathbf{z}_4 = \left[\mathbf{y}_{S_{n_1+n_2+n_3+3}}, \mathbf{y}_{S_{n_1+n_2+n_3+6}}, \dots, \mathbf{y}_{S_{n_1+n_2+n_3+n_4}}\right]$,

$\mathbf{z}_5 = \left[\mathbf{y}_{S_{n_1+n_2+n_3+n_4+4}}, \mathbf{y}_{S_{n_1+n_2+n_3+n_4+8}}, \dots, \mathbf{y}_{S_{n_1+n_2+n_3+n_4+n_5}}\right]$,

$\mathbf{z}_6 = \left[\mathbf{y}_{S_{n_1+n_2+n_3+n_4+n_5+5}}, \mathbf{y}_{S_{n_1+n_2+n_3+n_4+n_5+10}}, \dots, \mathbf{y}_{S_{n_1+n_2+n_3+n_4+n_5+n_6}}\right]$;

$S\sim\mathcal{U}(1, \Xi_D)$, Ξ_D – множество всех перестановок множества $\{1, 2, \dots, D\}$; $\mathbf{y} = \mathbf{x} - \mathbf{o}_i$; $n_1 = \lceil 0.1D\rceil$, $n_2 = \lceil 0.1D\rceil$, $n_3 = \lceil 0.2D\rceil$, $n_4 = \lceil 0.2D\rceil$, $n_5 = \lceil 0.2D\rceil$; $n_6 = \lceil 0.2D\rceil$; при следующих базовых функциях:

$$g_1(\mathbf{x}) = \left|\sum_{j=1}^{D}x_j^2 - D\right|^{1/4} + \frac{0.5\sum_{j=1}^{D}x_j^2 + \sum_{j=1}^{D}x_j}{D} + 0.5, \quad (46)$$

$$g_2(\mathbf{x}) = \frac{10}{D^2}\prod_{j=1}^{D}\left(1 + j\sum_{k=1}^{32}\frac{\left|2^k x_j - \mathrm{round}\left(2^k x_j\right)\right|}{2^k}\right)^{\frac{10}{D^{1.2}}} - \frac{10}{D^2}, \tag{47}$$

$$g_3(\mathbf{x}) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{j=1}^{D}x_j^2}\right) - \exp\left(\frac{1}{D}\sum_{j=1}^{D}\cos\left(2\pi x_j\right)\right) + 20 + \mathrm{e}, \tag{48}$$

$$g_4(\mathbf{x}) = \sum_{j=1}^{D}\left(x_j^2 - 10\cos\left(2\pi x_j\right) + 10\right), \tag{49}$$

$$g_5(\mathbf{x}) = 418.9829 \cdot D - \sum_{j=1}^{D}\xi\left(\chi_j\right),$$

$$\chi_j = x_j + 4.209687462275036\mathrm{e} + 002,$$

$$\xi\left(\chi_j\right) = \begin{cases} \chi_j \sin\left(\left|\chi_j\right|^{1/2}\right), & \text{если } \left|\chi_j\right| \le 500 \\ \left(500 - \mathrm{mod}\left(\chi_j, 500\right)\right)\sin\left(\sqrt{\left|500 - \mathrm{mod}\left(\chi_j, 500\right)\right|}\right) - \frac{\left(\chi_j - 500\right)^2}{10000D}, & \text{если } \chi_j > 500 \\ \left(\mathrm{mod}\left(\left|\chi_j\right|, 500\right) - 500\right)\sin\left(\sqrt{\left|\mathrm{mod}\left(\left|\chi_j\right|, 500\right) - 500\right|}\right) - \frac{\left(\chi_j + 500\right)^2}{10000D}, & \text{если } \chi_j < -500 \end{cases}, \tag{50}$$

$$g_6(\mathbf{x}) = \left[\frac{1}{D-1}\sum_{j=1}^{D-1}\left(\sqrt{\sqrt{x_j^2 + x_{j+1}^2}}\cdot\left(\sin\left(50\sqrt{x_j^2 + x_{j+1}^2}^{0.2}\right) + 1\right)\right)\right]^2. \tag{51}$$

8. T_8 – композитная функция №1:

$$F_8(\mathbf{x}) = \frac{\frac{1}{\sqrt{\sum_{j=1}^{D}\left(x_j - o_{1j}\right)^2}}\exp\left(-\frac{\sum_{j=1}^{D}\left(x_j - o_{1j}\right)^2}{200D}\right)}{\frac{1}{\sqrt{\sum_{j=1}^{D}\left(x_j - o_{1j}\right)^2}}\exp\left(-\frac{\sum_{j=1}^{D}\left(x_j - o_{1j}\right)^2}{200D}\right) + \frac{1}{\sqrt{\sum_{j=1}^{D}\left(x_j - o_{2j}\right)^2}}\exp\left(-\frac{\sum_{j=1}^{D}\left(x_j - o_{2j}\right)^2}{800D}\right) + \frac{1}{\sqrt{\sum_{j=1}^{D}\left(x_j - o_{3j}\right)^2}}\exp\left(-\frac{\sum_{j=1}^{D}\left(x_j - o_{3j}\right)^2}{1800D}\right)} \cdot$$
$$\cdot\left(\sum_{j=1}^{D}\left(x_j^2 - 10\cos\left(2\pi x_j\right) + 10\right) - 500\right) +$$
$$+\frac{\frac{1}{\sqrt{\sum_{j=1}^{D}\left(x_j - o_{2j}\right)^2}}\exp\left(-\frac{\sum_{j=1}^{D}\left(x_j - o_{2j}\right)^2}{800D}\right)}{\frac{1}{\sqrt{\sum_{j=1}^{D}\left(x_j - o_{1j}\right)^2}}\exp\left(-\frac{\sum_{j=1}^{D}\left(x_j - o_{1j}\right)^2}{200D}\right) + \frac{1}{\sqrt{\sum_{j=1}^{D}\left(x_j - o_{2j}\right)^2}}\exp\left(-\frac{\sum_{j=1}^{D}\left(x_j - o_{2j}\right)^2}{800D}\right) + \frac{1}{\sqrt{\sum_{j=1}^{D}\left(x_j - o_{3j}\right)^2}}\exp\left(-\frac{\sum_{j=1}^{D}\left(x_j - o_{3j}\right)^2}{1800D}\right)} \cdot$$
$$\cdot\left(10 \cdot \left(\sum_{j=1}^{D}\frac{x_j^2}{4000} - \prod_{j=1}^{D}\cos\left(\frac{x_j}{\sqrt{j}}\right) - 1499\right) + 100\right) +$$
$$+\frac{\frac{1}{\sqrt{\sum_{j=1}^{D}\left(x_j - o_{3j}\right)^2}}\exp\left(-\frac{\sum_{j=1}^{D}\left(x_j - o_{3j}\right)^2}{1800D}\right)}{\frac{1}{\sqrt{\sum_{j=1}^{D}\left(x_j - o_{1j}\right)^2}}\exp\left(-\frac{\sum_{j=1}^{D}\left(x_j - o_{1j}\right)^2}{200D}\right) + \frac{1}{\sqrt{\sum_{j=1}^{D}\left(x_j - o_{2j}\right)^2}}\exp\left(-\frac{\sum_{j=1}^{D}\left(x_j - o_{2j}\right)^2}{800D}\right) + \frac{1}{\sqrt{\sum_{j=1}^{D}\left(x_j - o_{3j}\right)^2}}\exp\left(-\frac{\sum_{j=1}^{D}\left(x_j - o_{3j}\right)^2}{1800D}\right)} \cdot$$
$$\cdot\left(418.9829 \cdot D - \sum_{j=1}^{D}\xi\left(\chi_j\right) - 800\right) + 2200, \tag{52}$$

где o_1, o_2, o_3 – новые смещённые положения оптимумов, определяющие положения глобального и локальных оптимумов; $\chi_j = x_j + 4.209687462275036\mathrm{e} + 002$,

$$\xi\left(\chi_j\right) = \begin{cases} \chi_j \sin\left(\left|\chi_j\right|^{1/2}\right), & \text{если } \left|\chi_j\right| \le 500 \\ \left(500 - \mathrm{mod}\left(\chi_j, 500\right)\right)\sin\left(\sqrt{\left|500 - \mathrm{mod}\left(\chi_j, 500\right)\right|}\right) - \frac{\left(\chi_j - 500\right)^2}{10000D}, & \text{если } \chi_j > 500 \\ \left(\mathrm{mod}\left(\left|\chi_j\right|, 500\right) - 500\right)\sin\left(\sqrt{\left|\mathrm{mod}\left(\left|\chi_j\right|, 500\right) - 500\right|}\right) - \frac{\left(\chi_j + 500\right)^2}{10000D}, & \text{если } \chi_j < -500 \end{cases}.$$

9. T_9 – композитная функция №2:

$$F_9(\mathbf{x}) = \frac{\frac{1}{\sqrt{\sum_{j=1}^{D}(x_j-o_{1j})^2}}\exp\left(-\frac{\sum_{j=1}^{D}(x_j-o_{1j})^2}{200D}\right)}{\frac{1}{\sqrt{\sum_{j=1}^{D}(x_j-o_{1j})^2}}\exp\left(-\frac{\sum_{j=1}^{D}(x_j-o_{1j})^2}{200D}\right)+\frac{1}{\sqrt{\sum_{j=1}^{D}(x_j-o_{2j})^2}}\exp\left(-\frac{\sum_{j=1}^{D}(x_j-o_{2j})^2}{800D}\right)+\frac{1}{\sqrt{\sum_{j=1}^{D}(x_j-o_{3j})^2}}\exp\left(-\frac{\sum_{j=1}^{D}(x_j-o_{3j})^2}{1800D}\right)+\frac{1}{\sqrt{\sum_{j=1}^{D}(x_j-o_{4j})^2}}\exp\left(-\frac{\sum_{j=1}^{D}(x_j-o_{4j})^2}{3200D}\right)}\cdot$$
$$\cdot\left(\sum_{j=1}^{D-1}\left(100\left(x_j^2-x_{j+1}\right)^2+\left(x_j-1\right)^2\right)-400\right)+$$
$$+\frac{\frac{1}{\sqrt{\sum_{j=1}^{D}(x_j-o_{2j})^2}}\exp\left(-\frac{\sum_{j=1}^{D}(x_j-o_{2j})^2}{800D}\right)}{\frac{1}{\sqrt{\sum_{j=1}^{D}(x_j-o_{1j})^2}}\exp\left(-\frac{\sum_{j=1}^{D}(x_j-o_{1j})^2}{200D}\right)+\frac{1}{\sqrt{\sum_{j=1}^{D}(x_j-o_{2j})^2}}\exp\left(-\frac{\sum_{j=1}^{D}(x_j-o_{2j})^2}{800D}\right)+\frac{1}{\sqrt{\sum_{j=1}^{D}(x_j-o_{3j})^2}}\exp\left(-\frac{\sum_{j=1}^{D}(x_j-o_{3j})^2}{1800D}\right)+\frac{1}{\sqrt{\sum_{j=1}^{D}(x_j-o_{4j})^2}}\exp\left(-\frac{\sum_{j=1}^{D}(x_j-o_{4j})^2}{3200D}\right)}\cdot$$
$$\cdot\left(10\cdot\left(-20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{j=1}^{D}x_j^2}\right)-\exp\left(\frac{1}{D}\sum_{j=1}^{D}\cos(2\pi x_j)\right)-1280+\mathrm{e}\right)+100\right)+ \qquad (53)$$
$$+\frac{\frac{1}{\sqrt{\sum_{j=1}^{D}(x_j-o_{3j})^2}}\exp\left(-\frac{\sum_{j=1}^{D}(x_j-o_{3j})^2}{1800D}\right)}{\frac{1}{\sqrt{\sum_{j=1}^{D}(x_j-o_{1j})^2}}\exp\left(-\frac{\sum_{j=1}^{D}(x_j-o_{1j})^2}{200D}\right)+\frac{1}{\sqrt{\sum_{j=1}^{D}(x_j-o_{2j})^2}}\exp\left(-\frac{\sum_{j=1}^{D}(x_j-o_{2j})^2}{800D}\right)+\frac{1}{\sqrt{\sum_{j=1}^{D}(x_j-o_{3j})^2}}\exp\left(-\frac{\sum_{j=1}^{D}(x_j-o_{3j})^2}{1800D}\right)+\frac{1}{\sqrt{\sum_{j=1}^{D}(x_j-o_{4j})^2}}\exp\left(-\frac{\sum_{j=1}^{D}(x_j-o_{4j})^2}{3200D}\right)}\cdot$$
$$\cdot\left(418.9829\cdot D-\sum_{j=1}^{D}\xi(\chi_j)-800\right)+$$
$$+\frac{\frac{1}{\sqrt{\sum_{j=1}^{D}(x_j-o_{4j})^2}}\exp\left(-\frac{\sum_{j=1}^{D}(x_j-o_{4j})^2}{3200D}\right)}{\frac{1}{\sqrt{\sum_{j=1}^{D}(x_j-o_{1j})^2}}\exp\left(-\frac{\sum_{j=1}^{D}(x_j-o_{1j})^2}{200D}\right)+\frac{1}{\sqrt{\sum_{j=1}^{D}(x_j-o_{2j})^2}}\exp\left(-\frac{\sum_{j=1}^{D}(x_j-o_{2j})^2}{800D}\right)+\frac{1}{\sqrt{\sum_{j=1}^{D}(x_j-o_{3j})^2}}\exp\left(-\frac{\sum_{j=1}^{D}(x_j-o_{3j})^2}{1800D}\right)+\frac{1}{\sqrt{\sum_{j=1}^{D}(x_j-o_{4j})^2}}\exp\left(-\frac{\sum_{j=1}^{D}(x_j-o_{4j})^2}{3200D}\right)}\cdot$$
$$\cdot\left(\sum_{j=1}^{D}\left(x_j^2-10\cos(2\pi x_j)+10\right)-200\right)+2300,$$

где o_1, o_2, o_3, o_4 – новые смещённые положения оптимумов, определяющие положения глобального и локальных оптимумов; $\chi_j = x_j + 4.209687462275036\mathrm{e}+002$,

$$\xi(\chi_j)=\begin{cases}\chi_j\sin\left(\left|\chi_j\right|^{1/2}\right), & \text{если } \left|\chi_j\right|\le 500\\ \left(500-\mathrm{mod}(\chi_j,500)\right)\sin\left(\sqrt{\left|500-\mathrm{mod}(\chi_j,500)\right|}\right)-\frac{(\chi_j-500)^2}{10000D}, & \text{если } \chi_j>500\\ \left(\mathrm{mod}(\left|\chi_j\right|,500)-500\right)\sin\left(\sqrt{\left|\mathrm{mod}(\left|\chi_j\right|,500)-500\right|}\right)-\frac{(\chi_j+500)^2}{10000D}, & \text{если } \chi_j<-500\end{cases}.$$

10. T_{10} – композитная функция №3:

$$F_{10}(\mathbf{x}) = \frac{\frac{1}{\sqrt{\sum_{j=1}^{D}(x_j-o_{1j})^2}}\exp\left(-\frac{\sum_{j=1}^{D}(x_j-o_{1j})^2}{200D}\right)}{\frac{1}{\sqrt{\sum_{j=1}^{D}(x_j-o_{1j})^2}}\exp\left(-\frac{\sum_{j=1}^{D}(x_j-o_{1j})^2}{200D}\right)+\frac{1}{\sqrt{\sum_{j=1}^{D}(x_j-o_{2j})^2}}\exp\left(-\frac{\sum_{j=1}^{D}(x_j-o_{2j})^2}{800D}\right)+\frac{1}{\sqrt{\sum_{j=1}^{D}(x_j-o_{3j})^2}}\exp\left(-\frac{\sum_{j=1}^{D}(x_j-o_{3j})^2}{1800D}\right)+\frac{1}{\sqrt{\sum_{j=1}^{D}(x_j-o_{4j})^2}}\exp\left(-\frac{\sum_{j=1}^{D}(x_j-o_{4j})^2}{3200D}\right)}\cdot$$
$$\cdot 10\cdot\left(-20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{j=1}^{D}x_j^2}\right)-\exp\left(\frac{1}{D}\sum_{j=1}^{D}\cos(2\pi x_j)\right)-1280+\mathrm{e}\right)+$$
$$+\frac{\frac{1}{\sqrt{\sum_{j=1}^{D}(x_j-o_{2j})^2}}\exp\left(-\frac{\sum_{j=1}^{D}(x_j-o_{2j})^2}{800D}\right)}{\frac{1}{\sqrt{\sum_{j=1}^{D}(x_j-o_{1j})^2}}\exp\left(-\frac{\sum_{j=1}^{D}(x_j-o_{1j})^2}{200D}\right)+\frac{1}{\sqrt{\sum_{j=1}^{D}(x_j-o_{2j})^2}}\exp\left(-\frac{\sum_{j=1}^{D}(x_j-o_{2j})^2}{800D}\right)+\frac{1}{\sqrt{\sum_{j=1}^{D}(x_j-o_{3j})^2}}\exp\left(-\frac{\sum_{j=1}^{D}(x_j-o_{3j})^2}{1800D}\right)+\frac{1}{\sqrt{\sum_{j=1}^{D}(x_j-o_{4j})^2}}\exp\left(-\frac{\sum_{j=1}^{D}(x_j-o_{4j})^2}{3200D}\right)}\cdot$$
$$\cdot\left(1\mathrm{e}{-6}\cdot\left(\sum_{j=1}^{D}(10^6)^{\frac{j-1}{D-1}}x_j^2-1100\right)+100\right)+ \qquad (54)$$
$$+\frac{\frac{1}{\sqrt{\sum_{j=1}^{D}(x_j-o_{3j})^2}}\exp\left(-\frac{\sum_{j=1}^{D}(x_j-o_{3j})^2}{1800D}\right)}{\frac{1}{\sqrt{\sum_{j=1}^{D}(x_j-o_{1j})^2}}\exp\left(-\frac{\sum_{j=1}^{D}(x_j-o_{1j})^2}{200D}\right)+\frac{1}{\sqrt{\sum_{j=1}^{D}(x_j-o_{2j})^2}}\exp\left(-\frac{\sum_{j=1}^{D}(x_j-o_{2j})^2}{800D}\right)+\frac{1}{\sqrt{\sum_{j=1}^{D}(x_j-o_{3j})^2}}\exp\left(-\frac{\sum_{j=1}^{D}(x_j-o_{3j})^2}{1800D}\right)+\frac{1}{\sqrt{\sum_{j=1}^{D}(x_j-o_{4j})^2}}\exp\left(-\frac{\sum_{j=1}^{D}(x_j-o_{4j})^2}{3200D}\right)}\cdot$$
$$\cdot\left(10\cdot\left(\sum_{j=1}^{D}\frac{x_j^2}{4000}-\prod_{j=1}^{D}\cos\left(\frac{x_j}{\sqrt{j}}\right)-1499\right)+200\right)+$$
$$+\frac{\frac{1}{\sqrt{\sum_{j=1}^{D}(x_j-o_{4j})^2}}\exp\left(-\frac{\sum_{j=1}^{D}(x_j-o_{4j})^2}{3200D}\right)}{\frac{1}{\sqrt{\sum_{j=1}^{D}(x_j-o_{1j})^2}}\exp\left(-\frac{\sum_{j=1}^{D}(x_j-o_{1j})^2}{200D}\right)+\frac{1}{\sqrt{\sum_{j=1}^{D}(x_j-o_{2j})^2}}\exp\left(-\frac{\sum_{j=1}^{D}(x_j-o_{2j})^2}{800D}\right)+\frac{1}{\sqrt{\sum_{j=1}^{D}(x_j-o_{3j})^2}}\exp\left(-\frac{\sum_{j=1}^{D}(x_j-o_{3j})^2}{1800D}\right)+\frac{1}{\sqrt{\sum_{j=1}^{D}(x_j-o_{4j})^2}}\exp\left(-\frac{\sum_{j=1}^{D}(x_j-o_{4j})^2}{3200D}\right)}\cdot$$

$$\cdot \left(\sum_{j=1}^{D}\left(x_j^2 - 10\cos(2\pi x_j) + 10\right) - 200\right) + 2400,$$

где o_1, o_2, o_3, o_4 – новые смещённые положения оптимумов, определяющие положения глобального и локальных оптимумов.

На рисунке 1 представлены графики ландшафтов тестовых функций T_1-T_{10} с учётом того, что функции T_1-T_4 и T_8-T_{10} имеют размерность $n = 2$, а функции T_5-T_7 – размерность $n = 10$ (значения первых двух переменных варьируются в заданном диапазоне, а остальные переменные принимают нулевые значения).

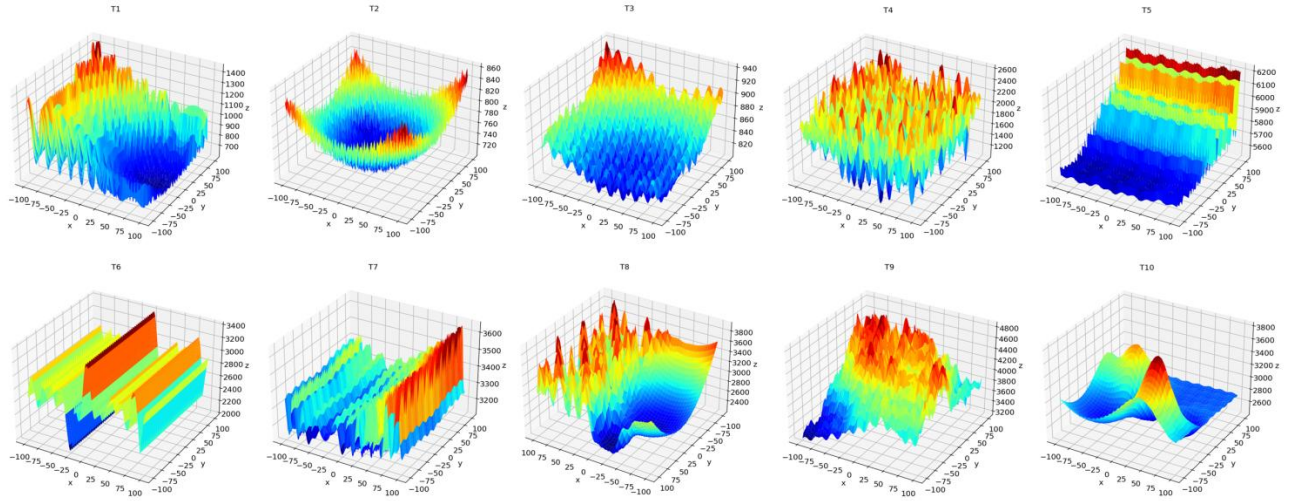

Рисунок 1. Графики ландшафтов тестовых функций для оптимизации

Сравнение результатов тестирования предложенного алгоритма MQIGA-SHA с результатами тестирования алгоритмов MQIGA [11], GA, SelfCGA, PDPGA, SHAGA, CSHAGA, SelfCSHAGA, PDPSHAGA [22] проведено на основе значения метрики профиля точности (Accuracy Profile, далее – AP), определяющей вероятность достижения заданного уровня относительной точности при случайном запуске алгоритма, среднего значения индикатора производительности оптимизации на базе стохастического доминирования (Optimization Performance Indicator based on Stochastic Dominance, далее – OPISD), среднего значения скорости сходимости к оптимуму целевой функции (Convergence Rate, далее – CR) [23] и среднего значения времени выполнения t.

Все тестовые функции T_1-T_{10} определены с размерностью $n = 100$. В алгоритме MQIGA-SHA использован размер истории успеха $H_{size} = 3$. История в MQIGA-SHA инициализируется нулевыми значениями для угла поворота θ_{shift} и начальным значением вероятности мутации $P_{mut} = 0.05$. Для метрики AP использован уровень относительной точности $\tau = 2.0$, что указывает на приближение найденного решения к оптимуму как минимум в 100 раз ближе, чем начальное решение.

Алгоритмы оптимизации выполнены итеративно в течение 30 запусков. Эксперименты проведены на базе вычислительного устройства с процессором Intel Core i9-13980HX и объёмом оперативной памяти 32 Гб.

В таблице 1 приведены результаты выполнения алгоритмов оптимизации на тестовых задачах T_1-T_{10}. Полужирным шрифтом выделены лучшие значения метрик.

Таблица 1. Результаты тестирования алгоритмов оптимизации

Задача	Метрика	MQIGA-SHA	MQIGA	GA	SelfCGA	PDPGA	SHAGA	CSHAGA	SelfCSHAGA	PDPSHAGA
T_1	AP	0.9000	0.8667	0.3933	0.9667	0.9333	0.8000	**1.0000**	**1.0000**	**1.0000**
	OPISD	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9993	**1.0000**	**1.0000**	**1.0000**
	CR	**0.1953**	0.2297	5.3634	17.4169	28.8485	41.6826	0.8926	55.1085	12.8038
	t	5.4101	5.4396	3.6044	0.7181	0.8038	**0.4206**	0.4863	0.6401	0.6554
T_2	AP	0.3333	0.2000	0.0000	0.4667	0.3000	0.0333	0.6000	0.6667	**0.8000**
	OPISD	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
	CR	**0.3962**	0.7941	8.1431	27.8971	98.4581	11.8109	1.0893	19.4558	24.2510
	t	5.1786	6.4116	1.1038	0.7762	0.9767	**0.5189**	0.5739	0.7589	0.7305
T_3	AP	0.7333	0.5333	0.0000	**0.9333**	0.5667	0.1333	**0.9333**	0.8667	0.9000

	OPISD	**1.0000**	0.9989	**1.0000**	**1.0000**	**1.0000**	0.9983	**1.0000**	**1.0000**	**1.0000**
	CR	**0.1196**	0.2101	4.7746	2.1103	9.9488	2.7062	0.5467	0.2410	0.4698
	t	5.3577	5.5065	3.7303	1.0678	1.4983	**0.8023**	0.8614	1.0243	1.0279
T_4	AP	0.5000	0.3667	0.0000	0.5000	0.4333	0.2667	**0.8333**	0.8000	**0.8333**
	OPISD	**1.0000**	**1.0000**	0.9860	**1.0000**	**1.0000**	0.9915	**1.0000**	**1.0000**	**1.0000**
	CR	**0.6301**	0.6352	3.7028	57.1135	95.1968	12.3457	0.8326	9.3299	99.6764
	t	5.1031	5.2686	3.3998	0.7327	0.9561	**0.4920**	0.5405	0.6726	0.7058
T_5	AP	**0.4000**	0.3667	0.0000	0.0000	0.0003	0.1005	0.0000	0.0000	0.0000
	OPISD	**1.0000**	**1.0000**	0.9967	**1.0000**	**1.0000**	0.9958	**1.0000**	**1.0000**	0.9837
	CR	**0.2109**	0.2116	0.9122	0.7300	0.6802	0.7039	0.8980	1.2352	3.6920
	t	26.3456	45.6002	3.3236	0.9040	0.6976	**0.3785**	0.4087	0.7303	0.7211
T_6	AP	**1.0000**	**1.0000**	0.6400	0.1700	0.2350	0.2212	0.2629	0.2146	0.1053
	OPISD	0.9978	0.9939	0.9800	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9790	0.9900
	CR	0.8977	0.7044	0.6257	**0.6086**	0.6260	0.9011	1.0034	1.5892	4.5428
	t	26.7814	47.8540	7.3188	0.7470	0.6849	**0.3418**	0.3976	0.8001	0.7504
T_7	AP	**0.6002**	0.5800	0.0159	0.2791	0.3990	0.3705	0.3999	0.4583	0.5079
	OPISD	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
	CR	**0.4425**	0.8209	2.0994	1.0047	1.0000	0.8324	0.5508	0.4950	0.4483
	t	27.0958	44.8183	7.1788	0.7093	0.6616	**0.3126**	0.3532	0.5267	0.5006
T_8	AP	**1.0000**	**1.0000**	0.4820	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
	OPISD	0.9873	**1.0000**	**1.0000**	0.9967	**1.0000**	**1.0000**	0.9806	0.9878	0.9893
	CR	0.2172	0.4857	1.7173	71.3051	0.1281	**0.0987**	0.1339	0.2556	31.3807
	t	5.4648	8.9539	3.2527	1.0157	1.4659	**0.7847**	0.8088	1.0326	1.0194
T_9	AP	**1.0000**	**1.0000**	0.0091	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
	OPISD	**1.0000**	**1.0000**	0.9850	0.9892	**1.0000**	0.9944	**1.0000**	0.9926	0.9968
	CR	0.8435	0.9031	9.7420	25.0184	21.8204	4.0644	**0.1558**	66.4248	31.0327
	t	5.4641	9.5684	2.6979	1.0946	1.6102	**0.8524**	0.8834	1.0827	1.0507
T_{10}	AP	**1.0000**	**1.0000**	0.0000	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
	OPISD	**1.0000**	**1.0000**	**1.0000**	0.9981	**1.0000**	**1.0000**	**1.0000**	0.9951	0.9900
	CR	0.3472	0.5409	2.8408	55.8863	**0.0045**	0.1595	3.3106	0.1122	0.3037
	t	5.5306	10.3124	3.5215	1.1453	1.7390	**0.9198**	0.9590	1.2017	1.0943

Результаты тестирования алгоритмов оптимизации, представленные в таблице 1, показывают, что алгоритм MQIGA-SHA демонстрирует высокую надёжность. Значение метрики AP=1.0000 зафиксировано на функциях T_5-T_{10} (60% экспериментов), что указывает на достижение целевой точности $\tau = 2.0$ во всех 30 запусках. На остальных функциях T_1-T_4 (40% экспериментов) значение метрики AP для алгоритма MQIGA-SHA составляет от 0.3333 до 0.9000, что превосходит результаты алгоритмов MQIGA, GA, SHAGA на функции T_1 и результаты алгоритмов MQIGA, GA, PDPGA, SHAGA на функциях T_2, T_3 и T_4. В то же время классический алгоритм GA и адаптивные алгоритмы SelfCGA, PDPGA, SHAGA, CSHAGA, SelfCSHAGA и PDPSHAGA показывают низкие значения AP на гибридных функциях T_5-T_7, что свидетельствует об их неспособности решать сложные мультимодальные задачи.

Метрика OPISD подтверждает высокое качество траектории сходимости MQIGA-SHA. Максимальное значение OPISD=1.0000 достигается алгоритмом в 80% экспериментов на функциях T_1-T_5, T_7, T_9, T_{10}, что указывает на превосходство траектории сходимости MQIGA-SHA над линейной моделью улучшения.

По метрике CR алгоритм MQIGA-SHA показывает наименьшие значения на функциях T_1-T_5, T_7 (60% экспериментов), соответствующие наиболее высокой скорости сходимости к оптимуму. В то же время

алгоритмы SelfCGA, PDPGA, SHAGA, SelfCSHAGA, PDPSHAGA демонстрируют аномально высокие значения метрики CR на функциях T_1, T_2, T_4 и T_9, что свидетельствует о слишком медленной и неустойчивой сходимости, несмотря на низкое время выполнения.

Анализ времени выполнения t показывает, что MQIGA-SHA требует в среднем 5.4-6.2 секунд на один запуск, что значительно превышает время работы алгоритма GA и остальных классических адаптивных алгоритмов. Однако данное увеличение вычислительных затрат компенсируется существенных ростом надёжности и качества решения. Например, на функции T_5 алгоритм MQIGA-SHA при среднем времени выполнения 26.4 секунд достигает AP=0.4000, OPISD=1.0000 и CR=0.2109, в то время как алгоритмы GA, SelfCGA, PDPGA, SelfCSHAGA, PDPSHAGA при среднем времени выполнения 0.4-3.3 секунд имеют AP≈0.0000, OPISD≈1.0000 и CR в диапазоне от 3.6920 до 0.6802.

Сравнительный анализ полученных результатов экспериментов позволяет сделать вывод о том, что предложенный алгоритм MQIGA-SHA демонстрирует систематическое превосходство над рассматриваемыми адаптивными генетическими алгоритмами оптимизации по совокупности ключевых показателей эффективности, а также свидетельствует о том, что алгоритм MQIGA-SHA обеспечивает наилучший баланс между надёжностью, скоростью и качеством оптимизации. В частности, алгоритм показывает максимальную или близкую к максимальной вероятность успеха, что подтверждается AP≥0.5000 в задачах T_1, T_3, T_4, T_6-T_{10} (80% экспериментов), включая наиболее сложные гибридные и композитные функции. Одновременно с этим алгоритм MQIGA-SHA демонстрирует наиболее эффективную динамику поиска, что подтверждается стабильно высокими значениями метрики OPISD≈1.0000 и наименьшими значениями метрики CR в большинстве задач. Хотя вычислительные затраты MQIGA-SHA превышают затраты классических адаптивных алгоритмов, это увеличение оправдывается достигаемым качеством решений – MQIGA-SHA не допускает полного спада вероятности по метрике профиля точности AP до 0% ни в одной из тестовых функций. Полученные данные свидетельствуют о том, что интеграция квантово-инспирированной модели представления решений с механизмом адаптации значений параметров на основе истории успеха позволяет создать робастный оптимизационный инструмент, способный эффективно решать широкий спектр задач, включая многоэкстремальные, зашумленные и плохо обусловленные функции.

Заключение

В ходе проведённого исследования разработан подход к повышению эффективности квантово-инспирированных эволюционных алгоритмов за счёт интеграции механизма динамической самонастройки параметров на основе истории успеха. Предложенный алгоритм MQIGA-SHA впервые объединяет преимущества квантово-инспирированного моделирования, в частности, использование многоуровневых квантовых систем и физически обоснованной модели декогеренции, с гибким и самообучающимся механизмом адаптации значений ключевых параметров эволюции. Анализ существующих методов показал, что эффективность даже самых продвинутых квантово-инспирированных алгоритмов, таких как QSA и AHQSOA, остаётся ограниченной из-за необходимости ручной настройки или использования фиксированных эвристик для управления параметрами. Это создаёт существенный разрыв между теоретическим потенциалом квантово-инспирированных алгоритмов и их практической применимостью к разнородным задачам оптимизации. Предложенный в работе алгоритм MQIGA-SHA устраняет этот пробел, реализуя полностью автономную стратегию адаптации. Ключевой инновацией является использование взвешенного среднего Лемера второго порядка для обновления значений угла поворота и вероятности мутации на основе не просто факта улучшения решения, но и степени этого улучшения. Данный подход позволяет алгоритму MQIGA-SHA не только запоминать успешные настройки, но и придавать больший вес тем из них, которые привели к значительному улучшению качества решений. В совокупности с циклической историей фиксированной длины и пороговым ограничением это создаёт устойчивый, саморегулирующийся алгоритм, способный эффективно балансировать между исследованием и эксплуатацией на всех этапах эволюционного процесса. Экспериментальная оценка подтвердила высокую эффективность и робастность предложенного алгоритма. Алгоритм MQIGA-SHA демонстрирует систематическое превосходство над базовой версией MQIGA и рядом современных адаптивных генетических алгоритмов по профилю точности, показателю стохастического доминирования и скорости сходимости. Необходимо добавить, что MQIGA-SHA сохраняет высокую надёжность в том числе на наиболее сложных гибридных и композитных функциях, не допуская ни в одном из запусков падения вероятности успеха до нуля. Хотя это достигается за счёт повышенных вычислительных затрат, получаемый прирост в качестве и надёжности решений полностью оправдывает данные издержки, особенно в условиях «чёрного ящика», где надёжность является критически важным фактором. Стоит отметить, что предложенный алгоритм MQIGA-SHA

не только решает конкретную задачу повышения эффективности вещественной оптимизации, но и демонстрирует общий принцип построения интеллектуальных оптимизационных систем, способных к обучению на собственном опыте.

Список литературы

1. Pehnec I., Sedlar D., Marinic-Kragic I., Vučina D. A New Approach to Topology Optimization with Genetic Algorithm and Parameterization Level Set Function. Computation. 2025. vol. 13. no. 7. p. 153. DOI: 10.3390/computation13070153.
2. Kim Y.-H., Geem Z.W., Yoon Y. Population-Based Redundancy Control in Genetic Algorithms: Enhancing Max-Cut Optimization. Mathematics. 2025. vol. 13. no. 9. p. 1409. DOI: 10.3390/math13091409.
3. Wu L. A note on no-free-lunch theorem. Discrete Mathematics, Algorithms and Applications. 2021. vol. 14. DOI: 10.1142/S1793830922500276.
4. Шерстнев П.А., Семенкин Е.С. SelfCSHAGA: самоконфигурируемый генетический алгоритм оптимизации с адаптацией на основе истории успеха. Вестник Московского государственного технического университета им. Н.Э. Баумана. Серия: Приборостроение. 2025. № 2(151). С. 122-139. EDN: TSKBOX.
5. Шерстнев П.А., Семенкин Е.С. Самоконфигурируемые алгоритмы генетического программирования с адаптацией на основе истории успеха. Сибирский аэрокосмический журнал. 2025. Т. 26. № 1. С. 60-70. DOI: 10.31772/2712-8970-2025-26-1-60-70.
6. Tanabe R., Fukunaga A. Success-history based parameter adaptation for Differential Evolution. Proceedings of the 2013 IEEE Congress on Evolutionary Computation, Cancun, Mexico. IEEE; 2013. pp. 71-78. DOI: 10.1109/CEC.2013.6557555.
7. Stanovov V., Akhmedova S., Semenkin E. Genetic Algorithm with Success History based Parameter Adaptation. Proceedings of the 11th International Joint Conference on Computational Intelligence (IJCCI 2019) – ECTA. SciTePress; 2019. pp. 180-187. DOI: 10.5220/0008071201800187.
8. Hakemi S., Houshmand M., KheirKhah E. et al. A review of recent advances in quantum-inspired metaheuristics. Evolutionary Intelligence. 2024. vol. 17. pp. 627-642. DOI: 10.1007/s12065-022-00783-2.
9. Zhang G. Quantum-inspired evolutionary algorithms: a survey and empirical study. Journal of Heuristics. 2011. vol. 17. pp. 303-351. DOI: 10.1007/s10732-010-9136-0.
10. Iovane G. Quantum-Inspired Algorithms and Perspectives for Optimization. Electronics. 2025. vol. 14. no. 14. p. 2839. DOI: 10.3390/electronics14142839.
11. Масленников В.В., Демидова Л.А. Модификация квантово-инспирированного генетического алгоритма численной оптимизации с использованием кудита в условиях имитации квантовой декогеренции. Computational Nanotechnology. 2024. Т. 11. № 2. С. 58-85. DOI: 10.33693/2313-223X-2024-11-2-58-85. EDN: MRWGYA.
12. Oralhan Z., Oralhan B. Quantum Snowflake Algorithm (QSA): A Snowflake-Inspired, Quantum-Driven Metaheuristic for Large-Scale Continuous and Discrete Optimization with Application to the Traveling Salesman Problem. Applied Sciences. 2025. vol. 15. no. 9. p. 5117. DOI: 10.3390/app15095117.
13. Rupa A., Devi A., Pranay V., Krushna S., Ramya G., Janakiraman R., Adithya B., Ranjith Jr. Adaptive Hybrid Quantum-Inspired Optimization for Enhanced Global Search Efficiency. Journal of Information Systems Engineering & Management. 2025. vol. 10. no. 31s. pp. 60-67. DOI: 10.52783/jisem.v10i31s.5011.
14. Kikuchi A. A Formulation of Quantum Particle Swarm Optimization. 2018.
15. Khaer A. Quantum Evolutionary Algorithm based on Particle Swarm theory in multiobjective problems. 2010.
16. Philip-Kpae F.O., Nkolika N., Toobari B.S. Quantum-inspired Optimization for Efficient Vehicular Edge Computing Resource Allocation in Intelligent Transportation Systems. American Journal of Networks and Communications. 2025. vol. 14. no. 2. pp. 47-58. DOI: 10.11648/j.ajnc.20251402.13.
17. Khorrami M. Tuning the Probabilistic Distribution and the Gini Index. Brazilian Journal of Physics. 2025. vol. 55. p. 243. DOI: 10.1007/s13538-025-01879-7.
18. Preskill J. Quantum Computing in the NISQ era and beyond. Quantum. 2018. vol. 2. p. 79. DOI: 10.22331/q-2018-08-06-79.
19. Li Q. Wigner's friend scenario, Born's rule and a new interpretation of quantum mechanics. 2023. DOI: 10.21203/rs.3.rs-3270399/v1.
20. Johannesson B. Sequences related to Lehmer's problem. arXiv preprint arXiv:2308.16305. 2023. DOI: 10.48550/arXiv.2308.16305.
21. Wu G., Mallipeddi R., Suganthan P. Problem Definitions and Evaluation Criteria for the CEC 2017 Competition and Special Session on Constrained Single Objective Real-Parameter Optimization. 2016.

22. The fittest – Evolutionary machine learning in Python // URL: https://github.com/sherstpasha/thefittest/ (дата обращения: 30.10.2025).

23. Halim A.H., Ismail I., Das S. Performance assessment of the metaheuristic optimization algorithms: an exhaustive review. Artificial Intelligence Review. 2021. vol. 54. pp. 2323-2409. DOI: 10.1007/s10462-020-09906-6.

References

1. Pehnec I., Sedlar D., Marinic-Kragic I., Vučina D. A New Approach to Topology Optimization with Genetic Algorithm and Parameterization Level Set Function. Computation. 2025. vol. 13. no. 7. p. 153. DOI: 10.3390/computation13070153.

2. Kim Y.-H., Geem Z.W., Yoon Y. Population-Based Redundancy Control in Genetic Algorithms: Enhancing Max-Cut Optimization. Mathematics. 2025. vol. 13. no. 9. p. 1409. DOI: 10.3390/math13091409.

3. Wu L. A note on no-free-lunch theorem. Discrete Mathematics, Algorithms and Applications. 2021. vol. 14. DOI: 10.1142/S1793830922500276.

4. Sherstnev P.A., Semenkin E.S. SelfCSHAGA: A self-configuring genetic optimization algorithm with success-history-based adaptation. Herald of the Bauman Moscow State Technical University. Series: Instrument Engineering. 2025. No. 2(151). Pp. 122–139. EDN: TSKBOX.

5. Sherstnev P.A., Semenkin E.S. Self-configuring genetic programming algorithms with success-history-based adaptation. Siberian Aerospace Journal. 2025. Vol. 26. No. 1. Pp. 60–70. DOI: 10.31772/2712-8970-2025-26-1-60-70.

6. Tanabe R., Fukunaga A. Success-history based parameter adaptation for Differential Evolution. Proceedings of the 2013 IEEE Congress on Evolutionary Computation, Cancun, Mexico. IEEE; 2013. pp. 71-78. DOI: 10.1109/CEC.2013.6557555.

7. Stanovov V., Akhmedova S., Semenkin E. Genetic Algorithm with Success History based Parameter Adaptation. Proceedings of the 11th International Joint Conference on Computational Intelligence (IJCCI 2019) – ECTA. SciTePress; 2019. pp. 180-187. DOI: 10.5220/0008071201800187.

8. Hakemi S., Houshmand M., KheirKhah E. et al. A review of recent advances in quantum-inspired metaheuristics. Evolutionary Intelligence. 2024. vol. 17. pp. 627-642. DOI: 10.1007/s12065-022-00783-2.

9. Zhang G. Quantum-inspired evolutionary algorithms: a survey and empirical study. Journal of Heuristics. 2011. vol. 17. pp. 303-351. DOI: 10.1007/s10732-010-9136-0.

10. Iovane G. Quantum-Inspired Algorithms and Perspectives for Optimization. Electronics. 2025. vol. 14. no. 14. p. 2839. DOI: 10.3390/electronics14142839.

11. Maslennikov V.V., Demidova L.A. Modification of a quantum-inspired genetic algorithm for numerical optimization using qudits under simulated quantum decoherence. Computational Nanotechnology. 2024. Vol. 11. No. 2. Pp. 58–85. DOI: 10.33693/2313-223X-2024-11-2-58-85. EDN: MRWGYA.

12. Oralhan Z., Oralhan B. Quantum Snowflake Algorithm (QSA): A Snowflake-Inspired, Quantum-Driven Metaheuristic for Large-Scale Continuous and Discrete Optimization with Application to the Traveling Salesman Problem. Applied Sciences. 2025. vol. 15. no. 9. p. 5117. DOI: 10.3390/app15095117.

13. Rupa A., Devi A., Pranay V., Krushna S., Ramya G., Janakiraman R., Adithya B., Ranjith Jr. Adaptive Hybrid Quantum-Inspired Optimization for Enhanced Global Search Efficiency. Journal of Information Systems Engineering & Management. 2025. vol. 10. no. 31s. pp. 60-67. DOI: 10.52783/jisem.v10i31s.5011.

14. Kikuchi A. A Formulation of Quantum Particle Swarm Optimization. 2018.

15. Khaer A. Quantum Evolutionary Algorithm based on Particle Swarm theory in multiobjective problems. 2010.

16. Philip-Kpae F.O., Nkolika N., Toobari B.S. Quantum-inspired Optimization for Efficient Vehicular Edge Computing Resource Allocation in Intelligent Transportation Systems. American Journal of Networks and Communications. 2025. vol. 14. no. 2. pp. 47-58. DOI: 10.11648/j.ajnc.20251402.13.

17. Khorrami M. Tuning the Probabilistic Distribution and the Gini Index. Brazilian Journal of Physics. 2025. vol. 55. p. 243. DOI: 10.1007/s13538-025-01879-7.

18. Preskill J. Quantum Computing in the NISQ era and beyond. Quantum. 2018. vol. 2. p. 79. DOI: 10.22331/q-2018-08-06-79.

19. Li Q. Wigner's friend scenario, Born's rule and a new interpretation of quantum mechanics. 2023. DOI: 10.21203/rs.3.rs-3270399/v1.

20. Johannesson B. Sequences related to Lehmer's problem. arXiv preprint arXiv:2308.16305. 2023. DOI: 10.48550/arXiv.2308.16305.

21. Wu G., Mallipeddi R., Suganthan P. Problem Definitions and Evaluation Criteria for the CEC 2017 Competition and Special Session on Constrained Single Objective Real-Parameter Optimization. 2016.

22. The fittest – Evolutionary machine learning in Python // URL: https://github.com/sherstpasha/thefittest/ (accessed on October 30, 2025).

23. Halim A.H., Ismail I., Das S. Performance assessment of the metaheuristic optimization algorithms: an exhaustive review. Artificial Intelligence Review. 2021. vol. 54. pp. 2323-2409. DOI: 10.1007/s10462-020-09906-6.