

СРАВНИТЕЛЬНЫЙ АНАЛИЗ СКОРОСТИ СХОДИМОСТИ ПАРАЛЛЕЛЬНОГО АДАПТИВНОГО И ТРАДИЦИОННОГО ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ

COMPARATIVE ANALYSIS OF THE CONVERGENCE RATE OF PARALLEL ADAPTIVE AND TRADITIONAL GENETIC ALGORITHMS

A. Chernikov

Summary. The article presents a comparative analysis of two popular optimization methods: the adaptive population algorithm and the traditional genetic algorithm. The main purpose of the study is to evaluate and compare the convergence rate of these two algorithms. The convergence rate is critically important for optimization problems, as it determines the effectiveness of the algorithm in achieving an optimal solution. As part of the study, several test functions were selected, covering a wide range of complexity, from simple non-convex to complex multimodal. This allowed for a comprehensive analysis, evaluating algorithms on a variety of data. Various parameters of the algorithms were considered, including population size, probability of mutation and crossover, as well as selection strategies. The results showed that adaptive population algorithms demonstrate a significant increase in the convergence rate on many test functions, especially in multimodal scenarios. Based on these observations, conclusions are drawn about the applicability and effectiveness of each approach in various areas of optimization. This study represents an important contribution to understanding the dynamics and effectiveness of adaptive and traditional approaches in optimization algorithms, highlighting their strengths and weaknesses in different settings.

Keywords: adaptive population algorithm, genetic algorithm, convergence rate, optimization, multimodal function.

Черников Александр Владимирович

аспирант, ФГБОУ ВО «МГТУ «СТАНКИН», Москва
aleksandrchernikov98@gmail.com

Аннотация. В статье представлен сравнительный анализ двух популярных методов оптимизации: адаптивного популяционного алгоритма и традиционного генетического алгоритма. Основной целью исследования является оценка и сравнение скорости сходимости этих двух алгоритмов. Скорость сходимости критически важна для оптимизационных задач, поскольку она определяет эффективность алгоритма в достижении оптимального решения. В рамках исследования были выбраны несколько тестовых функций, охватывающих широкий спектр сложности, от простых невыпуклых до сложных мультимодальных. Это позволило провести всесторонний анализ, оценивая алгоритмы на разнообразных данных. Были рассмотрены различные параметры алгоритмов, включая размер популяции, вероятность мутации и кроссовера, а также стратегии отбора. Результаты показали, что адаптивные популяционные алгоритмы демонстрируют значительное повышение скорости сходимости на многих тестовых функциях, особенно в мультимодальных сценариях. На основе этих наблюдений делаются выводы о применимости и эффективности каждого подхода в различных областях оптимизации. Это исследование представляет собой важный вклад в понимание динамики и эффективности адаптивных и традиционных подходов в оптимизационных алгоритмах, подчеркивая их сильные и слабые стороны в разных условиях.

Ключевые слова: адаптивный популяционный алгоритм, генетический алгоритм, скорость сходимости, оптимизация, мультимодальная функция.

Введение

Поиск экстремумов нелинейных мультимодальных функций с ограничениями представляет собой ключевую задачу в области оптимизации, играющую критическую роль в различных сферах, таких как инженерное проектирование, экономический анализ, машинное обучение и другие области [1]. Эта задача призвана найти оптимальные решения в условиях, где целевые функции характеризуются наличием множества локальных экстремумов, что делает задачу особенно сложной и интересной для исследований.

Основной проблемой при работе с такими функциями является поиск их экстремумов, что требует особого

подхода в анализе и вычислительных методах [2]. В контексте оптимизационных задач широко используются популяционные алгоритмы, которые представляют собой мощный инструмент для исследования пространства решений [3, 4]. Эти алгоритмы, работая с группой потенциальных решений или популяцией, могут эффективно исследовать сложные пространства поиска [5], однако их эффективность во многом зависит от правильного выбора параметров. Задача выбора оптимальных параметров и проблема низкой скорости сходимости являются ключевыми аспектами, требующими внимания при использовании популяционных алгоритмов [6, 9–11].

Эксперименты, проведенные в рамках данной работы, включали тестирование алгоритма на различных ма-

тематических функциях в двумерном пространстве. Эти функции были выбраны из-за их разнообразия и способности демонстрировать различные вызовы, с которыми может столкнуться алгоритм при поиске экстремумов. Результаты этих экспериментов показывают значительное улучшение скорости сходимости по сравнению с традиционными методами.

Таким образом, предложенный адаптивный популяционный алгоритм представляет собой перспективное решение в области оптимизационных вычислений, открывая новые возможности для более эффективного решения сложных задач в различных областях науки и техники.

Адаптивные генетические алгоритмы

Адаптивные генетические алгоритмы — это разновидность генетических алгоритмов, которые динамически корректируют свои параметры для повышения производительности. Они разработаны для удовлетворения потребности в тщательной настройке традиционных генетических алгоритмов, при этом не отнимающей много времени. Было показано, что адаптивные алгоритмы эффективны при решении сложных задач и могут привести к значительному повышению производительности по сравнению с неадаптивными генетическими алгоритмами [21, 22]. Адаптация (настройка) в адаптивных алгоритмах может включать: динамическую автоконфигурацию параметров алгоритма генетического поиска, гибридное кодирование, стратегии элитарности и стратегии разнообразия популяции. Исследования [17-20] продемонстрировали эффективность и надежность адаптивных алгоритмов в различных задачах оптимизации и поиска. Адаптивные алгоритмы считаются многообещающим и значимым вариантом генетических алгоритмов, особенно для моделирования сложных адаптивных систем и эволюционных процессов.

Адаптивные генетические алгоритмы, несмотря на вышеописанные преимущества, также имеют и некоторые недостатки. Одним из основных недостатков является проблема выбора параметров алгоритма, которая может быть сложной из-за большого спектра возможных параметров, которые необходимо настроить. Кроме того, адаптивные генетические алгоритмы могут быть дорогостоящими с точки зрения вычислений, особенно при решении крупномасштабных задач. Другим недостатком является то, что адаптивные генетические алгоритмы не всегда могут сходиться к глобальному оптимальному решению, и существует вероятность преждевременной сходимости. Однако эти недостатки могут быть смягчены тщательной настройкой параметров и использованием соответствующих стратегий отбора [23]. В целом преимущества адаптивных генетических алгоритмов перевешивают их недостатки.

Описание разработанного адаптивного алгоритма

Предложенный адаптивный алгоритм подбора и калибровки параметров параллельного популяционного алгоритма, основанный на серии взаимосвязанных шагов [16], представляет собой перспективный подход к оптимизации, позволяющий повысить эффективность поиска экстремумов нелинейных мультимодальных функций. Важным аспектом данного алгоритма является его способность к самонастройке, что позволяет адаптироваться к изменяющимся условиям оптимизационной задачи. Ниже представлено подробное описание каждого шага алгоритма:

Шаг 1: Инициализация параметров алгоритма. Инициализация является первым и самым важным этапом, на котором устанавливаются основные параметры работы алгоритма. Эти параметры включают в себя определение размера популяции, максимального количества итераций, вероятности кроссовера и шанс мутации. Эти параметры могут быть установлены вручную, основываясь на предварительном опыте или предыдущих исследованиях, либо же заданы автоматически с использованием опыта предыдущих испытаний, проводимых в системе. В случае автоматического подхода, система пользуется расчетом расстояния Левенштейна для определения степени схожести между текущей задачей оптимизации и предыдущими случаями, которые хранятся в базе данных, что позволяет адаптировать параметры алгоритма более эффективно. Также для алгоритма сохраняется история задания параметров алгоритма для последующего использования.

Шаг 2: Генерация начальной популяции. На втором шаге происходит создание начальной популяции решений. Эти решения генерируются случайным образом, но при этом удовлетворяют всем заданным ограничениям. Например, ограничения типа диапазона значений переменных, специфические условия задачи или другие факторы, которые определяют допустимость решения. Инициализация начальной популяции также является одним из ключевых моментов, поскольку от нее зависит разнообразие и потенциальная эффективность последующего поиска решений.

Шаг 3: Определение лучшего решения. На этом этапе каждое решение из популяции оценивается на основе целевой функции задачи оптимизации. Целевая функция определяет пригодность каждого решения — чем меньше значение функции, тем лучше решение (для задач минимизации). Из всей популяции отбираются наилучшие решения, которые показывают лучшую пригодность. В процессе поиска оптимального решения из сгенерированной популяции выбираются решения с наилучшей пригодностью.

Шаг 4: Скрещивание (кроссовер или кроссинговер). На этом шаге из текущей популяции случайным образом выбираются два родительских решения. Далее происходит процесс комбинирования их генетического материала для создания новых потомков. Этот процесс может быть выполнен различными способами, включая одно-точечный, двухточечный или равномерный кроссовер. В результате кроссовера образуются потомки, которые унаследуют характеристики обоих родителей, что может привести к появлению решений, сочетающих в себе их лучшие черты. Это позволяет популяции обогатиться новыми и, возможно, более пригодными индивидами.

Шаг 5: Мутация. Мутация — это процесс случайного изменения генетического материала отдельного решения в популяции. Этот процесс выполняется с некоторой малой вероятностью и является фундаментальным для внесения разнообразия в популяцию. Мутация может изменять один или несколько генов в генотипе индивида, что приводит к появлению новых характеристик. Это может помочь алгоритму избежать застревания в локальных минимумах, исследовать неизведанные области пространства решений и повысить общую способность популяции к адаптации. Важно правильно балансировать вероятность мутации, чтобы избежать чрезмерной случайности и сохранить полезные характеристики, накопленные в процессе эволюции.

Шаг 6: Генерация новой популяции. На этом шаге происходит создание новой популяции, которая заменит текущую. В этом процессе используются подстроенные параметры основного алгоритма, включая скорость кроссовера, вероятность мутации и методы отбора. Новая популяция формируется путем отбора наиболее пригодных индивидов из старой популяции, а также включения новых потомков, полученных в результате кроссовера и мутации. Важным аспектом является управление размером популяции, что может включать в себя стратегии, такие как элитизм (сохранение лучших индивидов) или генерационное обновление (замена всей популяции). Этот этап обеспечивает дальнейшее развитие популяции и является важным аспектом для постепенного улучшения качества найденных решений.

Шаг 7: Проверка критерия останова. На этом этапе алгоритм проверяет, выполнены ли условия для завершения работы. Критерии могут включать достижение заданного уровня точности решения, превышение максимального числа итераций или отсутствие значительного улучшения решения за определенное количество шагов. Если условия останова выполнены, алгоритм завершается и возвращает наилучшее найденное решение. Этот механизм предотвращает бесконечные вычисления и обеспечивает возврат оптимизированного результата. Если выполняется критерий останова, то алгоритм завершает работу и возвращает лучшее найденное решение. В противном случае алгоритм переходит на шаг 8.

Шаг 8: Обновление истории параметров алгоритма. На данном этапе происходит запись текущих значений всех параметров алгоритма, а также соответствующих значений пригодности решения. Это включает в себя детальный анализ каждого параметра, таких как вероятность мутации, размер популяции, скорость селекции и другие. Сохранение этой истории позволяет алгоритму анализировать свое прошлое поведение и оптимизировать дальнейшие действия.

Шаг 9. Оценка весов параметров на основании истории. На этом шаге алгоритм анализирует прошлую эффективность каждого параметра, чтобы определить, какие из них наиболее успешно способствовали нахождению оптимальных решений. Этот процесс включает в себя сложную статистическую обработку и анализ данных, чтобы выявить тенденции и закономерности в поведении параметров. В этом контексте используется популяционный алгоритм, где популяцией являются исторические данные о параметрах, а функцией пригодности — качество решения, полученное с использованием конкретного набора параметров. На основе этого анализа алгоритм может корректировать величину изменения параметров для следующих итераций, что повышает шансы на более эффективное нахождение оптимального решения.

Шаг 10. Калибровка и запуск основного алгоритма. Затем проводится подстройка параметров основного алгоритма путем изменения значений вероятности кроссовера, вероятности мутации и других параметров в соответствии с текущей историей параметров алгоритма. В результате калибровки получают новые значения внутренних параметров, которые затем передаются основному алгоритму для решения задачи поиска экстремума заданной функции.

В целом, данный подход позволяет создать более гибкий и адаптивный инструмент для решения сложных оптимизационных задач, предоставляя значительные преимущества по сравнению с традиционными методами.

Методология эксперимента

В рамках исследования были проведены серии экспериментов, направленные на оценку эффективности адаптивного параллельного популяционного алгоритма в сравнении с традиционным параллельным популяционным алгоритмом с фиксированными параметрами. Целью этих экспериментов было выяснить, какие из алгоритмов показывает лучшую скорость сходимости в задачах поиска экстремумов нелинейных мультимодальных функций.

Оба алгоритма были настроены с четырьмя генетическими алгоритмами, работающими параллельно. Каж-

дый генетический алгоритм обменивался некоторым количеством лучших решений с остальными тремя, что способствовало обмену информацией и улучшению решений в каждом из алгоритмов.

Базовый размер популяции для каждого из генетических алгоритмов составлял 1000 особей. Применялся турнирный метод селекции с вероятностью мутации 5 %. Скрещивание было многоточечным, что предоставляло большую генетическую разнообразность внутри популяции.

Поиск решений осуществлялся в двумерном пространстве с переменными X и Y , каждая из которых варьировалась в диапазоне от -100 до 100 . Это позволяло оценить способность алгоритмов работать в широком пространстве решений.

Эксперименты проводились в разработанной системе оптимизации математических функций на персональной ЭВМ с характеристиками: ЦП Intel Core i7-12700, тактовая частота 2,7 ГГц; оперативная память 64 ГБ. Это обеспечивало достаточную вычислительную мощность для проведения экспериментов.

Для исследования были выбраны четыре математических функций: Растригина, Розенброка, Леви, Химмельблау. Для удобства проведения экспериментов, направленных на оценку эффективности адаптивного параллельного популяционного алгоритма в сравнении с традиционным параллельным популяционным алгоритмом с фиксированными параметрами, оптимизируемые функции были разделены на две группы — группу функций с высокой сложностью и группу функций с низкой сложностью. Функции определялись как принадлежащие одной из групп после оценки сложности ландшафта, количества локальных минимумов и общей сложности формы функции.

Группа с низкой сложностью включает функцию Розенброка и функцию Химмельблау. Эти функции имеют более простой ландшафт с меньшим количеством локальных минимумов, что облегчает задачу оптимизации.

Функция Розенброка, также известная как «бананная функция», она имеет извилистую форму с четко выраженным глобальным минимумом. Хотя она и представляет собой сложную задачу для некоторых алгоритмов из-за узкой долины, ведущей к минимуму, в целом она менее сложна по сравнению с другими функциями. Функция часто используется для проверки способности алгоритмов оптимизации следовать сложным путям к минимуму.

Функция Химмельблау имеет несколько локальных минимумов, но они являются относительно легкими для

определения. Ее ландшафт сравнительно простой, с четкими областями минимумов. Функция часто используется для демонстрации эффективности алгоритмов оптимизации в нахождении множественных решений. Функция имеет сравнительно плоский и простой ландшафт. Подходит для начального этапа тестирования алгоритмов, чтобы убедиться в их способности находить базовые решения.

Группа с высокой сложностью включает функцию Растригина и функцию Леви. В этих функциях ландшафт содержит множество локальных минимумов, что и делает задачу оптимизации более сложной.

Функция Растригина характеризуется большим количеством локальных минимумов, расположенных периодически. Это делает ее одной из самых сложных функций для глобальной оптимизации, так как легко «застрять» в локальном минимуме. Оптимальное решение находится в центре, но из-за волновой структуры поверхности найти его бывает непросто.

Функция Леви сложна из-за своего множества локальных минимумов и резких изменений в ландшафте функции. Она имеет характерные «волновые» узоры, которые затрудняют нахождение глобального минимума. Поверхность функции имеет несколько «долин» и «гребней», что добавляет сложности в процесс оптимизации.

Это разделение позволяет сравнивать, разработанный адаптивный параллельный генетический алгоритм справляется с различной степенью сложности ландшафта функций. Такой подход помогает определить, насколько быстро алгоритм может сходиться при поисках глобальных минимумов в условиях с низкой и высокой сложностью ландшафта.

Оптимизация функций с низкой сложностью

Функция Розенброка для двумерного пространства:

$$f(x, y) = (1 - x)^2 + 100(y - x^2)^2$$

Функция Розенброка [25] — невыпуклая функция, которая была разработана и предложена Ховардом Розенброком в 1960 году. Также считается классической функцией, используемой для проверки эффективности стохастических популяционных алгоритмов. Считается, что поиск глобального минимума для данной функции является нетривиальной задачей.

В общем виде формула имеет вид:

$$f(x) = \sum_{i=1}^{n-1} \left[(1 - x_i^2) + 100(x_{i+1} - x_i^2)^2 \right]$$

Функция Розенброка имеет глобальный экстремум в точках:

$$\begin{cases} f(1,1) = 0, n = 2 \\ f(1,1,1) = 0, n = 3 \\ f(1, \dots, 1) = 0, n > 3 \end{cases}$$

где n — это размерность пространства.

На рисунке 1 изображен 3D график поверхности функции Розенброка.

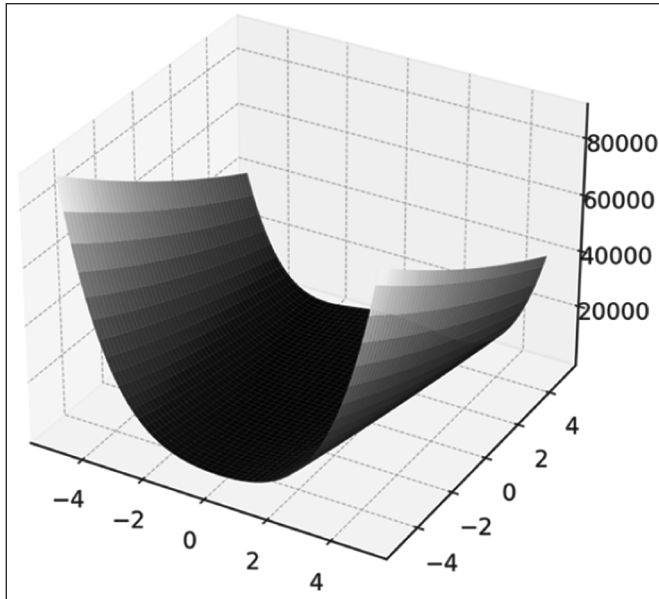


Рис. 1. 3D график поверхности функции Розенброка

Функция Химмельблау:

$$f(x, y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2$$

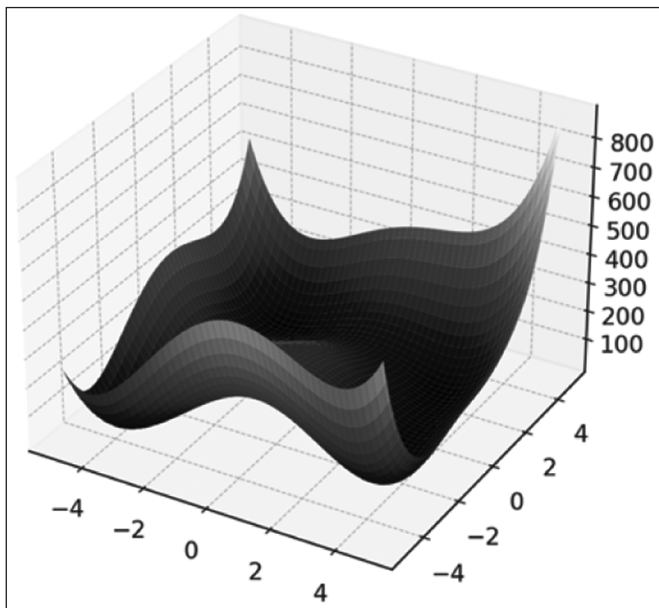


Рис. 2. 3D график поверхности функции Химмельблау

Функция Химмельблау [26] — функция, названная в честь Дэвида Мотнера Химмельблау, который впервые ее использовал в 1972 году. Функция представлена только для двумерного пространства и не имеет расширения на большие размерности области поиска. Является мультимодальной, имеет несколько обманных минимумов и только одну точку глобального экстремума.

Функция Химмельблау имеет глобальные экстремумы в точках:

$$\begin{cases} f(3.0, 2.0) = 0 \\ f(-2.805118, 3.131312) = 0 \\ f(-3.779310, -3.283186) = 0 \\ f(3.584428, -1.848126) = 0 \end{cases}$$

Оптимизация функций с высокой сложностью

Функция Леви:

$$f(x, y) = \sin^2(3\pi x) + (x - 1)^2(1 + \sin^2(3\pi y)) + (y - 1)^2(1 + \sin^2(2\pi y))$$

Функция Леви [29] определена в двумерном пространстве и представляет собой непрерывную, мультимодальную дифференцируемую функцию.

Глобальный экстремум функции Леви находится в точке:

$$f(1, 1) = 0$$

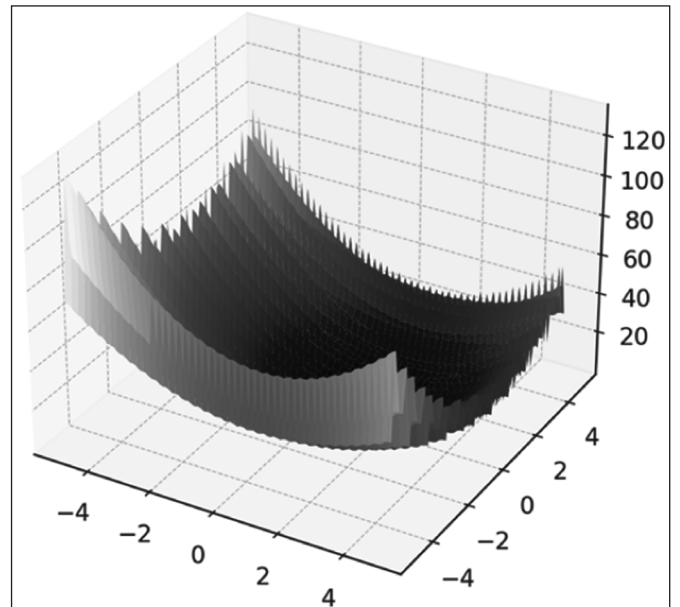


Рис. 3. 3D график поверхности функции Леви

Функция Растригина:

$$f(x, y) = 10n + (x^2 - 10\cos(2\pi x)) + (y^2 - 10\cos(2\pi y))$$

Функция Растригина [30] — невыпуклая функция, предложенная в 1974 году Леонардом Растригиным как функция двух переменных и используемая для тестирования эффективности алгоритмов оптимизации. Эта функция является классическим примером нелинейной мультимодальной функции. Отличительной особенностью является то, что функция обобщена на высшие размерности. Нахождение минимума этой функции принято считать достаточно трудной и нетривиальной задачей из-за большого числа локальных минимумов и широкой области поиска.

Функция Растригина имеет глобальный экстремум в точках:

$$\begin{cases} f(0,0) = 0, n = 2 \\ f(0,0,0) = 0, n = 3 \\ f(0,\dots,0) = 0, n > 3 \end{cases}$$

где n — это размерность пространства.

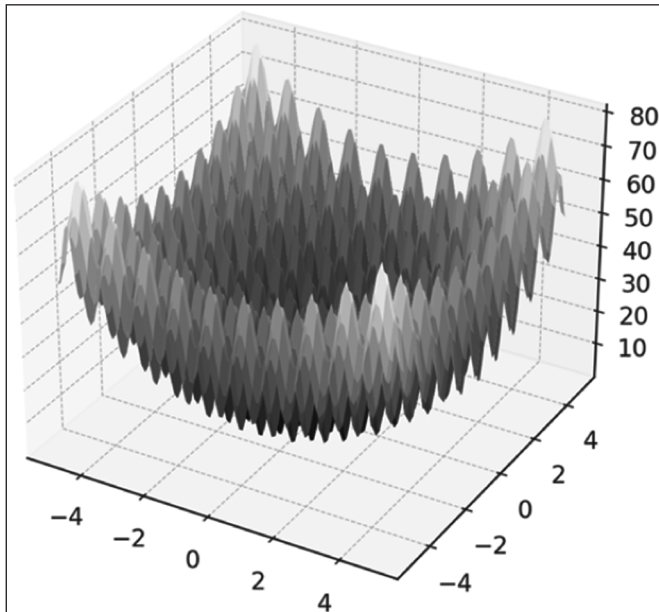


Рис. 4. 3D график поверхности функции Растригина

Эксперименты и результаты

Для функций критерий останова был задан точностью по значению функции. Для функции Розенброка был выбран диапазон $[0, 5]$, для функции Химмельблау $[0, 20]$, для функции Леви $[0, 10]$, а для функции Растригина $[2, 10]$.

В таблицы занесены результаты из десяти экспериментов на функциях с низкой сложностью поверхности, проведенных в соответствии с описанной методологией.

Средние значения результатов и число итераций для каждого из алгоритмов по каждой из оптимизируемых функций занесены в таблицу 5.

Выводы по проделанному исследованию

Из таблицы видно, что проведено сравнение двух типов алгоритмов — параллельного генетического алгоритма и параллельного адаптивного алгоритма — на четырёх разных функциях: функция Розенброка, функция Химмельблау, функция Леви и функция Растригина.

Функция Розенброка:

Параллельный адаптивный алгоритм показал немного лучший средний результат (3,279 против 3,211), но требовал меньше итераций для достижения результата (70,4 против 103,3).

Функция Химмельблау:

Здесь также адаптивный алгоритм показал лучший средний результат (9,773 против 11,902) с меньшим числом итераций (135,1 против 115,8).

Функция Леви:

Адаптивный алгоритм не только достиг лучшего среднего результата (3,996 против 4,538), но и существенно сократил количество итераций (293,5 против 317).

Функция Растригина:

Адаптивный алгоритм дал значительно лучший средний результат (7,117 против 6,91), при этом требуя меньше итераций (277,2 против 299,6).

Общий анализ показывает, что параллельный адаптивный алгоритм систематически превосходит параллельный генетический алгоритм по качеству результатов и эффективности в смысле числа итераций и скорости сходимости. Это может свидетельствовать о более высокой адаптивности и эффективности адаптивных методов в задачах оптимизации на различных типах функций, что делает его предпочтительным выбором в ситуациях, когда необходимо достигнуть высокой точности при минимальных вычислительных затратах.

Заключение

Данные результаты демонстрируют превосходство параллельного адаптивного алгоритма по сравнению с традиционным параллельным генетическим алгоритмом в контексте рассмотренных оптимизационных задач. Это подчеркивает важность разработки и использования адаптивных подходов для улучшения эффективности алгоритмов оптимизации.

Данное исследование может быть использовано для дальнейшего улучшения и адаптации алгоритмов, а также для исследования их поведения на других типах функций или при различных условиях испытаний.

Таблица 1.

Результаты оптимизации функции Розенброка

Номер запуска	Параллельный генетический алгоритм				Параллельный адаптивный алгоритм			
	X	Y	Значение	Число итераций	X	Y	Значение	Число итераций
Запуск 1	1.24	1.37	2.83	87	1.42	2.18	2.64	51
Запуск 2	1.18	1.21	3.36	83	-1.00	1.02	4.02	66
Запуск 3	-0.05	-0.16	3.72	121	0.48	0.37	2.19	99
Запуск 4	1.41	1.82	3.00	82	-0.25	-0.10	4.29	73
Запуск 5	0.07	0.19	4.39	62	-0.25	-0.09	3.95	67
Запуск 6	-0.60	0.36	2.55	96	-0.77	0.61	3.15	54
Запуск 7	0.32	0.25	2.60	104	0.56	0.48	3.07	50
Запуск 8	1.28	1.81	2.65	111	0.70	0.34	2.21	109
Запуск 9	0.48	0.03	4.32	150	-1.04	1.01	4.66	62
Запуск 10	0.11	-0.12	2.69	137	-0.32	0.01	2.61	73

Таблица 2.

Результаты оптимизации функции Химмельблау

Номер запуска	Параллельный генетический алгоритм				Параллельный адаптивный алгоритм			
	X	Y	Значение	Число итераций	X	Y	Значение	Число итераций
Запуск 1	-3.44	-3.44	8.74	130	3.23	-1.40	7.27	124
Запуск 2	-3.18	-3.17	16.37	121	3.36	1.96	5.09	91
Запуск 3	-4.12	-3.04	11.88	69	3.55	2.04	13.98	163
Запуск 4	3.29	-1.43	5.4	122	3.14	-2.18	11.66	139
Запуск 5	2.07	2.29	19.73	77	3.58	-0.71	9.77	59
Запуск 6	3.63	1.23	15.1	162	2.51	2.55	8.62	65
Запуск 7	3.17	0.55	12.56	67	2.92	2.90	18.68	177
Запуск 8	-1.90	3.14	18.81	105	3.51	-1.83	0.27	235
Запуск 9	2.98	1.50	3.49	198	3.38	-0.31	12.4	139
Запуск 10	3.90	-1.62	6.94	107	3.50	-2.53	9.99	159

Таблица 3.

Результаты оптимизации функции Леви

Номер запуска	Параллельный генетический алгоритм				Параллельный адаптивный алгоритм			
	X	Y	Значение	Число итераций	X	Y	Значение	Число итераций
Запуск 1	1,39	0,59	0,71	417	-1,31	0,26	8,86	116
Запуск 2	0,61	1,25	0,64	447	1,24	1,37	0,86	288
Запуск 3	-0,26	1,71	3,22	184	2,18	0,74	3,07	256
Запуск 4	2,79	1,85	8,42	312	1,38	0,83	0,50	467

Номер запуска	Параллельный генетический алгоритм				Параллельный адаптивный алгоритм			
	X	Y	Значение	Число итераций	X	Y	Значение	Число итераций
Запуск 5	1,11	−0,36	3,67	280	2,32	−0,17	5,89	137
Запуск 6	−1,45	0,98	7,09	240	1,56	0,37	1,64	368
Запуск 7	2,82	1,65	5,06	138	1,63	−0,04	1,71	398
Запуск 8	2,64	2,37	5,93	312	0,14	−0,43	4,55	205
Запуск 9	0,59	2,53	3,22	430	−1,11	0,05	7,20	232
Запуск 10	1,97	−0,86	7,42	410	−0,54	1,53	5,68	468

Таблица 4.

Результаты оптимизации функции Растригина

Номер запуска	Параллельный генетический алгоритм				Параллельный адаптивный алгоритм			
	X	Y	Значение	Число итераций	X	Y	Значение	Число итераций
Запуск 1	1,04	2,13	9,04	236	0,96	1,83	9,59	311
Запуск 2	−1,90	0,09	6,98	230	2,96	−0,02	9,12	126
Запуск 3	−0,03	−2,04	4,61	467	0,85	−1,02	5,73	163
Запуск 4	1,12	0,03	4,20	424	0,06	1,14	5,70	337
Запуск 5	3,02	0,03	9,33	363	0,93	−1,03	3,00	307
Запуск 6	0,79	−0,08	9,43	181	−0,02	−0,83	5,92	371
Запуск 7	2,02	2,07	9,47	209	2,02	−2,08	9,58	295
Запуск 8	0,12	0,02	2,61	173	1,10	0,13	6,42	183
Запуск 9	−1,04	1,19	8,89	439	0,95	0,82	7,77	328
Запуск 10	0,03	1,95	4,54	274	−0,20	0,95	8,34	351

Таблица 5.

Сводная таблица средних результатов алгоритмов

	функция Розенброка		функция Химмельблау		функция Леви		функция Растригина	
	Средний результат	Среднее число итераций	Средний результат	Среднее число итераций	Средний результат	Среднее число итераций	Средний результат	Среднее число итераций
Параллельный генетический алгоритм	3,211	103,3	11,902	115,8	4,538	317	6,91	299,6
Параллельный адаптивный алгоритм	3,279	70,4	9,773	135,1	3,996	293,5	7,117	277,2

ЛИТЕРАТУРА

1. Skiena S.S. The data science design manual. — Springer, 2017.
2. Deb K. Multi-objective optimisation using evolutionary algorithms: an introduction //Multi-objective evolutionary optimisation for product design and manufacturing. — London: Springer London, 2011. — C. 3–34.
3. Reeves C.R. Genetic algorithms //Handbook of metaheuristics. — 2010. — C. 109–139.
4. Yang X.S. Nature-inspired optimization algorithms: Challenges and open problems //Journal of Computational Science. — 2020. — T. 46. — C. 101104.
5. Coello C.A. C. Evolutionary multi-objective optimization: a historical view of the field //IEEE computational intelligence magazine. — 2006. — T. 1. — №. 1. — C. 28–36.
6. Knowles J.D., Corne D.W. Approximating the nondominated front using the Pareto archived evolution strategy //Evolutionary computation. — 2000. — T. 8. — №. 2. — C. 149–172.
7. Wang L., Wang T., Luo Y. Improved non-dominated sorting genetic algorithm (NSGA)-II in multi-objective optimization studies of wind turbine blades //Applied Mathematics and Mechanics. — 2011. — T. 32. — №. 6. — C. 739–748.
8. Emmerich M.T.M., Deutz A.H. A tutorial on multiobjective optimization: fundamentals and evolutionary methods //Natural computing. — 2018. — T. 17. — C. 585–609.
9. Leguizamón G., Blum C., Alba E. Evolutionary computation //Handbook of approximation algorithms and metaheuristics. — Chapman and Hall/CRC, 2018. — C. 379–394.
10. Shi Y. et al. Condition-based maintenance optimization for multi-component systems subject to a system reliability requirement //Reliability Engineering & System Safety. — 2020. — T. 202. — C. 107042.
11. Kochergin V. et al. Optimization of technical monitoring processes //Transportation Research Procedia. — 2021. — T. 54. — C. 166–172.
12. Rahimi I. et al. A review on constraint handling techniques for population-based algorithms: from single-objective to multi-objective optimization //Archives of Computational Methods in Engineering. — 2023. — T. 30. — №. 3. — C. 2181–2209.
13. Melhem F.Y. Optimization methods and energy management in « smart grids»: дис. — Université Bourgogne Franche-Comté, 2018.
14. Chekanin V.A., Chekanin A.V. Solving the problem of dense packing of objects of complex geometry //Advances in Mechanical Engineering: Selected Contributions from the Conference «Modern Engineering: Science and Education», Saint Petersburg, Russia, June 2021. — Cham: Springer International Publishing, 2021. — C. 108–116.
15. Talatahari S., Azizi M., Toloo M. Fuzzy adaptive charged system search for global optimization //Applied soft computing. — 2021. — T. 109. — C. 107518.
16. Chekanin V.A., Chekanin A.V. Design of library of metaheuristic algorithms for solving the problems of discrete optimization //Advances in Mechanical Engineering: Selected Contributions from the Conference «Modern Engineering: Science and Education», Saint Petersburg, Russia, June 2017. — Springer International Publishing, 2018. — C. 25–32.
17. Sudholt D. The benefits of population diversity in evolutionary algorithms: a survey of rigorous runtime analyses //Theory of evolutionary computation: Recent developments in discrete optimization. — 2020. — C. 359–404.
18. Lambora A., Gupta K., Chopra K. Genetic algorithm-A literature review //2019 international conference on machine learning, big data, cloud and parallel computing (COMITCon). — IEEE, 2019. — C. 380–384.
19. Manzoni L., Mariot L., Tuba E. Balanced crossover operators in genetic algorithms //Swarm and Evolutionary Computation. — 2020. — T. 54. — C. 100646.
20. Katoch S., Chauhan S. S., Kumar V. A review on genetic algorithm: past, present, and future //Multimedia tools and applications. — 2021. — T. 80. — C. 8091–8126.
21. Shigekawa E. et al. The current state of telehealth evidence: a rapid review //Health Affairs. — 2018. — T. 37. — №. 12. — C. 1975–1982.
22. Rahman M. A. et al. Nature-inspired metaheuristic techniques for combinatorial optimization problems: Overview and recent advances //Mathematics. — 2021. — T. 9. — №. 20. — C. 2633.
23. Han S., Xiao L. An improved adaptive genetic algorithm //SHS web of conferences. — EDP Sciences, 2022. — T. 140. — C. 01044.
24. Lin C. An adaptive genetic algorithm based on population diversity strategy //2009 Third International Conference on Genetic and Evolutionary Computing. — IEEE, 2009. — C. 93–96.
25. Tani L. et al. Evolutionary algorithms for hyperparameter optimization in machine learning for application in high energy physics //The European Physical Journal C. — 2021. — T. 81. — C. 1–9.
26. Karlsen M.R., Moschogiannis S. Evolution of control with learning classifier systems //Applied network science. — 2018. — T. 3. — C. 1–36.
27. Cao T. et al. Function and mechanisms of truncated BDNF receptor TrkB. T1 in neuropathic pain //Cells. — 2020. — T. 9. — №. 5. — C. 1194.
28. Kargar R., Ebadian A., Sokół J. On Booth lemniscate and starlike functions //Analysis and Mathematical Physics. — 2019. — T. 9. — C. 143–154.
29. Sayas F.J., Brown T.S., Hassell M.E. Variational techniques for elliptic partial differential equations: Theoretical tools and advanced applications. — CRC Press, 2019.
30. Omeradzic A., Beyer H.G. Progress rate analysis of evolution strategies on the rastrigin function: First results //International Conference on Parallel Problem Solving from Nature. — Cham: Springer International Publishing, 2022. — C. 499–511.
31. Han W. et al. Global-simplex optimization algorithm applied to FEM-based optimal design of electric machine //IEEE Transactions on Magnetics. — 2017. — T. 53. — №. 6. — C. 1–4.
32. Gajawada S., Mustafa H.M.H. Testing Multiple Strategy Human Optimization based Artificial Human Optimization Algorithms //Computer Reviews Journal. — 2018. — T. 1. — №. 2.

© Черников Александр Владимирович (aleksandrchernikov98@gmail.com)

Журнал «Современная наука: актуальные проблемы теории и практики»