

Методы, технологии и приложения искусственного интеллекта

УДК 519.6

DOI:10.25729/ESI.2026.42.2.003

Модифицированный подход к управлению параметрами генетического алгоритма на основе глубокого обучения с подкреплением

Привалов Константин Сергеевич

Финансовый университет при Правительстве РФ,
Россия, Москва, *qqstik@gmail.com*

Аннотация. Актуальность исследования обусловлена тем, что эффективность классических генетических алгоритмов (ГА) при решении задач глобальной оптимизации существенно зависит от выбора вероятностей кроссовера и мутации, а фиксированные значения параметров нередко приводят к преждевременной сходимости и стабилизации популяции в окрестности локальных экстремумов. Цель работы – разработать и экспериментально оценить подходы к адаптивному управлению параметрами ГА на основе искусственных нейронных сетей и обучения с подкреплением. В рамках единой математической постановки определены признаки состояния популяции, множество действий (дискретные изменения вероятностей мутации и кроссовера p_m и p_c) и функция вознаграждения, отражающая улучшение качества решения между поколениями с учетом штрафа за чрезмерно высокую мутацию. Рассмотрены четыре варианта алгоритма: классический ГА с фиксированными параметрами, гибридный ГА с нейросетевым регулятором (GA+NN), ГА с табличным Q-learning (GA+RL) и предлагаемый метод управления параметрами на основе глубокого Q-обучения, использующий нейронную сеть для аппроксимации Q-функции (GA+DQN). Научная новизна работы состоит в интеграции DQN-агента в контур параметрического управления ГА в рамках формализованной модели «состояние–действие–награда» и в сопоставлении его эффективности с нейросетевым регулятором и табличным Q-обучением на задачах непрерывной оптимизации. Численные эксперименты выполнены на тестовых функциях Растригина и Шаффера при 20 независимых прогонах для каждой конфигурации. В качестве итоговых метрик использовались лучшее значение целевой функции в последнем поколении $f_{min}(T_{max})$ и наилучшее значение, достигнутое за весь прогон. Показано, что GA+RL обеспечивает наибольшее улучшение качества решений. Метод GA+DQN демонстрирует умеренное улучшение относительно базового ГА, подтверждая работоспособность глубокой аппроксимации Q-функции при управлении параметрами. Нейросетевой регулятор в рассмотренной схеме обучения показывает высокую чувствительность к настройкам и в данных экспериментах уступает RL-подходам. Результаты сравнения представлены в виде графиков сходимости, анализа показателей разнообразия популяции и сводной таблицы.

Ключевые слова: генетический алгоритм, адаптивное управление параметрами, обучение с подкреплением, гибридные эволюционные алгоритмы

Цитирование: Привалов К.С. Модифицированный подход к управлению параметрами генетического алгоритма на основе глубокого обучения с подкреплением / К.С. Привалов // Информационные и математические технологии в науке и управлении, 2026. – № 2(42). – С. 29-41. – DOI:10.25729/ESI.2026.42.2.003.

Введение. В различных областях науки и техники задачи оптимизации играют ключевую роль, поскольку позволяют находить наилучшие решения при ограниченных ресурсах. Одним из наиболее распространённых подходов к поиску оптимальных решений являются эволюционные алгоритмы, в частности генетические алгоритмы (ГА), зарекомендовавшие себя способностью решать сложные многопараметрические задачи с множеством локальных экстремумов [1-4]. Однако классический ГА нередко сталкивается с проблемой преждевременной сходимости, когда популяция быстро теряет генетическое разнообразие и концентрируется около локального экстремума. Эта проблема особенно проявляется при росте размерности задачи или на так называемых «обманых» тестовых функциях с большим числом локальных минимумов, таких, как функция Растригина [5]. В результате эффективность классического алгоритма снижается, и он нередко ограничивается поиском лишь субоптимальных (локальных) решений.

Для повышения эффективности эволюционного поиска и предотвращения преждевременной сходимости предложены различные методы адаптивного управления параметрами ГА. Наиболее изучены подходы автоматической настройки вероятностей мутации и кроссовера, а также введение специальных операторов для поддержания разнообразия популяции [6]. В последнее время, включая предыдущие работы автора [7, 8] и ряд других исследований [9-11], перспективным направлением становится использование искусственных нейронных сетей (ИНС) в структуре эволюционного алгоритма. Нейросетевой модуль способен в реальном времени корректировать параметры эволюционного процесса (например, вероятности мутации и кроссовера) на основе текущей статистики популяции, что повышает устойчивость алгоритма к локальным экстремумам и ускоряет его сходимость. Однако нейросетевые подходы требуют тщательного выбора архитектуры сети, предварительного обучения на специализированных данных и чувствительны к подбору гиперпараметров, что ограничивает их универсальность в разных задачах.

Одновременно с этим, успехи методов обучения с подкреплением (Reinforcement Learning, RL) в задачах управления и оптимизации открывают новые возможности для адаптивного регулирования эволюционного поиска [9, 10]. В отличие от статических или даже нейросетевых схем, RL-агент способен не только реагировать на текущее состояние поиска, но и обучаться долгосрочной стратегии взаимодействия с популяцией на основе получаемого вознаграждения за улучшение решения. Такой подход позволяет агенту накапливать «опыт» процесса оптимизации и более гибко адаптировать параметры алгоритма при изменении обстановки в пространстве решений. Благодаря обучению без учителя RL-агент формирует внутреннюю модель динамики поиска, что даёт потенциал для более эффективного выхода из локальных экстремумов и ускоренного достижения глобального экстремума.

В предыдущей работе автора [8] была предложена гибридная схема, в которой табличный алгоритм Q-обучения использовался для адаптивного выбора вероятностей мутации и кроссовера в зависимости от состояния популяции. Эксперименты показали, что такой подход обеспечивает более быстрое и устойчивое приближение к глобальному экстремуму по сравнению с классическим ГА и схемой с нейросетевым регулятором. Вместе с тем табличный Q-алгоритм требует дискретизации пространства состояний и действий, что ограничивает его применимость при большом числе признаков состояния и усложняет расширение модели.

Отметим, что в отечественной научной литературе также активно развиваются подходы к управлению и самонастройке генетических алгоритмов. В частности, предлагаются методы представления состояния популяции и обучения нейросетевых моделей для управления работой ГА в процессе оптимизации [12], а также схемы управления ГА с использованием формализованных моделей (в т.ч. на основе аппарата сетей Петри) [13]. Рассматриваются подходы, анализирующие влияние операторов ГА (кроссовера и мутации) на динамику сходимости и предлагающие правила их применения для повышения устойчивости эволюционного поиска [14]. Отдельный интерес представляют самоконфигурируемые ГА с адаптацией по истории успешности параметров, ориентированные на повышение надежности и воспроизводимости результатов [15], а также адаптивные ГА для прикладных задач, где автоматическая настройка параметров сочетается с оптимизированными схемами рекомбинации [16].

1. Цели и задачи исследования. Целью настоящего исследования является разработка и экспериментальная оценка моделей адаптивного управления параметрами генетического алгоритма на основе методов обучения с подкреплением, а также сравнение их эффективности с классическим генетическим алгоритмом и гибридным подходом с нейросетевым регулятором в задачах многомерной оптимизации. Объектом исследования выступают

эволюционные методы глобальной оптимизации, основанные на генетическом алгоритме. Предметом исследования являются математические модели и алгоритмы адаптивного управления параметрами генетического алгоритма (прежде всего вероятностями кроссовера и мутации) с использованием искусственных нейронных сетей и методов обучения с подкреплением, а также их программная реализация и численная оценка эффективности.

Для достижения поставленной цели в работе решаются следующие задачи:

1. Построить единую формальную модель генетического алгоритма с параметрическим управлением, в которой вероятности кроссовера и мутации рассматриваются как управляемые переменные, зависящие от состояния популяции и хода эволюции.
2. Систематизировать и уточнить ранее предложенные подходы к адаптивному управлению параметрами ГА:
 - Схему с нейросетевым регулятором, использующим статистику популяции в качестве входных признаков.
 - Схему с табличным Q-обучением, в которой выбор управляющих воздействий трактуется как задача обучения с подкреплением.
3. Разработать усовершенствованный вариант гибридного алгоритма на основе глубокого обучения с подкреплением (Deep Q-Network), в котором нейронная сеть аппроксимирует функцию ценности и позволяет отказаться от жёсткой дискретизации пространства состояний популяции.
4. Реализовать программный прототип исследуемых алгоритмов (классический ГА, ГА с нейросетевым регулятором, ГА с Q-learning и ГА с DQN) в виде единого программного комплекса, обеспечивающего воспроизводимое проведение вычислительных экспериментов и сбор статистики.
5. Провести серию вычислительных экспериментов на наборе тестовых задач оптимизации (функции Растригина, Шаффера), сравнив рассматриваемые алгоритмы по критериям:
 - Скорость сходимости.
 - Качество найденных решений.
 - Устойчивость к преждевременной сходимости.
 - Уровень генетического разнообразия популяции.
6. Выполнить анализ полученных результатов, выявить характерные стратегии адаптивного управления параметрами (в том числе на основе тепловых карт и динамики параметров), сформулировать выводы о преимуществах и ограничениях каждого подхода и предложить практические рекомендации по применению разработанных алгоритмов в задачах численной оптимизации.

Научная новизна состоит в разработке и экспериментальной проверке схемы адаптивного управления параметрами генетического алгоритма на основе глубокого обучения с подкреплением. В предлагаемом подходе используется нейронная сеть для аппроксимации функции ценности (Q-функции) и выбора управляющих воздействий на параметры $p_c^{(t)}$, $p_m^{(t)}$ в зависимости от текущего состояния популяции. В отличие от табличного Q-learning, предложенная модификация не требует жёсткой дискретизации пространства состояний и может масштабироваться на более информативные признаки популяции.

2. Методы и используемые модели. Во всех рассматриваемых вариантах используется одна и та же эволюционная основа – генетический алгоритм с действительным кодированием, описанный в [7, 8]. Поэтому в настоящей работе приведём лишь обобщённую схему, а подробное описание операторов селекции, кроссовера и мутации будем считать известными.

2.1 Общая структура гибридного алгоритма. Пусть на шаге эволюции t задана популяция:

$$P^{(t)} = \{x_1^{(t)}, \dots, x_N^{(t)}\}, x_k^{(t)} \in \Omega \in \mathbb{R}^n,$$

а целевая функция $f(x)$ определяется, например, как функция Растригина или Шаффера. Для каждой особи вычисляется значение целевой функции:

$$F_k^{(t)} = f(x_k^{(t)}),$$

и далее выполняются стандартные шаги ГА: селекция, кроссовер, мутация, формирование нового поколения и проверка критерия остановки. Ключевую роль в работе алгоритма играют вероятности кроссовера и мутации:

- $p_c^{(t)}$ – вероятность применения оператора кроссовера (скрещивания) к паре родителей на шаге t .
- $p_m^{(t)}$ – вероятность мутации отдельного гена (компоненты вектора решения) на шаге t .

В классическом ГА эти величины постоянны и задаются до начала работы алгоритма, а во всех гибридных схемах, рассматриваемых в настоящей работе, предполагается наличие модуля управления параметрами U , который по состоянию популяции на шаге t формирует значения $p_c^{(t)}, p_m^{(t)}$:

$$(p_c^{(t)}, p_m^{(t)}) = U(s_t),$$

где s_t – вектор признаков, описывающих текущее состояние популяции (статистики по значениям функции, разнообразие и др.). Конкретная форма оператора U различается для разных подходов:

1. Классический ГА: U тождественно возвращает фиксированные значения p_c, p_m .
2. ГА + ИНС: U реализуется нейронной сетью, обучаемой по мере работы алгоритма.
3. ГА + табличный Q-learning: U задаётся стратегией RL-агента, использующего Q-таблицу.
4. ГА + DQN: U задаётся нейросетью, аппроксимирующей Q-функцию, и обучаемой по алгоритму глубокого Q-обучения.

Таким образом, эволюционная часть алгоритма во всех случаях одинакова; различия касаются только способа формирования управляющих параметров $p_c^{(t)}, p_m^{(t)}$. Это удобно как для теоретического сравнения, так и для программной реализации, поскольку позволяет использовать один и тот же код ГА и переключать только модуль управления.

Детально описывать базовый ГА и нейросетевой регулятор не будем. Основное внимание уделяется новой модификации модуля U на основе Q-обучения [11].

2.2 Модель управления параметрами ГА на основе глубокого Q-обучения. В предлагаемой модификации гибридного алгоритма модуль управления параметрами генетического алгоритма реализуется в виде агента обучения с подкреплением, использующего нейросеть для аппроксимации функции ценности (DQN) [9–11]. Эволюционный процесс ГА рассматривается как среда, с которой взаимодействует RL-агент.

На шаге эволюции t состояние среды описывается вектором статистик текущей популяции:

$$s_t = (\bar{f}^{(t)}, f_{min}^{(t)}, f_{max}^{(t)}, \sigma_f^{(t)}, D^{(t)})^T,$$

где $\bar{f}^{(t)}$ – среднее значение целевой функции по популяции, $f_{min}^{(t)}$ и $f_{max}^{(t)}$ – лучшее и худшее значения, $\sigma_f^{(t)}$ – стандартное отклонение, а $D^{(t)}$ – мера разнообразия популяции (например, среднее попарное расстояние между особями).

Действием агента на шаге t является выбор приращений к вероятностям кроссовера и мутации для следующего поколения. Рассматривается конечное множество действий:

$$A = \{(\Delta p_c, \Delta p_m)\},$$

где Δp_c и Δp_m принимают значения из ограниченного набора. Новые значения параметров вычисляются по формулам:

$$\begin{aligned} p_c^{(t+1)} &= \text{clip}(p_c^{(t)} + \Delta p_c, p_c^{\min}, p_c^{\max}), \\ p_m^{(t+1)} &= \text{clip}(p_m^{(t)} + \Delta p_m, p_m^{\min}, p_m^{\max}), \end{aligned}$$

где функция *clip* ограничивает значение на заданном отрезке. Тем самым, как и в табличном Q-алгоритме [8], агент дискретно изменяет параметры ГА, но оценка качества действий выполняется нейросетью.

После формирования нового поколения $P^{(t+1)}$ вычисляется награда, отражающая изменение качества популяции:

$$r_{t+1} = \bar{f}^{(t)} - \bar{f}^{(t+1)}.$$

Положительное значение r_{t+1} соответствует уменьшению среднего значения целевой функции (улучшение), отрицательное – ухудшению или стагнации.

Для аппроксимации функции ценности $Q(s, a)$ используется полносвязная нейронная сеть с параметрами θ . Сеть принимает на вход вектор состояния s_t и возвращает оценки значений $Q(s_t, a_j)$ для всех действий $a_j \in A$:

$$Q(s_t, a_j; \theta), \quad j = 1, \dots, |A|.$$

Выбор действия выполняется по ε – жадной стратегии. с вероятностью ε_t выбирается случайное действие (исследование), с вероятностью $1 - \varepsilon_t$ выбирается действие с максимальной оценкой $Q(s_t, a; \theta)$.

В процессе работы алгоритма в буфер опыта накапливаются переходы вида:

$$(s_t, a_t, r_{t+1}, s_{t+1}).$$

Через регулярные интервалы из буфера выбирается случайный мини-батч переходов, и для каждого перехода вычисляется целевое значение:

$$y_t = r_{t+1} + \gamma \max_{a' \in A} Q(s_{t+1}, a'; \theta^-),$$

где γ – коэффициент дисконтирования, а θ^- – параметры целевой сети, обновляемой реже, чем основная сеть. Параметры θ основной сети находятся путём минимизации средней квадратичной ошибки:

$$\mathcal{L}(\theta) = \frac{1}{B} \sum_{t \in \text{batch}} (y_t - Q(s_t, a_t; \theta))^2,$$

где B – размер мини-батча. Обновление параметров θ выполняется методом стохастического градиентного спуска.

Таким образом, на каждом шаге t гибридный алгоритм «ГА + DQN» работает по следующей схеме:

1. По текущей популяции $P^{(t)}$ вычисляется состояние s_t .
2. DQN-агент выбирает действие a_t и, следовательно, значения $p_c^{(t)}$ и $p_m^{(t)}$.
3. С выбранными параметрами выполняются операторы селекции, кроссовера и мутации, формируются особи нового поколения $P^{(t+1)}$.
4. Вычисляется награда r_{t+1} и новое состояние s_{t+1} .
5. Переход $(s_t, a_t, r_{t+1}, s_{t+1})$ добавляется в буфер опыта, выполняется шаг обучения сети.

В итоге, DQN-агент по ходу эволюции ГА самообучается стратегии управления параметрами, стремясь максимизировать суммарное улучшение качества популяции. В отличие от табличного Q-обучения, нейросетевая аппроксимация Q-функции позволяет работать с непрерывным пространством признаков состояния и обобщать накопленный опыт, что особенно важно при использовании более богатого набора статистик популяции и при росте размерности задач [9–11].

3. Постановка эксперимента. Для экспериментальной оценки предложенного алгоритма и базовых схем были проведены численные эксперименты на наборе стандартных тестовых функций глобальной оптимизации. В настоящей работе подробно рассматриваются две функции:

1. Функция Растригина размерности $n = 10$ с областью допустимых значений $x_i \in [-5.12, 5.12]$.
2. Функция Шаффера (модификация N.2) в двумерном пространстве с областью $x, y \in [-100, 100]$.

В расчётном блоке реализованы четыре варианта алгоритма:

1. GA – классический генетический алгоритм с фиксированными параметрами p_c, p_m .
2. GA+NN – гибридный ГА с нейросетевым регулятором параметров, обучаемый по разности средних значений целевой функции между поколениями.
3. GA+RL – гибридный ГА с табличным Q-learning, управляющим дискретными изменениями p_c, p_m .
4. GA+DQN – предлагаемый в работе гибридный алгоритм, в котором управление параметрами выполняет агент глубокого Q-обучения (DQN) с нейросетевой аппроксимацией Q-функции.

Во всех вариантах использовалась единая эволюционная схема, реализованная с помощью библиотеки DEAP:

- Тип кодирования: действительный вектор длины n .
- Способ инициализации: равномерное распределение на отрезке $[x_i^{min}, x_i^{max}]$.
- Размер популяции: $N = 50$ для обеих задач.
- Оператор селекции: турнирный отбор с размером турнира 3.
- Оператор кроссовера: двухточечный кроссовер.
- Оператор мутации: гауссовская мутация с нулевым средним и фиксированным стандартным отклонением, с последующей проекцией решения в допустимую область.
- Элитизм: сохранение лучшей особи из поколения в поколение.

Для классического ГА начальные значения параметров выбирались, как:

$$p_c = 0.7, \quad p_m = 0.2,$$

и далее оставались неизменными. Во всех адаптивных вариантах алгоритма эти значения использовались в качестве стартовых, после чего динамически корректировались соответствующим управляющим модулем (нейросетью, Q-агентом или DQN-агентом). Число поколений в одном прогоне составляло $T_{max} = 50$. Для оценки устойчивости алгоритмов каждая конфигурация (комбинация функции и варианта ГА) запускалась на 20 независимых прогонах при фиксированном лимите поколений. Во всех сериях экспериментов использовались одинаковые генераторы случайных чисел (NumPy/DEAP) и фиксированное начальное состояние генератора для воспроизводимости эксперимента.

В гибридном алгоритме GA+NN управляющий модуль реализовывался полносвязной нейронной сетью с одним скрытым слоем из 16 нейронов и функцией активации ReLU. На вход сети подавался вектор статистик популяции s_t . Выход сети интерпретировался, как желательные значения вероятностей мутации и кроссовера, ограниченные в допустимом диапазоне. Обучение проводилось методом стохастического градиентного спуска (оптимизатор Adam, скорость обучения 10^{-3}) по квадратичной функции потерь между фактическим выходом сети и «целевыми» значениями параметров, вычисляемыми из улучшения среднего значения функции между поколениями.

В предлагаемом алгоритме GA+DQN Q-функция аппроксимировалась нейронной сетью с двумя скрытыми слоями по 32 нейрона и активацией ReLU. На вход сети подавался

расширенный вектор состояния, включающий статистику популяции и текущие значения p_c и p_m . На выходе сеть возвращала оценки $Q(s_t, a_j)$ для всех действий a_j из того же дискретного набора, что и в табличном Q-learning. Обучение сети осуществлялось по методу Deep Q-Network с буфером повторов объёмом 10^4 переходов, размером мини-батча 32, коэффициентом дисконтирования $\gamma = 0.99$, скоростью обучения 10^{-3} и периодическим обновлением целевой сети (каждые 50 шагов обучения). Выбор действий осуществлялся по ϵ – жадной стратегии.

Для каждого прогона фиксировались следующие величины по поколениям:

- Лучшее значение целевой функции в популяции $f_{min}^{(t)}$.
- Среднее значение $\bar{f}^{(t)}$.
- Мера разнообразия популяции $D^{(t)}$.
- Фактические значения параметров мутации и кроссовера $p_m^{(t)}, p_c^{(t)}$.

На основе этих данных строились усреднённые кривые сходимости по лучшему значению функции, графики изменения вероятности мутации, а также сводные таблицы с числовыми характеристиками (среднее и стандартное отклонение финального лучшего значения по 20 прогонам).

3.1 Сходимость и качество решения. На рис. 1 и рис. 2 представлены усреднённые по 20 независимым прогонам кривые сходимости по лучшему значению функции в поколении для функций Растригина и Шаффера соответственно. Для каждой задачи сравниваются четыре варианта алгоритма: классический ГА, гибрид с нейросетевым регулятором (GA+NN), гибрид с табличным Q-learning (GA+RL) и предлагаемый алгоритм GA+DQN.

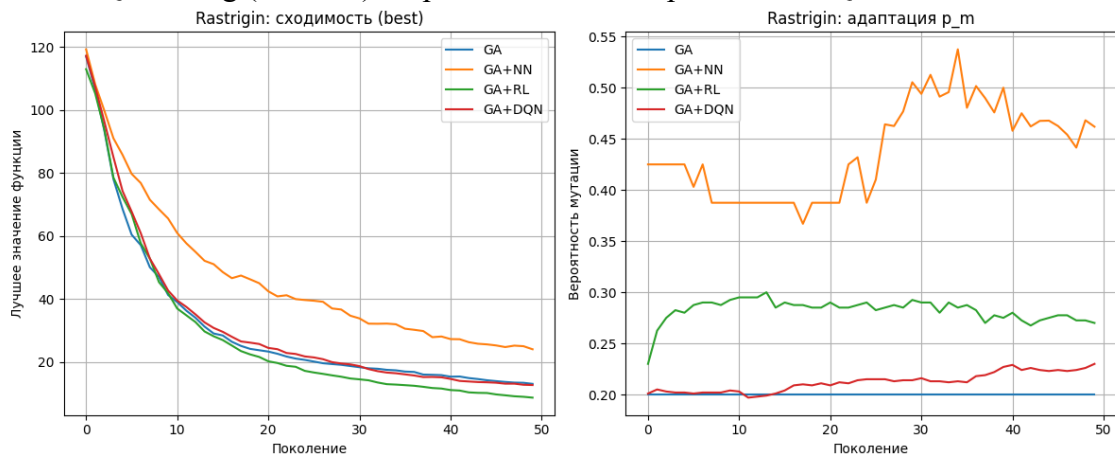


Рис. 1. Сходимость по лучшему значению и динамика вероятности мутации для функции Растригина

Из рис. 1 видно, что классический ГА демонстрирует типичное поведение: в первые 10–15 поколений наблюдается существенное снижение $f_{min}^{(t)}$, после чего сходимость замедляется, и к 50-му поколению среднее лучшее значение составляет порядка 13. Гибрид с нейросетевым регулятором в данной постановке задач уступает базовому алгоритму: его кривая сходимости на всём интервале поколений проходит выше, а итоговое среднее значение $f_{min}^{(50)}$ находится около 24. Наилучший результат показывает гибрид GA+RL: за счёт более агрессивной адаптации параметров на ранних этапах поиска он быстрее снижает $f_{min}^{(t)}$ и к 50-му поколению достигает в среднем значения около 8.7, что заметно лучше классического ГА. Предлагаемый алгоритм GA+DQN на функции Растригина демонстрирует поведение, близкое к базовому методу: его кривая чуть смещена вниз, а среднее финальное значение порядка 12.7 лишь

немного превосходит результат классического ГА, но существенно лучше, чем у варианта GA+NN.

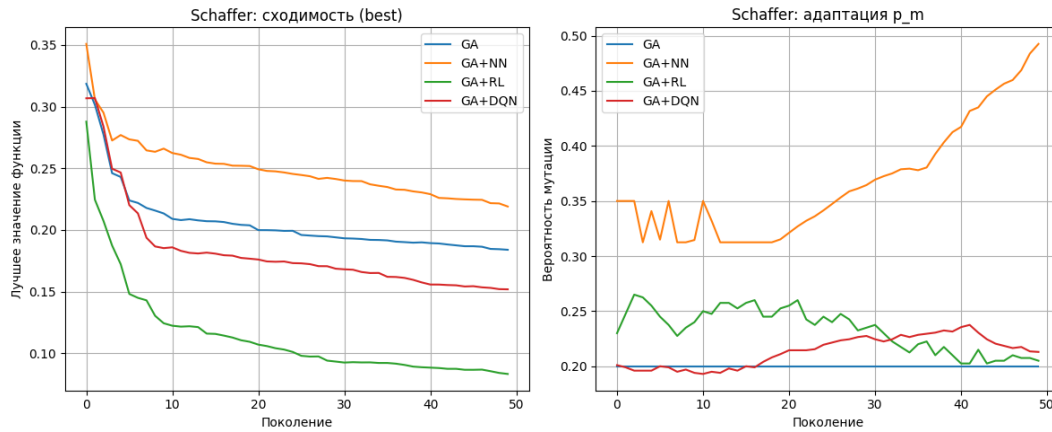


Рис. 2. Сходимость по лучшему значению и динамика вероятности мутации для функции Шаффера

Для двумерной функции Шаффера (рис. 2) классический ГА в среднем выходит на значения $f_{min}^{(50)}$ порядка 0.18, причём сходимость после первых 10 поколений становится более полой. Нейросетевой гибридом GA+NN также не даёт выигрыша и даже несколько ухудшает результаты: его кривая сходимости на всём протяжении остаётся выше, а финальное среднее значение достигает около 0.22. Наиболее быстрое и глубокое снижение $f_{min}^{(t)}$ обеспечивает метод GA+RL: уже после 10–15 поколений он достигает уровня порядка 0.1, а к 50-му поколению среднее лучшее значение становится около 0.08, что примерно в два раза лучше результата классического ГА. Предлагаемый алгоритм GA+DQN на этой задаче занимает промежуточное положение: кривая сходимости проходит ниже классического ГА и выше GA+RL, финальное среднее значение $f_{min}^{(50)}$ составляет около 0.15.

В целом, представленные графики показывают, что применение методов обучения с подкреплением (Q-learning и DQN) позволяет улучшить качество решений и ускорить сходимость генетического алгоритма на рассматриваемых тестовых функциях. При этом простая схема нейросетевого регулирования параметров без явно заданной функции награды оказывается менее эффективной и в ряде случаев уступает базовому варианту ГА.

3.2 Сводные количественные показатели. Для количественной оценки качества работы алгоритмов были использованы две метрики:

1. Среднее значение лучшей найденной особи в последнем поколении $f_{min}^{(Tmax)}$ по 20 прогонам (со стандартным отклонением).
2. Среднее по прогонам наименьшее значение целевой функции, достигнутое за всё время работы алгоритма (минимум по поколениям), также со стандартным отклонением.

Сводные результаты для функций Растригина (10 измерений) и Шаффера (2 измерения) приведены в таблице 1.

На функции Растригина гибридом GA+RL демонстрирует наилучшие показатели: среднее финальное значение $f_{min}^{(Tmax)}$ составляет 8.67 при стандартном отклонении 3.12, тогда как классический ГА достигает 13.04 ± 5.23 . Предлагаемый алгоритм GA+DQN по финальному значению (12.67 ± 3.99) сопоставим с базовым методом и немного улучшает его результат, тогда как вариант с нейросетевым регулятором (GA+NN) заметно уступает остальным (24.02 ± 8.19). Аналогичная картина наблюдается и по метрике «лучшее значение за прогон»: 8.49 ± 3.02 для GA+RL против 13.00 ± 5.26 для классического ГА и 12.41 ± 3.93 для GA+DQN.

Таблица 1. Результаты экспериментов на функциях Растригина и Шаффера

Функция	Алгоритм	$f_{min}^{(Tmax)}$ (среднее $\pm \sigma$)	f_{min}^{best} (среднее $\pm \sigma$)
Растринг	GA	13.043 \pm 5.230	12.996 \pm 5.255
Растринг	GA+NN	24.016 \pm 8.189	21.998 \pm 8.421
Растринг	GA+RL	8.670 \pm 3.123	8.489 \pm 3.019
Растринг	GA+DQN	12.665 \pm 3.995	12.413 \pm 3.925
Шаффер	GA	0.184 \pm 0.137	0.157 \pm 0.120
Шаффер	GA+NN	0.219 \pm 0.128	0.192 \pm 0.125
Шаффер	GA+RL	0.083 \pm 0.084	0.078 \pm 0.076
Шаффер	GA+DQN	0.152 \pm 0.125	0.141 \pm 0.118

На функции Шаффера лучшим по обоим критериям также оказывается метод GA+RL: среднее финальное значение составляет 0.083 ± 0.084 , тогда как классический ГА даёт 0.184 ± 0.137 , а GA+DQN — 0.152 ± 0.125 . Вариант GA+NN снова показывает наихудший результат (0.219 ± 0.128). По метрике минимального значения за прогон GA+RL достигает в среднем 0.078 ± 0.076 , что существенно лучше как классического ГА (0.157 ± 0.120), так и алгоритма GA+DQN (0.141 ± 0.118).

Таким образом, сводные численные показатели подтверждают выводы, сделанные по графикам сходимости: табличный Q-learning обеспечивает наибольший выигрыш по качеству найденных решений, особенно на многомодальных задачах, тогда как DQN даёт устойчивые, сопоставимые с классическим ГА результаты и выступает как более универсальный, но менее «острый» регулятор. Нейросетевой регулятор в рассмотренной схеме обучения показал чувствительность к настройкам и в текущей конфигурации уступает как базовому методу, так и алгоритмам с обучением с подкреплением.

3.2 Выводы по результатам экспериментов. Проведённые вычислительные эксперименты позволяют сделать следующие выводы:

1. Использование обучения с подкреплением для управления параметрами ГА приводит к значительному выигрышу по качеству решений. На обеих тестовых функциях гибридный GA+RL обеспечивает существенное снижение финального значения $f_{min}^{(Tmax)}$ по сравнению с классическим ГА. На функции Растригина улучшение составляет порядка 30–35 %, на функции Шаффера – примерно двукратное уменьшение средних значений.
2. Глубокий вариант Q-обучения (GA+DQN) демонстрирует устойчивое поведение и умеренный выигрыш. На функции Растригина DQN незначительно улучшает классический ГА, а на функции Шаффера даёт результат лучше базового алгоритма, но хуже табличного Q-learning. Это показывает, что при ограниченном числе поколений и сравнительно небольшом объёме данных более простая табличная схема может оказаться эффективнее сложной нейросетевой аппроксимации.
3. Нейросетевой регулятор параметров в простой схеме обучения не гарантирует улучшения. В рассмотренной конфигурации метод GA+NN на обеих функциях уступает даже классическому ГА. Это подчёркивает, что применение ИНС для адаптивной настройки параметров требует тщательного выбора архитектуры, нормировки признаков и стратегии обучения, в противном случае эффект от такого усложнения алгоритма может быть отрицательным.
4. На правых графиках рис. 1 и 2 видно, что GA+RL активно изменяет p_m , повышая её на начальных этапах поиска и постепенно снижая по мере приближения к минимуму, тогда как DQN предпочитает более плавные изменения, а классический ГА работает в фиксированном режиме. Нейросетевая схема показывает менее интерпретируемую и более вариативную траекторию p_m , что коррелирует с её нестабильной эффективностью.

Заключение. В работе выполнено сравнение четырех схем управления параметрами генетического алгоритма: классического ГА с фиксированными p_m и p_c , ГА с нейросетевым регулятором (GA+NN), ГА с табличным Q-learning (GA+RL) и предложенного подхода GA+DQN, использующего нейронную аппроксимацию Q-функции. Сформулирована единая математическая модель в терминах «состояние–действие–награда» и реализован программный комплекс для воспроизводимых численных экспериментов. По результатам 20 независимых прогонов показано, что применение обучения с подкреплением повышает качество решений и устойчивость поиска: для функции Растригина среднее значение $f_{min}(T_{max})$ уменьшилось с 13.04 ± 5.23 (ГА) до 8.67 ± 3.12 (GA+RL), а для функции Шаффера — с 0.184 ± 0.137 до 0.083 ± 0.084 , что соответствует улучшению порядка 33% и 55% соответственно. Подход GA+DQN демонстрирует сопоставимые с базовым ГА результаты (12.67 ± 3.99 и 0.152 ± 0.125) и подтверждает применимость глубокого обучения с подкреплением к задаче параметрического управления эволюционным поиском. Нейросетевой регулятор в текущей схеме обучения оказался чувствителен к настройкам и уступил RL-методам. Полученные выводы подтверждают достижение цели исследования и могут быть использованы при разработке прикладных систем глобальной оптимизации и адаптивной настройки метаэвристик.

Практическая значимость результатов заключается в возможности применения предложенных регуляторов параметров p_m и p_c при разработке прикладных систем глобальной оптимизации и модулей автоматической настройки метаэвристик, в том числе для задач с риском преждевременной сходимости. Материалы статьи могут быть интересны специалистам по численным методам оптимизации, эволюционным вычислениям, а также исследователям, использующим нейросетевые и RL-подходы для управления процессом поиска.

Список источников

1. Goldberg D.E. Genetic Algorithms in Search, Optimization, and Machine Learning. Reading, MA, Addison-Wesley, 1989, 432 p.
2. Holland J.H. Adaptation in Natural and Artificial Systems. Ann Arbor, MI, University of Michigan Press, 1975, 183 p.
3. Fogel D.B. Evolutionary Computation: Toward a New Philosophy of Machine Intelligence. 3rd ed. Hoboken, NJ, IEEE Press, 2006, 296 p.
4. Wolpert D.H., Macready W.G. No Free Lunch Theorems for Optimization. IEEE Transactions on Evolutionary Computation, 1997, vol. 1, no. 1, pp. 67–82.
5. Растринг Л.А. О сходимости случайного поиска при экстремальном управлении многопараметрическими системами / Л.А. Растринг // Автоматика и телемеханика, 1963. – Т. 24. – С. 1337–1342.
6. Zhang J., Liao S., Lee S. The Use of Neural Networks for Adaptive Optimization Problems in Large Dimensions. Applied Soft Computing, 2020, vol. 90, art. 106187.
7. Привалов К.С. Гибридные методы оптимизации: адаптивное управление эволюционным процессом с использованием искусственных нейронных сетей / К.С. Привалов // Инженерный вестник Дона, 2025. – № 3. – URL: <http://www.ivdon.ru/ru/magazine/archive/n3y2025/9910> (дата обращения: 06.12.2025).
8. Привалов К.С. Обучение с подкреплением в адаптивном управлении параметрами генетического алгоритма / К.С. Привалов // Инженерный вестник Дона, 2025. – № 8. – URL: <http://www.ivdon.ru/ru/magazine/archive/n8y2025/10315> (дата обращения: 06.12.2025).
9. Sutton R.S., Barto A.G. Reinforcement Learning: An Introduction. 2nd ed. Cambridge, MA, The MIT Press, 2018, 552 p.
10. Mnih V., Kavukcuoglu K., Silver D., et al. Human-level Control through Deep Reinforcement Learning. Nature, 2015, vol. 518, no. 7540, pp. 529–533.
11. Sharma M., Komninos A., López-Ibáñez M., et al. Deep Reinforcement Learning Based Parameter Control in Differential Evolution. GECCO '19: Proceedings of the Genetic and Evolutionary Computation Conference. Prague, ACM, 2019, pp. 709–717.
12. Петросов Д.А. Представление данных о состоянии популяции и обучение искусственной нейронной сети в задаче управления работой генетическим алгоритмом / Д.А. Петросов, Н.А. Андриянов, А.Н. Алуюнов,

- Е.В. Нежданов // Экономика. Информатика, 2023. – Т. 50. – № 4. – С. 924–935. – DOI: 10.52575/2712-746X-2023-50-4-924-935. – URL: <https://econom-inform-journal.ru/index.php/journal/article/view/330> (дата обращения: 21.01.2026).
13. Петросов Д.А. Модель искусственной нейронной сети для решения задачи управления генетическим алгоритмом с применением математического аппарата теории сетей Петри / Д.А. Петросов, А.Н. Зеленина // Моделирование, оптимизация и информационные технологии, 2020. – Т. 8. – № 4. – DOI: 10.26102/2310-6018/2020.31.4.031. – URL: <https://moitvvt.ru/ru/journal/pdf?id=877> (дата обращения: 21.01.2026).
14. Петросов Д.А. Правила применения разрушающей способности операторов генетического алгоритма в задаче структурно-параметрического синтеза имитационных моделей бизнес-процессов / Д.А. Петросов, Н.Ю. Сурова, А.В. Поляков // Моделирование, оптимизация и информационные технологии, 2023. – Т. 11. – № 4. – DOI: 10.26102/2310-6018/2023.43.4.013. – URL: <https://moitvvt.ru/ru/journal/pdf?id=1463> (дата обращения: 21.01.2026).
15. Шерстнев П.А. SelfCSHAGA: самоконфигурируемый генетический алгоритм оптимизации с адаптацией на основе истории успеха / П.А. Шерстнев, Е.С. Семенкин // Вестник МГТУ им. Н.Э. Баумана. Серия «Приборостроение», 2025. – № 2 (151). – С. 122–139. – EDN: TSKBOX. – URL: <https://vestnikprib.bmstu.ru/catalog/icec/sysan/1363.html> (дата обращения: 21.01.2026).
16. Сахно М.Ю. Адаптивный генетический алгоритм с оптимальной рекомбинацией для задачи составления расписаний с учетом расхода энергии / М.Ю. Сахно // Сибирский журнал вычислительной математики. – 2025. – Т. 28, № 3. – С. 327–346. – DOI: 10.15372/SJNM20250307. – URL: <https://www.mathnet.ru/sjvm912> (дата обращения: 21.01.2026).

Привалов Константин Сергеевич. Аспирант кафедры информационных технологий, Финансовый университет при Правительстве РФ. AuthorID: 1321657, SPIN: 5501-7941, ORCID: 0009-0001-5843-2944. qqstik@gmail.com. 125167, Москва, пр-кт Ленинградский, д. 49/2.

UDC 519.6

DOI:10.25729/ESI.2026.42.2.003

A modified approach to controlling the parameters of a genetic algorithm based on deep reinforcement learning

Konstantin S. Privalov

Financial University under the Government of the Russian Federation,
Russia, Moscow, qqstik@gmail.com

Abstract. The relevance of this study is determined by the fact that the efficiency of classical genetic algorithms (GAs) in solving global optimization problems significantly depends on the choice of crossover and mutation probabilities, while fixed parameter values often lead to premature convergence and stabilization of the population in the vicinity of local extrema. The aim of this work is to develop and experimentally evaluate approaches to adaptive control of GA parameters based on artificial neural networks and reinforcement learning. Within a unified mathematical formulation, population state features, a set of actions represented by discrete changes in mutation and crossover probabilities p_m and p_c , and a reward function reflecting the improvement in solution quality between generations with a penalty for excessively high mutation are defined. Four algorithmic variants are considered: a classical GA with fixed parameters, a hybrid GA with a neural-network controller (GA+NN), a GA with tabular Q-learning (GA+RL), and the proposed parameter control method based on deep Q-learning, which uses a neural network to approximate the Q-function (GA+DQN). The scientific novelty of the study lies in the integration of a DQN agent into the GA parameter control loop within a formalized “state–action–reward” model and in comparing its efficiency with that of a neural-network controller and tabular Q-learning on continuous optimization problems. Numerical experiments were carried out on the Rastrigin and Schaffer test functions with 20 independent runs for each configuration. The final metrics used were the best objective function value in the

last generation $f_{min}(T_{max})$ and the best value achieved over the entire run. It is shown that GA+RL provides the greatest improvement in solution quality. The GA+DQN method demonstrates a moderate improvement over the baseline GA, confirming the applicability of deep Q-function approximation to parameter control. The neural-network controller in the considered training scheme shows high sensitivity to parameter settings and, in these experiments, performs worse than the reinforcement learning approaches. The comparison results are presented in the form of convergence plots, an analysis of population diversity indicators, and a summary table.

Keywords: genetic algorithm, adaptive parameter control, reinforcement learning, hybrid evolutionary algorithms

References

1. Goldberg D.E. Genetic Algorithms in Search, Optimization, and Machine Learning. Reading, MA, Addison-Wesley, 1989, 432 p.
2. Holland J.H. Adaptation in Natural and Artificial Systems. Ann Arbor, MI, University of Michigan Press, 1975, 183 p.
3. Fogel D.B. Evolutionary Computation: Toward a New Philosophy of Machine Intelligence. 3rd ed. Hoboken, NJ, IEEE Press, 2006, 296 p.
4. Wolpert D.H., Macready W.G. No Free Lunch Theorems for Optimization. IEEE Transactions on Evolutionary Computation, 1997, vol. 1, no. 1, pp. 67–82.
5. Rastrigin L. A. O skhodimosti sluchajnoogo poiska pri ehkstreml'nom upravlenii mnogoparametricheskimi sistemami [On the convergence of random search in extremal control of multi-parameter systems], Avtomatika i telemekhanika [Automation and remote control], 1963, vol. 24, pp. 1337–1342.
6. Zhang J., Liao S., Lee S. The use of neural networks for adaptive optimization problems in large dimensions, Applied Soft Computing, 2020, vol. 90, art. 106187.
7. Privalov K. S. Gibridnye metody optimizatsii: adaptivnoe upravlenie evolyutsionnym protsessom s ispol'zovaniem iskusstvennykh nejronnykh setey [Hybrid optimization methods: adaptive control of the evolutionary process using artificial neural networks], Inzhenernyj vestnik Dona [Engineering bulletin of the Don], 2025, no. 3, available at: <http://www.ivdon.ru/ru/magazine/archive/n3y2025/9910> (accessed: 12/06/2025).
8. Privalov K. S. Obucheniye s podkrepleniem v adaptivnom upravlenii parametrami geneticheskogo algoritma [Reinforcement learning in adaptive control of genetic algorithm parameters], Inzhenernyj vestnik Dona [Engineering Bulletin of the Don], 2025, no. 8, available at: <http://www.ivdon.ru/ru/magazine/archive/n8y2025/10315> (accessed: 12/06/2025).
9. Sutton R. S., Barto A. G. Reinforcement learning: an introduction, 2nd ed., Cambridge, MA, The MIT Press, 2018, 552 p.
10. Mnih V., Kavukcuoglu K., Silver D., et al. Human-level control through deep reinforcement learning, Nature, 2015, vol. 518, no. 7540, pp. 529–533.
11. Sharma M., Komninos A., López-Ibáñez M., et al. Deep reinforcement learning based parameter control in differential evolution, GECCO '19: Proceedings of the Genetic and Evolutionary Computation Conference, Prague, ACM, 2019, pp. 709–717.
12. Petrosov D.A., Andriyanov N.A., Alyunov A.N., Nezhdanov E.V. Predstavleniye dannykh o sostoyanii populyatsii i obucheniye iskusstvennoy neyronnoy seti v zadache upravleniya rabotoy geneticheskim algoritmom [Population state data representation and training of an artificial neural network for controlling a genetic algorithm]. Ekonomika. Informatika [Economics. Computer science], 2023, vol. 50, no. 4, pp. 924–935, DOI: 10.52575/2712-746X-2023-50-4-924-935, available at: <https://econom-inform-journal.ru/index.php/journal/article/view/330> (accessed: 01/21/2026).
13. Petrosov D.A., Zelenina A.N. Model' iskusstvennoy neyronnoy seti dlya resheniya zadachi upravleniya geneticheskim algoritmom s primeneniyyem matematicheskogo apparata teorii setey Petri [Artificial neural network model for solving the problem of genetic algorithm control using the mathematical apparatus of Petri net theory]. Modelirovaniye, optimizatsiya i informatsionnyye tekhnologii [Modeling, optimization and information technology], 2020, vol. 8, no. 4, DOI: 10.26102/2310-6018/2020.31.4.031, available at: <https://moitvvt.ru/ru/journal/pdf?id=877> (accessed: 01/21/2026).
14. Petrosov D.A., Surova N.Yu., Polyakov A.V. Pravila primeneniya razrushayushchey sposobnosti operatorov geneticheskogo algoritma v zadache strukturno-parametricheskogo sinteza imitatsionnykh modeley biznes-protsessov [Rules for applying the destructive ability of genetic algorithm operators in the problem of structural-parametric synthesis of business process simulation models]. Modelirovaniye, optimizatsiya i informatsionnyye tekhnologii [Modeling, optimization and information technology], 2023, vol. 11, no. 4, DOI: 10.26102/2310-6018/2023.43.4.013, available at: <https://moitvvt.ru/ru/journal/pdf?id=1463> (accessed: 01/21/2026).
15. Sherstnev P.A., Semenkin E.S. SelfCSHAGA: samokonfiguriruyemyy geneticheskiy algoritm optimizatsii s adaptatsiyey na osnove istorii uspekha [SelfCSHAGA: self-configuring genetic optimization algorithm with

adaptation based on success history]. Vestnik MGTU im. N.E. Baumana. Seriya Priborostroyeniye [Bauman MSTU Bulletin. Series "instrument engineering"], 2025, no. 2 (151), pp. 122–139, EDN: TSKBOX. Available at: <https://vestnikprib.bmstu.ru/catalog/icec/sysan/1363.html> (accessed: 01/21/2026).

16. Sakhno M.Yu. Adaptivnyy geneticheskiy algoritm s optimal'noy rekombinatsiyey dlya zadachi sostavleniya raspisaniy s uchetom raskhoda energii [Adaptive genetic algorithm with optimal recombination for scheduling problems with energy consumption]. Sibirskiy zhurnal vychislitel'noy matematiki [Siberian journal of computational mathematics], 2025, vol. 28, no. 3, pp. 327–346, DOI: 10.15372/SJNM20250307, available at: <https://www.mathnet.ru/sjvm912> (accessed: 01/21/2026).

Privalov Konstantin Sergeevich. *Postgraduate student of information technology department, Financial University under the Government of the Russian Federation. AuthorID: 1321657, SPIN: 5501-7941, ORCID: 0009-0001-5843-2944. qqstik@gmail.com. 125167, Moscow, Leningradsky Avenue 49/2.*

Статья поступила в редакцию 15.12.2025; одобрена после рецензирования 22.01.2026; принята к публикации 25.05.26.

The article was submitted 12/15/2025; approved after reviewing 01/22/2026; accepted for publication 05/25/2026.