

УДК 004.8

УПРАВЛЕНИЕ СЛУЧАЙНЫМ ПОИСКОМ В FIREFLY ALGORITHM

Т. В. МРОЧЕК

Белорусско-Российский университет
Могилев, Беларусь

Firefly Algorithm (алгоритм светлячков) – метод роевого интеллекта, широко используемый для решения задач многомерной оптимизации [1]. Его преимуществом является то, что если в задаче есть несколько хороших решений, то агенты («светлячки») не соберутся в одной точке (т. е. не попадут все в один, скорее всего, локальный экстремум), а разделятся на несколько групп вокруг каждого хорошего решения, что дает возможность выбора наилучшего.

Баланс между скоростью сходимости и риском застревания в локальных экстремумах в Firefly Algorithm позволяет регулировать параметр рандомизации α , реализующий механизм случайного поиска и определяющий степень случайного перемещения агента для предотвращения его застревания в локальных экстремумах.

Целью данной работы является изучение возможностей управления значениями параметра рандомизации α в ходе работы алгоритма.

При поиске оптимума привлекательность β агента роя для других агентов зависит от его яркости I , при этом движение агента представляет собой направленный дрейф в сторону более яркого соседа. На расстоянии r реальная привлекательность β из-за поглощения света средой, задаваемого коэффициентом γ , начинает уменьшаться относительно начального β_0 по закону $\beta_0 / (1 + \gamma r^2)$ [2]. Если агент «видит» несколько более ярких особей, его результирующее перемещение становится суперпозицией притяжений ко всем ним с учетом случайного отклонения, задаваемого α , для расширения области поиска и предотвращения застревания в локальных экстремумах.

Известен ряд способов управления значениями параметра рандомизации α .

В классической формулировке Firefly Algorithm α – фиксированная константа в диапазоне $[0, 1]$, чаще всего $\alpha = 0,5$. Фиксированное значение α приводит к сохранению одинаковой степени хаотичности перемещения на протяжении работы алгоритма, что может ближе к окончанию мешать поиску экстремума, т. к. светлячки будут продолжать слишком сильно менять свои координаты. При малых α агенты могут застревать в локальных экстремумах, а при больших α рой агентов не всегда сможет оказаться в точке оптимума.

При линейном уменьшении $\alpha_t = \alpha_0 (1 - t/T)$, где α_0 – значение α на первой итерации алгоритма, t – текущая итерация, T – максимальное число итераций.

При экспоненциальном уменьшении обычно используется упрощенная формула $\alpha_t = \alpha_0 \theta^t$, где чаще всего $\theta \in [0,95...0,99]$, что позволяет быстрее перейти к локальному уточнению экстремума [3]. На последних итерациях алгоритма наиболее эффективными являются значения $\alpha \rightarrow 0,001$. Недостатком является риск слишком быстрой сходимости к локальному экстремуму.

Типовой способ адаптивного изменения α заключается в том, что если на протяжении нескольких итераций найденное решение не меняется, то α увеличивается для вывода агентов из локальной ловушки. Если найденное решение улучшается, α уменьшается. Недостатком здесь является сложность формализации критерия застревания алгоритма в локальных экстремумах.

Существуют также способы адаптивного управления α исходя из расстояния между агентами. Например, α можно рассчитывать на каждом шаге t как функцию от максимального расстояния D между любыми двумя агентами, автоматически уменьшая α при схождении алгоритма к оптимуму: $\alpha_t = \alpha_0 D_t / D_0$.

Одним из весьма перспективных подходов является адаптивное управление случайным поиском через задание для каждого агента своего собственного значения α . Для самых ярких агентов α устанавливается очень маленьким, с тем чтобы такие агенты не смогли далеко удалиться от найденного экстремума. Для тусклых агентов, наоборот, α должен иметь достаточно высокие значения, чтобы обеспечить для них возможность активного перемещения в области поиска.

Подход к заданию для каждого агента своего собственного значения α был проанализирован для задачи курьерской доставки – расчета маршрутов доставки грузов для множества машин. Светлячком тут является вариант маршрута. Если маршрут получается по длине выше среднего по рою, то значение α должно увеличиваться, что заставит данный маршрут сильнее менять порядок точек, т. е. делать резкие перестановки, чтобы попытаться сократить маршрут. Предложено также учитывать, насколько доставка успевает к клиентам. В данной работе предложена следующая формула для вычисления α для каждого агента: $\alpha_i = \alpha_0 (S_i / S_{worst})^2$, где α_0 – начальное значение α ; S_i – текущий суммарный штраф i -го агента, $S_i = w_1 \Delta L_i + w_2 \Delta T_i$, ΔL – лишние километры сверх лимита; ΔT – опоздание курьера в минутах; w_1 и w_2 – весовые коэффициенты, позволяющие учесть значимость превышения длины маршрута и опоздания (были приняты $w_1 = 1$, $w_2 = 4$); S_{worst} – штраф самого тусклого агента в рою. Если курьер успевает ко всем клиентам, α имеет небольшие значения, близкие к 0,1. Если курьер опаздывает, α принимает большие значения, ближе к 0,8.

Для проверки были взяты алгоритм с константой $\alpha = 0,5$ и алгоритм, в котором α рассчитывался по указанной формуле. Рассматривалась доставка от 20 до 50 заказов несколькими машинами (от 3 до 5). Расчет по данной формуле позволил находить маршруты с меньшим в среднем на 11 % количеством опозданий к клиентам. Таким образом, данная формула позволяет вынудить более тусклых агентов активнее искать маршруты, близкие к оптимальным.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Родзин, С. И. Машинное обучение: метаэвристики дифференциально-векторного движения / С. И. Родзин, О. Н. Родзина. – Чебоксары : Среда, 2024. – 140 с.
2. Firefly algorithm: overview, applications, and modifications / A. Shaban, S. Almufti, R. Asaad, R. Ali // Journal of Image Processing and Intelligent Remote Sensing. – 2025. – Vol. 5, № 5. – P. 1–16.
3. Mohan, K. An Improved Firefly Algorithm for Mining High Utility Itemsets / K. Mohan, J. Anitha // Eng. Technol. Appl. Sci. Res. – 2025. – Vol. 15, № 5. – P. 27922–27930.