

ЧИСЛЕННОЕ МОДЕЛИРОВАНИЕ РОЕВОГО АЛГОРИТМА ПЛАНИРОВАНИЯ ПУТИ В ДВУХМЕРНОЙ НЕКАРТОГРАФИРОВАННОЙ СРЕДЕ

В.А. Костюков, И.М. Медведев, М.Ю. Медведев, В.Х. Пшихопов

Южный федеральный университет, г. Таганрог, Российская Федерация

E-mail: medvmihal@sfnedu.ru

Аннотация. Исследуется эффективность роевых алгоритмов планирования пути в двумерной некартографированной среде. В качестве критериев эффективности используется число итераций в процессе поиска пути и оценка вероятности успешного достижения цели. В ходе исследования изменяется максимальная скорость перемещения роя и максимальное число итераций, в течение которых допускается отсутствие уменьшения расстояния до цели. Предполагается, что каждая частица может определять состояние среды в некоторой локальной области. Под определением состояния имеется в виду определение наличия препятствия в ячейке среды. Для решения проблемы локальных минимумов предлагается вводить виртуальное препятствие в точке локального минимума. Данный подход в целом известен. Новизна этого подхода заключается в том, что решается задача обнаружения локального минимума роем частиц. При одиночном движении обнаружение локального минимума тривиально и сводится к проверке движения к ранее посещенным ячейкам. В групповом случае требуется новое решение задачи обнаружения локального минимума. В данной статье приводится обзор и анализ задачи планирования пути, формулировка проблемы, постановка задачи, математическое описание алгоритмов глобального роевого планирования пути с предложенными модификациями, псевдокоды алгоритмов планирования и результаты численного исследования. В ходе численных исследований определены критерии эффективности планирования пути в среде размером 100×100 ячеек со случайно размещаемыми препятствиями.

Ключевые слова: роевые алгоритмы; двумерная среда; локальный минимум; виртуальные препятствия; локальный поиск; виртуальное препятствие.

Введение

Большинство известных алгоритмов планирования пути базируются на известной карте среды [1, 2]. При отсутствии карты среды высокую эффективность показывают алгоритмы, заимствованные в живой природе [3]. Можно выделить несколько наиболее популярных алгоритмов, используемых в задачах планирования пути. К ним относятся алгоритмы роя частиц (particle swarm optimization – PSO) [4], муравьиный алгоритм [5], пчелиный алгоритм [6], нейросетевые алгоритмы [7], эволюционные алгоритмы [8].

Нейросетевые алгоритмы планирования пути достаточно активно разрабатываются в настоящее время. В работах ведущей научной школы профессора З.В. Нагоева детально обсуждаются проблемы создания нейрокогнитивных мультиагентных систем, рассматриваются модели нейрокогнитивных систем управления роботом, методы и алгоритмы учета контекста диалога, методы и алгоритмы обучения рассматриваемых архитектур, методы и алгоритмы учета неявных знаний и фактов [9–13]. Авторы выражают признательность профессору З.В. Нагоеву за плодотворные дискуссии и считают своим приятным долгом поздравить профессора З.В. Нагоева с юбилеем.

Наиболее эффективные результаты в области планирования пути получены с применением нейронной сети, называемой визуальным трансформером. Однако такая нейронная сеть требует значительного объема вычислений. Кроме того, для обучения трансформера требуется значительный объем данных, который может достигать миллиардов примеров. Относительно простые нейронные сети, независимо от того, обучаются они с подкреплением или с учителем, не обеспечивают высокой надежности. Так, результаты многочисленных работ [14–17] показывают, что

для большинства простых относительно нейронных сетей вероятность успешного достижения целевой точки не превышает 80–90 %. В этой связи такие нейросетевые планировщики дополняются алгоритмами устранения коллизий и выхода из локальных минимумов.

Эволюционные алгоритмы обладают значительно меньшей вычислительной сложностью. В этой связи они также применяются в задачах планирования пути. Наиболее известны генетические алгоритмы, которые представляют собой оптимизационную процедуру, основанную на сочетании случайного поиска и эволюционных способах отбора наилучших особей. Генетические алгоритмы могут применяться как для планирования пути одиночных подвижных объектов [18, 19], так и для построения маршрутов групп роботов [20]. Так, в работе [18] решена задача оптимизации траектории движения космического аппарата по критерию минимальных затрат энергии при переходе от одной орбиты к другой. Для решения задачи нахождения оптимального управления использован генетический алгоритм, с помощью которого осуществлен расчет числа активных участков движения космического аппарата. В работе [19] разработан генетический алгоритм, который применен для планирования пути в динамической среде. Однако в данной статье отсутствуют оценки времени планирования пути, что не дает оценить эффективность примененного алгоритма для динамической среды. В статье [19] рассмотрена задача планирования пути группы беспилотных летательных аппаратов. Проблема сформулирована в виде множественной задачи коммивояжера, для решения которой предложен генетический алгоритм. Эволюционные подходы, в частности генетические алгоритмы, показывают высокую эффективность в статических задачах, характеризуемых большим числом вариантов решений. Однако они, как и другие методы случайного поиска, не гарантируют нахождение глобального оптимума.

Роевые алгоритмы поисковой оптимизации относятся к наиболее часто применяемым алгоритмам среди современных биоэвристик. На них приходится около половины всех ссылок в международных базах данных [21].

Так, в работе [22] предложен метод планирования глобального пути по критерию его длины и гладкости, который базируется на сочетании алгоритма PSO для планирования глобального пути и алгоритма вероятностных дорожных карт [1] для планирования локального пути. Такой подход позволяет устранить основной недостаток алгоритма PSO – попадание в локальные минимумы. Данная проблема является одной из наиболее распространенных при планировании пути [23, 24]. Показано, что предложенный метод способен функционировать в среде с невыпуклыми препятствиями. Однако метод сравнивался только с алгоритмом Дейкстры, который не предназначен для получения гладкого пути и не отличается высокой вычислительной эффективностью.

Для выпуклых препятствий метод PSO применяется более широко в задачах планирования пути. Так, в работе [25] производится сравнение метода PSO с бактериальным методом. Методами численного моделирования в неопределенной среде с несколькими выпуклыми препятствиями показано, что бактериальный алгоритм показывает более высокое быстродействие.

В статье [26] алгоритм PSO также используется для нахождения кратчайшего пути в плоской среде, разбитой на ячейки. Путем моделирования показано, что в простой среде с несколькими препятствиями, занимающими по одной ячейке, алгоритм PSO находит оптимальный по длине путь. Однако рассмотрена среда размерами 10×10 ячеек с одним и тремя препятствиями, что не дает возможности обобщить полученные результаты на более общий случай.

Более сложные среды рассмотрены в работах [27, 28]. В статье [27] рассмотрена среда размером 500×500 клеток, содержащая множество прямоугольных препятствий, расставленных в шахматном порядке. Препятствия не пересекаются друг с другом и не образуют тупики или углы. Для такой относительно простой среды численно исследован глобальный роевой алгоритм при различном числе частиц в рое. Основным отличием алгоритма, исследуемого в работе [27], является учет сенсорной системы, которая позволяет обнаруживать препятствия. Исследована сходимость алгоритма поиска пути при различных дистанциях системы обнаружения препятствий. В статье [28] рассмотрена более сложная среда, в которой препятствия могут образовывать тупики и углы. Используется алгоритм PSO с инерционным коэффициентом, который адаптируется во время движения. В процессе инициализации роя предлагается использовать хаотический подход, заимствованный из нелинейной динамики. Численные исследования показали, что предложенный алгоритм позволяет снизить число итераций для нахождения пути.

В последнее время появились работы по планированию пути для беспилотных летательных аппаратов, базирующиеся на алгоритме PSO. Кроме повышения размерности в трехмерной среде более сложные управляемые объекты [3, 29–31].

Так, в работе [32] исследуется задача планирования движения беспилотного летательного аппарата на основе алгоритма PSO. В статье использован стандартный алгоритм с инерциальным весом. При численном исследовании учтены наличие лидара на борту и модель кинематики движения, которая редуцирована до двумерного случая, т. е. планировалось движение в горизонтальной плоскости. Аналогично в статье [33] рассматривается задача планирования пути беспилотного летательного аппарата в горизонтальной плоскости. При этом начальный путь генерируется с использованием диаграмм Вороного и алгоритма Дейкстры, а алгоритм PSO с инерциальным коэффициентом используется для оптимизации начального пути в динамической конфликтной среде. Проведенное численное исследование показало эффективность предлагаемого подхода. Однако подчеркнем, что рассмотрена фактически двумерная среда и запрещенные области являются выпуклыми.

В работах [34, 35] рассмотрена задача планирования пути в трехмерной среде. В статье [34] используется стандартный алгоритм PSO, который модифицируется с целью улучшения сходимости и вычислительной сложности. Предложенный алгоритм позволяет проводить параллельный поиск по всем трем координатам и учитывает ограничения на скорости. При численном исследовании рассмотрены статические трехмерные среды с препятствиями, описываемыми прямоугольными параллелепипедами. Предложенные улучшения метода PSO позволили существенно в несколько раз уменьшить время нахождения траектории. В работе [35] исследуются различные варианты реализации алгоритма PSO в задаче планирования пути беспилотного летательного аппарата в трехмерной среде. Учитываются запретные области в виде вертикальных цилиндров, а также рельеф поверхности. Вокруг беспилотного летательного аппарата вводятся области опасности и коллизии. Предлагается версия алгоритма, названная сферическим PSO. Предложенный вариант сравнивается с другими версиями PSO в различных трехмерных средах. Путем численных исследований показано, что предлагаемый алгоритм PSO обеспечивает в среднем наименьшее число итераций при планировании пути. Также приводятся результаты натуральных экспериментов.

Проведенный анализ позволяет сделать вывод о том, что наиболее исследованным алгоритм PSO является в средах с выпуклыми препятствиями. Вопрос функционирования алгоритма PSO в средах с локальными минимумами проработан недостаточно. В этой связи в данной статье предлагаются:

- а) алгоритм обнаружения локального минимума роением частиц;
- б) алгоритм устранения локальных минимумов путем введения виртуальных препятствий.

1. Постановка задачи

Рассматривается глобальный алгоритм роя частиц [36] в пространстве второго порядка. Обозначим как N число частиц, положение и скорость которых в момент t описывается векторами $x_i(t) = [x_{i1}, x_{i2}]$ и $v_i(t) = [v_{i1}, v_{i2}]$, $i = 1, 2, \dots, N$.

Изменение положения частицы описывается уравнениями кинематики материальной точки:

$$x_{i1}(t+1) = x_{i1}(t) + v_{i1}(t), x_{i2}(t+1) = x_{i2}(t) + v_{i2}(t). \quad (1)$$

Скорости v_{i1}, v_{i2} в выражении (1) являются управлением, они определяются выражением [37]:

$$v_{ij}(t+1) = wv_{ij}(t) + k_1 r_{1j}(t) [x_{ij}^L(t) - x_{ij}(t)] + k_2 r_{2j}(t) [x_j^G(t) - x_{ij}(t)], \quad (2)$$

здесь $j = 1, 2$ – номер координаты; $x_{ij}^L(t)$ – значение j -й координаты, соответствующее наилучшему значению целевой функции J , достигнутому i -й частицей к моменту времени t ; $x_j^G(t)$ – значение j -й координаты, соответствующее наилучшему значению целевой функции J , достигнутому всем роением; k_1 и k_2 – положительный коэффициенты, определяющие важность локальной и глобальной компонент; $r_{1j}, r_{2j} [0,1]$ – случайные числа, вносящие элемент хаотичности в движение роя; w – весовой коэффициент инерции.

Скорости $v_{ij}(t)$ ограничены по величине максимальным значением, т. е.:

$$\begin{cases} v_{ij}(t+1) = (2), \text{ если } v_{ij}(t+1) < v_{\max}, \\ v_{\max}, \text{ если } v_{ij}(t+1) \geq v_{\max}. \end{cases} \quad (3)$$

В качестве целевой функции J в задачи движения в целевую точку используется Евклидово расстояние от текущего положения частицы роя до целевой точки:

$$J_i = \sqrt{(x_{i1} - x_{t1})^2 + (x_{i2} - x_{t2})^2}, \quad (4)$$

где x_{t1} , x_{t2} – координаты целевой точки.

Коэффициенты r_{1j}, r_{2j} вычисляются в соответствии с выражениями

$$r_{1j} = \text{rand}(0,1), r_{2j} = 1 - r_{1j}, \quad (5)$$

где $\text{rand}(0,1)$ – функция генерации случайного равномерно распределенного в диапазоне (0, 1) числа.

В качестве критерия останова алгоритма используется выполнение одного из двух условий:

$$J_i < \varepsilon, \quad (6)$$

$$k_{it} > k_{\max}, \quad (7)$$

где k_{it} – требуемая точность достижения целевой точки; k_{\max} – максимально допустимое число итераций. При этом неравенство (6) должно выполняться для одного из $i = 1, 2, \dots, N$. Иными словами, цель считается достигнутой, если любая из частиц достигает целевой точки.

Требуется решить следующие задачи:

- предложить и исследовать алгоритм обнаружения локального минимума роем частиц;
- предложить и исследовать алгоритмы выхода из обнаруженного локального минимума;
- исследовать влияние параметров роевого алгоритма, размер роя, весовой коэффициент инерции w и коэффициенты k_1, k_2 на число итераций, необходимых для достижения целевой точки.

2. Описание алгоритмов

Глобальный алгоритм роя частиц может быть описан ПСЕВДОКОДОМ 1, представленным ниже.

ПСЕВДОКОД 1. Глобальный роевой алгоритм

ВХОДНЫЕ ДАННЫЕ: число частиц N , целевая функция J , координаты целевой точки (x_{t1} , x_{t2}), координаты начальной точки (x_{01} , x_{02}), значения коэффициентов w, k_1, k_2 , максимальное значение скорости v_{\max} , требуемая точность достижения целевой точки, максимально допустимое число итераций k_{\max} .

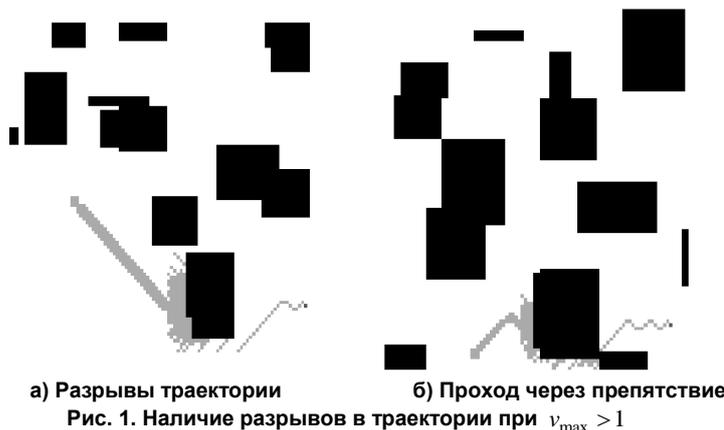
ВЫХОДНЫЕ ДАННЫЕ: Массив координат x_{ij} .

1. СОЗДАНИЕ РОЯ N ЧАСТИЦ.
2. ВЫЧИСЛЕНИЕ НАЧАЛЬНЫХ ЗНАЧЕНИЙ ЦЕЛЕВОЙ ФУНКЦИИ J_i , ЛОКАЛЬНОЙ x_{ij}^L И ГЛОБАЛЬНОЙ x_j^G ЛУЧШИХ ПОЗИЦИЙ.
3. В ЦИКЛЕ ПОКА НЕ ВЫПОЛНЕНО (6) ИЛИ (7):
 - 3.1. ВЫЧИСЛЕНИЕ КОЭФФИЦИЕНТОВ k_1, k_2 .
 - 3.2. ВЫЧИСЛЕНИЕ СКОРОСТЕЙ v_{i1}, v_{i2} .
 - 3.3. ЕСЛИ СЛЕДУЮЩЕЕ ПОЛОЖЕНИЕ ЧАСТИЦЫ НЕ ЗАНЯТО ПРЕПЯТСТВИЕМ, ТО ОБНОВЛЕНИЕ x_{ij} КООРДИНАТ СОГЛАСНО (1).
 - 3.4. ЕСЛИ ОБНАРУЖЕН ЛОКАЛЬНЫЙ МИНИМУМ, ТО ВЫПОЛНЕНИЕ АЛГОРИТМА ВЫХОДА
 - 3.5. ОБНОВЛЕНИЕ ЗНАЧЕНИЙ $J_i, x_{ij}^L(t), x_j^G(t)$.
 - 3.6. ОБНОВЛЕНИЕ КООРДИНАТ x_{ij} .
 - 3.7. УВЕЛИЧЕНИЕ СЧЕТЧИКА ИТЕРАЦИЙ.
 - 3.8. КОНЕЦ ЦИКЛА.

4. КОНЕЦ АЛГОРИТМА.

Предварительное исследование алгоритма, описанного в ПСЕВДОКОДЕ 1, методами численного моделирования привело к следующим модификациям и условиям реализации алгоритма роя частиц.

Реализация алгоритма произведена для дискретного случая. Среда разбита на отдельные ячейки. Для получения непрерывного следа, оставляемого роем при движении от начальной в целевую точку, необходимо, чтобы перемещение частицы на одном шаге осуществлялось в соседнюю ячейку. Отсюда следует, что максимальная скорость перемещения равна $v_{max} = 1$. Нарушение указанного условия может приводить к разрывам в траектории и эффекту пробития препятствий, которые показаны на рис. 1.



При условии, что значения координат x_{ij} и скоростей v_{ij} изменяются дискретно и ограничены единицей на одном такте, более эффективным является аддитивное добавление случайных вариаций в алгоритм вычисления скоростей (2). В данной работе предложена следующая форма уравнений (1), (2):

$$x_{i1}(t+1) = x_{i1}(t) + v_{i1}(t) + r_{1j}(t), \quad x_{i2}(t+1) = x_{i2}(t) + v_{i2}(t) + r_{2j}(t), \quad (8)$$

$$v_{ij}(t+1) = wv_{ij}(t) + k_1[x_{ij}^L(t) - x_{ij}(t)] + k_2[x_j^G(t) - x_{ij}(t)], \quad (9)$$

где r_{1j}, r_{2j} – случайные числа, принимающие значения $-1, 0$ или 1 с равной вероятностью.

Для адаптивной корректировки коэффициенты k_1, k_2 предлагается выбирать в виде функций времени и отдельно настраивать их для каждого элемента роя. Тогда эти коэффициенты обозначаются как $k_{ij}^L(t)$ и $k_{ij}^G(t)$, $j = 1, 2$ для каждого i -го элемента роя на t -шаге. Для их вычисления предлагается следующая процедура. Предположим, что эти коэффициенты нормированы к единице:

$$k_{ij}^L(t) + k_{ij}^G(t) = 1. \quad (10)$$

Пусть $b_{ij}^L(t) = |x_{ij}^L(t) - x_{ij}(t)|$, $b_{ij}^G(t) = |x_j^G(t) - x_{ij}(t)|$, $j = 1, 2$, $i = 1, 2, \dots, N$ есть абсолютные отклонения наилучших достигнутых на данной итерации t локальных и глобальных координат от соответствующих значений координат целевой точки.

Введем следующие величины, характеризующие динамику изменения локальных и глобальных координат для i -го элемента на данном t -шаге:

$$\Delta b_{ij}^L(t) = b_{ij}^L(t) - b_{ij}^L(t-1), \quad j = 1, 2, \quad (11)$$

$$\Delta b_{ij}^G(t) = b_{ij}^G(t) - b_{ij}^G(t-1), \quad j = 1, 2. \quad (12)$$

Заметим, что при уменьшении величины b_{ij}^L , т. е. когда выполнено $\Delta b_{ij}^L(t) < 0$, локальное движение i -й частицы роя происходит успешно, причем тем лучше, чем меньше $\Delta b_{ij}^L(t)$. Анало-

гично при $\Delta b_{ij}^G(t) < 0$ глобальное движение роя как целого происходит успешно, причем тем лучше, чем меньше $\Delta b_{ij}^G(t)$.

Для сопоставления обоих факторов рассмотрим для некоторого i -го элемента роя следующее условие:

$$\Delta b_{ij}^G(t) \geq \Delta b_{ij}^L(t). \quad (13)$$

Если условие (13) выполнено, тогда скажем, что локальное перемещение i -го элемента роя по j -й координате на данный момент t происходит не менее успешно, чем соответствующее глобальное. При этом целесообразно увеличить коэффициент влияния локального перемещения $k_{ij}^L(t) \rightarrow k_{ij}^L(t+1) \geq k_{ij}^L(t)$. Тогда в силу условия (10) заключаем, что коэффициент k_{ij}^G должен уменьшиться, причем по модулю на ту же величину, на которую увеличился коэффициент k_{ij}^L . Аналогично, если выполнено условие, обратное (13), тогда будем говорить, что глобальное перемещение i -го элемента роя по j -й координате на данный момент t происходит более успешно, чем соответствующее локальное.

Чтобы удовлетворить всем указанным выше условиям, можно предложить следующий закон адаптивной корректировки коэффициентов k_{ij}^L и k_{ij}^G :

$$k_{ij}^L(t+1) = \begin{cases} k_{ij}^L(t) + \mu_{ij}^L(t)k_{ij}^G(t), & \text{если } \Delta b_{ij}^G(t) \geq \Delta b_{ij}^L(t); \\ k_{ij}^L(t)(1 - \mu_{ij}^G(t)), & \text{иначе;} \end{cases} \quad (14)$$

$$k_{ij}^G(t+1) = \begin{cases} k_{ij}^G(t)(1 - \mu_{ij}^L(t)), & \text{если } \Delta b_{ij}^G(t) \geq \Delta b_{ij}^L(t); \\ k_{ij}^G(t) + \mu_{ij}^G(t)k_{ij}^L(t), & \text{иначе.} \end{cases} \quad (15)$$

Величины $\mu_{ij}^L(t)$ и $\mu_{ij}^G(t)$ заданы следующими выражениями, изменяющимися на единичном отрезке $[0, 1]$:

$$\mu_{ij}^L(t) = \text{th} [0, 5\eta \hat{\Delta} b_{ij}(t)], \quad (16)$$

$$\mu_{ij}^G(t) = \text{th} [-0, 5\eta \hat{\Delta} b_{ij}(t)], \quad (17)$$

где th – функция гиперболического тангенса.

В выражениях (16), (17) введено обозначение:

$$\hat{\Delta} b_{ij}(t) = \Delta b_{ij}^G(t) - \Delta b_{ij}^L(t). \quad (18)$$

Также принято, что η – некоторый коэффициент, имеющий размерность обратной длины и характеризующий степень нарастания $\mu_{ij}^L(t)$ / спада $\mu_{ij}^G(t)$ с ростом $\hat{\Delta} b_{ij}(t)$. Коэффициент η подбирается эмпирически.

В частности, в этот коэффициент можно внести элемент случайности, свойственный роевым алгоритмам. Тогда он становится функцией j и t . Один из простейших случаев сделать это следующий:

$$\eta = \eta_j(t) = \eta_{min} + r_j(t)(\eta_{max} - \eta_{min}), \quad (19)$$

где $r_j(t)$ – есть случайное число с равномерным распределением в интервале $[0, 1]$ и генерируемое для каждого i -го элемента на временном шаге t , а η_{min} , $\eta_{max} > \eta_{min}$ – некоторые заданные положительные числа – пределы изменения параметра $\eta_j(t)$.

Выполнение ПСЕВДОКОДА 1 требует наличия алгоритма обнаружения локального минимума и алгоритма выхода из этого локального минимума.

Наиболее простым способом выхода из локальных минимумов является внесение случайных вариаций в движение роя. Как следует из выражений (8), (9), такие случайные вариации присутствуют в самом алгоритме роя частиц. В данной статье предлагается числа r_{1j}, r_{2j} приравнять к нулю при движении в свободной области пространства. При попытке движения в ячейку, заня-

тую препятствием, числа r_{1j}, r_{2j} генерируются случайной процедурой. Как показали результаты численных исследований, случайного поиска в локальной области недостаточно для выхода из локальных препятствий. Например, на рис. 2 представлены результаты моделирования, при которых наблюдается попадание в локальный минимум.

При одиночном планировании движения детерминированными алгоритмами обнаружение локального минимума не представляет проблемы. Достаточно обнаружить движение в уже посещенную ячейку, чтобы сделать вывод о наличии локального минимума в текущем положении.

В рассматриваемом случае роя частиц задача осложнена двумя факторами. Первый фактор заключается в наличии компоненты, обусловленной случайными вариациями коэффициентов k_1 и k_2 . Второй фактор заключается в наличии роя.

В данной статье исследуется для обнаружения локального минимума предлагается эвристический алгоритм, определяющий отсутствие изменений в координатах $x_j^G(t)$, $j = 1, 2$. Данный алгоритм содержит настроечный параметр n_{lm} , означающий число итераций, в течение которых координаты, соответствующие наилучшему значению целевой функции J , достигнутому всем роем, не изменяются. При этом точкой локального минимума считается точка $[x_1^G(t), x_2^G(t)]$.

Для выхода из локального минимума предлагается вводить виртуальное препятствие, положение которого совпадает с точкой локального минимума. Виртуальные препятствия активно используются для различных задач обхода препятствий, выхода из локальных минимумов, учета ограничений на маневренность подвижных объектов [24, 37].

Предлагаемый вариант выхода из локального минимума описывается ПСЕВДОКОДОМ 2, представленным ниже.

ПСЕВДОКОД 2. Вариант 1 выхода из локального минимума.

ВХОДНЫЕ ДАННЫЕ: значение n_{lm} , координаты глобального минимума $x_j^G(t)$, $j = 1, 2$, текущие координаты частиц роя x_{ij} .

1. ЕСЛИ КООРДИНАТЫ $x_j^G(t) = x_j^G(t-1)$, $j = 1, 2$, ТО ИНКРЕМЕНТ ДЕТКЕТОРА ЛОКАЛЬНОГО МИНИМУМ $d_{lm} = d_{lm} + 1$. ИНАЧЕ $d_{lm} = 1$.

2. ЕСЛИ $d_{lm} > n_{lm}$, ТО:

2.1. ДОБАВЛЕНИЕ ВИРТУАЛЬНОГО ПРЕПЯТСТВИЯ В ТОЧКУ $x_j^G(t)$.

2.2. СЛУЧАЙНОЕ ИЗМЕНЕНИЕ КООРДИНАТ ЧАСТИЦ, НАХОДЯЩИХСЯ В ЛОКАЛЬНОМ МИНИМУМЕ.

2.3. ВЫЧИСЛЕНИЕ НОВЫХ ЗНАЧЕНИЙ $x_{ij}^L(t)$.

2.4. ВЫЧИСЛЕНИЕ НОВОГО ЗНАЧЕНИЯ $x_j^G(t)$.

3. КОНЕЦ АЛГОРИТМА.

Численное исследование алгоритмов, описанных в ПСЕВДОКОДЕ 1 и ПСЕВДОКОДЕ 2, показывает, что данный способ детектирования и выхода из локальных минимумов чувствителен к параметру n_{lm} . При небольшом значении параметра n_{lm} алгоритм может устанавливать препятствие наиболее перспективной для движения роя точке. Например, на рис. 3 приведен пример, на котором из-за случайных флуктуаций рой временно остановился у края препятствия. При этом алгоритм при $n_{lm} = 3$, установил виртуальное препятствие, отмеченное окружностью. Данная проблема решается путем увеличения параметра n_{lm} , однако при этом число циклов при выходе из тупиков и углов значительно возрастает.

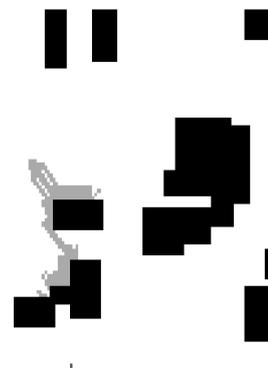


Рис. 2. Попадание роевого алгоритма в локальный минимум

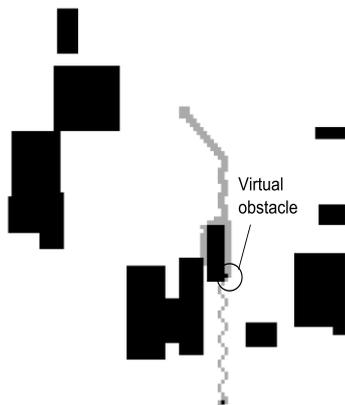


Рис. 3. Установка виртуального препятствия, мешающего движению

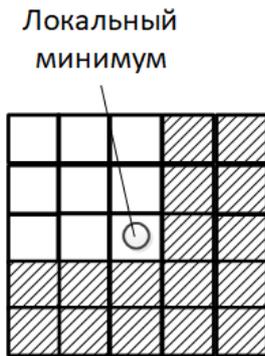


Рис. 4. Анализируемая окрестность локального минимума. Заштрихованы ячейки, занятые препятствиями

В связи с выявленными недостатками алгоритм выхода из локального минимума дополняется анализом локальной области вокруг частицы, находящейся в точке $[x_1^G(t), x_2^G(t)]$. В ходе данного анализа рассматривается область, включающая ближайшие к обнаруженному локальному минимуму 24 ячейки пространства, представленные на рис. 4. Основная цель проводимого анализа – обнаруживать углы, после чего в область локального минимума добавляются виртуальные препятствия.

3. Результаты численного моделирования предложенных алгоритмов и обсуждение

Вначале исследуем исходный роевой алгоритм, описанный в разделе 1. При исследовании будем определять два параметра – среднее число шагов для достижения цели и процент успешного достижения цели.

Моделирование проводилось при объеме экспериментов $N_{\text{э}} = 100$. Размеры моделируемой среды составляют 100×100 ячеек. В среде присутствует 15 препятствий. Минимальные ширина и высота препятствий составляют 1 ячейку. Максимальная ширина и высота препятствий равны 25 ячейкам. Положение нижнего левого угла, ширина и высота прямоугольных препятствий генерируются случайным образом. Максимально допустимое число итераций алгоритма составляет 1200.

Положение начальной и целевой точки, между которыми должен быть проложен путь, также задаются случайным образом. При этом они генерируются таким образом, чтобы расстояние до препятствий было не менее 2 ячеек.

Число частиц роя $N = 9$. Начальное положение частиц роя генерируется следующим образом. Первая частица генерируется в начальной точке траектории. Остальные частицы располагаются в ближайших 8 ячейках, расположенных вокруг начальной точки траектории.

Значения коэффициентов k_1 и k_2 , определяющих веса локальной и глобальной компонент, равны 0,5. Инерционный весовой коэффициент равен $w = 1,0$. Число итераций, при котором не наблюдается уменьшение расстояния до целевой точки и фиксируется локальный минимум, равно $n_{lm} = 5$.

Результаты численного исследования исходного роевого алгоритма представлены в табл. 1. При исследовании в алгоритме использовались случайные коэффициенты r_{1j}, r_{2j} , аддитивно добавляемые в уравнения обновления положений (8). При этом указанные коэффициенты изменяются в соответствии с выражениями

$$r_{ij}(t) = \text{round}(-r_{\max} + r_{\max} \text{rand}), \quad (20)$$

где r_{\max} – максимальное число ячеек, на которые может быть осуществлено перемещение по одной координате за один такт; round – операция округления до ближайшего целого числа; rand – генератор случайного числа, равномерно распределенного в диапазоне $[0, 1]$.

Таблица 1

Исследование исходного роевого алгоритма			
Значение r_{\max}	1	2	4
Среднее число шагов	183	210	155
Доля успешных запусков	90 %	88 %	90 %

В ходе исследования предложенного в данной статье алгоритма изменялся параметр $n_{lm} = 5$. Результаты исследования представлены в табл. 2.

Исследование модифицированного роевого алгоритма

Значение n_{lm}	5	7	10
Среднее число шагов	159	100	93
Доля успешных запусков	94 %	98 %	96 %

Анализ результатов моделирования показывает, что в исходном роевом алгоритме для рассматриваемой среды среднее число шагов для достижения цели равно 183. С ростом параметра r_{max} уменьшается число шагов на достижение цели, что связано с большим разнообразием возможных случайных движений. Средний процент успешности достижения целевой точки составил 89,33 %. Анализ причин непопадания в целевую точку показывает, что при $r_{max} = 1$ это попадание в локальный минимум. При $r_{max} = 2$ большую часть (10 %) составило попадание в локальный минимум, и 2 % – разрывы в траекториях. При $r_{max} = 4$ –60 % составили разрывы в траектории, а 40 % – попадания в локальный минимум. Таким образом увеличение параметра r_{max} является неэффективным для решения проблемы локальных минимумов.

Анализ результатов моделирования предложенного модифицированного роевого алгоритма планирования показывает, что среднее число шагов на достижение цели составляет 118 шагов. Таким образом, число шагов для получения траектории снижено примерно на 35,5 % по сравнению с исходным алгоритмом. Процент успешного попадания в целевую точку составил 96 %, что на 6,67 % выше, чем для исходного алгоритма. На рис. 5 представлены некоторые результаты моделирования предложенных алгоритмов с успешным попаданием в целевую точку

На рис. 5 можно заметить, что в области локальных минимумов наблюдаются достаточно разнообразные движения, что позволяет их успешно преодолевать. Разнообразие движений обусловлено как случайными составляющими, так и изменением значений весовых коэффициентов k_1 , k_2 . На рис. 5 также отдельно показаны виртуальные препятствия, добавленные модифицированным алгоритмом.

Виртуальные препятствия

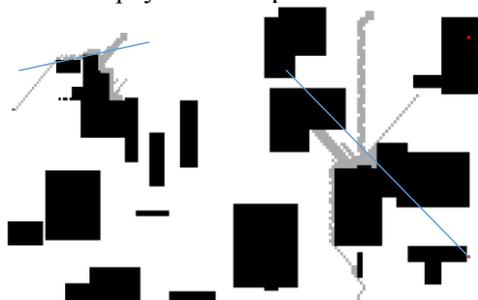


Рис. 5. Результаты моделирования предложенного алгоритма

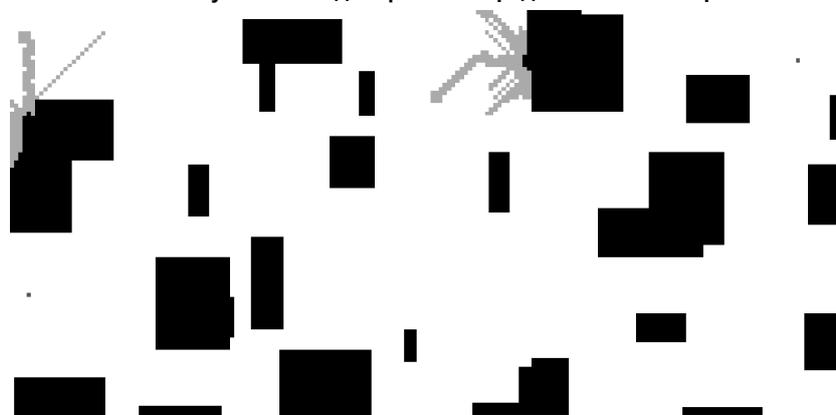


Рис. 6. Примеры попадания предложенного алгоритма в локальные минимумы

Примеры попадания в локальный минимум процедуры планирования с использованием предложенного алгоритма представлен на рис. 6.

На рис. 6 показан локальный минимум, обусловленный глубоким тупиком, и локальный минимум в виде длинной стенки, расположенной поперек движения роя. При максимально допустимом числе итераций, равном 1200, алгоритм не смог выйти из указанных локальных минимумов.

Заключение

В данном исследовании предложена модификация роевого алгоритма поиска пути из заданной начальной точки в заданную целевую точку в некартографированной среде. Сформулирована основная проблема алгоритма, заключающаяся в попадании в локальные минимумы.

Предложены модификации, заключающиеся в новых алгоритмах определения выхода из локального минимума. Для определения локального минимума предложено использовать отсутствие уменьшения целевого функционала – расстояния до целевой точки в течение некоторого количества итераций. При обнаружении локального минимума в него добавляется виртуальное препятствие, что позволяет решить указанную проблему. Дополнительным механизмом выхода из локальных минимумов является предложенный в статье алгоритм изменения весовых коэффициентов. Данный алгоритм позволяет уменьшать влияние ранее обнаруженных текущих «оптимальных» положений роя на выбор следующих направлений движения.

Исследование встроенного в роевой алгоритм механизма случайного выхода из локальных минимумов показало, что при увеличении области случайного поиска возрастает риск получить граф, состоящий из нескольких не связанных друг с другом подграфов. В этом случае требуется дополнительная процедура связывания отдельных подграфов в единый граф.

Исследование предложенного алгоритма показало, что полностью устранить проблему локальных минимумов не удалось, по крайней мере, в пределах заданного максимального числа итераций. В некоторых случаях, например, в глубоких тупиках, выйти из локального минимума не удастся. Одним из возможных направлений решения указанной проблемы является использование поэлементного анализа и выхода из локальных минимумов, а также применение алгоритмов, которые в таких ситуациях более часто добавляют локальные минимумы.

Исследование выполнено за счет гранта Российского научного фонда № 24-19-00063, «Теоретические основы и методы группового управления безэкипажными подводными аппаратами», <https://rscf.ru/project/24-19-00063/> на базе ФГАОУ ВО «Южный федеральный университет».

Литература

1. Казаков, К.А. Обзор современных методов планирования движения / К.А. Казаков, В.А. Семенов // *Труды ИСП РАН*. – 2016. – Т. 28, № 4. – С. 241–294.
2. Generalized Sampling-Based Motion Planners / S. Chakravorty, S. Kumar // *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*. – 2011. – Vol. 41, no. 3. – P. 855–866.
3. Групповое управление подвижными объектами в неопределенных средах / Д.А. Белоглазов, А.П. Гайдук, Е.Ю. Косенко. – М.: ФИЗМАТЛИТ, 2015. – 305 с.
4. Reynolds, C. Flocks, Herds, and Schools: A Distributed Behavioral Model / C. Reynolds // *ACM SIGGRAPH Computer Graphics*. – Vol. 21, Iss. 4. – P. 25–34.
5. Dorigo, M. Ant System: Optimization by a Colony of Cooperating Agents / M. Dorigo, V. Maniezzo, A. Coloni // *IEEE Transactions on Systems, Man, and Cybernetics-Part B*. – 1996. – Vol. 26, no. 1. – P. 29–41.
6. Karaboga, D. An Idea Based On Honey Bee Swarm for Numerical Optimization. Technical Report-TR06 / D. Karaboga. – Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.
7. Pomerleau, D.A. ALVINN: An Autonomous Land Vehicle in a Neural Network / D.A. Pomerleau // *NeurIPS Proceedings*. – 1988. – P. 305–313.
8. Гладков, Л.А. Генетические алгоритмы / Л.А. Гладков, В.В. Курейчик, В.М. Курейчик. – М.: Физматлит, 2010. – 365 с.
9. Архитектура САПР распределенного искусственного интеллекта на основе самоорганизующихся нейрокognитивных архитектур / З.В. Нагоев, З.А. Сундуков, И.А. Пшенокова, В.А. Денисенко // *Известия Кабардино-Балкарского научного центра РАН*. – 2020. – № 2(94). – С. 40–47.
10. Автономный синтез пространственных онтологий в системе принятия решений мобильного робота на основе самоорганизации мультиагентной нейрокognитивной архитектуры / З.В. Нагоев, К.Ч. Бжихатлов, И.А. Пшенокова и др. // *Известия Кабардино-Балкарского научного центра РАН*. – 2020. – № 6 (98). – С. 68–79.

11. Разработка интеллектуальной интегрированной системы «умное поле» / З.В. Нагоев, В.М. Шуганов, А.У. Заммоев и др. // Известия ЮФУ. Технические науки. – 2022. – № 1 (225). – С. 81–91.
12. Классификация и условия применения алгоритмов автоматической онтологизации пространства состояний агента общего искусственного интеллекта под управлением нейрокогнитивной архитектуры / З.В. Нагоев, И.А. Пшенокова, М.И. Анчёков и др. // Известия Кабардино-Балкарского научного центра РАН. – 2023. – № 6 (116). – С. 210–225.
13. Формальная модель генома агента общего искусственного интеллекта на основе мультиагентных нейрокогнитивных архитектур / М.И. Анчёков, А.З. Апшев, К.Ч. Бжихатлов и др. // Известия Кабардино-Балкарского научного центра РАН. – 2023. – № 5 (115). – С. 11–24.
14. Нейросетевая система управления группой роботов в неопределенной двумерной среде / А.Р. Гайдук, О.В. Мартьянов, М.Ю. Медведев и др. // Мехатроника, автоматизация, управление. – 2020. – Т. 21, № 8. – С. 470–479.
15. End to End Learning for Self-Driving Cars / M. Bojarski, D.D. Testa, D. Dworakowski *et al.* // arXiv:1604.07316v1.
16. Off-Road Obstacle Avoidance through End-to-End Learning / Y. LeCun, U. Muller, J. Ben *et al.* // Part of Advances in Neural Information Processing Systems 18 (NIPS 2005). – P. 739–746.
17. Urban Driving with Conditional Imitation Learning / J. Hawke, R. Shen, C. Gurau *et al.* // 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France. – 2020. – P. 251–257.
18. Панкратов, И.А. Генетический алгоритм оптимизации затрат энергии на переориентацию плоскости орбиты космического аппарата / И.А. Панкратов // Мехатроника, Автоматизация, Управление. – 2022. – Т. 23, no. 5. – С. 256–262.
19. Elshamli, A. Genetic Algorithm for Dynamic Path Planning / A. Elshamli, H.A. Abdullah, S. Areibi // Canadian Conference on Electrical and Computer Engineering 2004 (IEEE Cat. No.04CH37513). Niagara Falls, ON, Canada. – 2004. – Vol. 2. – P. 677–680.
20. Планирование маршрутов полета БПЛА в задачах группового патрулирования протяженных территорий / А.Б. Филимонов, Н.Б. Филимонов, Т.К. Нгуен, К.Ф. Фам // Мехатроника, автоматизация, управление. – 2023. – Т. 24, № 7. – С. 374–381.
21. Родзин, С.И. Современное состояние биоэвристик: классификация, бенчмаркинг, области применения / С.И. Родзин // Известия ЮФУ. Технические науки. – 2023. – № 2. – С. 280–298.
22. Masehian, E. A Multi-Objective PSO-based Algorithm for Robot Path Planning / E. Masehian, D. Sedighizadeh // 2010 IEEE International Conference on Industrial Technology, Via del Mar, Chile. – 2010. – P. 465–470.
23. Костюков, В.А. Алгоритм планирования пути в двухмерной среде с полигональными препятствиями на классе кусочно-ломаных траекторий / В.А. Костюков, М.Ю. Медведев, В.Х. Пшихопов // Известия ЮФУ. Технические науки. – 2023. – № 5(235). – С. 34–48.
24. Алгоритмы планирования траекторий в двумерной среде с препятствиями / В.Х. Пшихопов, М.Ю. Медведев, В.А. Костюков и др. // Информатика и автоматизация. – 2022. – Вып. 21(3). – С. 459–492.
25. Nandanwar, M. Implementation and Comparison between PSO and BAT Algorithms for Path Planning with Unknown Environment / M. Nandanwar, A. Nandanwar // *International Journal of Latest Technology in Engineering, Management & Applied Science (IJLTEMAS)*. – 2017. – Vol. 6, Iss. 8S. – P. 67–72.
26. Shortest Path Planning Algorithm – A Particle Swarm Optimization (PSO) Approach / P.I. Adamu, J.T. Jegede, H.I. Okagbue, P.E. Oguntunde // Proceedings of the World Congress on Engineering 2018 Vol I, WCE 2018, July 4–6, 2018. – P. 19–24.
27. Nandanwar, M.K. Path Planning through BAT Algorithm in Complex Environments / M.K. Nandanwar, A.S. Zadagaonkar, D.A. Shukla // *International Journal of Computer Science Trends and Technology (IJCTST)*. – 2016. – Vol. 4, Iss. 1. – P. 79–86.
28. An Improved PSO-GWO Algorithm With Chaos and Adaptive Inertial Weight for Robot Path Planning / X. Cheng, J. Li, C. Zheng *et al.* // *Front. Neurobot.* – 2021. – Vol. 15. – 770361.
29. Qu, Y. A Global Path Planning Algorithm for Fixed-wing UAVs / Y. Qu, Y. Zhang, Y. Zhang // *J. Intell. Robot. Syst.* – 2018. – Vol. 91. – P. 691–707.

30. Автономный подводный аппарат «Скат» для решения задач поиска и обнаружения затонувших объектов / В.Х. Пшихопов, С.Я. Сукокин, Д.Ш. Нагучев и др. // Известия ЮФУ. Технические науки. – 2010. – № 3(104). – С. 153–163.

31. Position-Trajectory Control System for Unmanned Robotic Airship / V.Kh. Pshikhopov, M.Yu. Medvedev, A.R. Gaiduk. *et al.* // IFAC Proceedings Volumes. – 2014. – Vol. 47, Iss. 3. – P. 8953–8958.

32. A Real-Time Path Planning Algorithm for AUV in Unknown Underwater Environment Based on Combining PSO and Waypoint Guidance / Z. Yan, J. Li, Y. Wu, G. Zhang // Sensors. – 2018. – Vol. 19, Iss. 1. – P. 20.

33. Shin, J.J. UAV Path Planning under Dynamic Threats Using an Improved PSO Algorithm / J.J. Shin, H. Bang // International Journal of Aerospace Engineering. – 2020. – Article ID 8820284 (17 pages).

34. A 3D Path Planning Algorithm Based on PSO for Autonomous UAVs Navigation / A. Mirshamsi, S. Godio, A. Nobakhti // Bioinspired Optimization Methods and Their Applications. BIOMA 2020. Lecture Notes in Computer Science, Vol. 12438. – Springer, Cham, 2020.

35. Phung, M.D. Safety-Enhanced UAV Path Planning with Spherical Vector-Based Particle Swarm Optimization / M.D. Phung, Q.P. Ha // Applied Soft Computing. – 2021. – Vol. 107. – Article ID 107376.

36. Скобцов, Ю.А. Эволюционные вычисления / Ю.А. Скобцов, Д.В. Сперанский. – М.: Национальный Открытый Университет «ИНТУИТ», 2016. – Текст: электронный // ЭБС «Консультант студента»: [сайт]. – URL : https://www.studentlibrary.ru/book/intuit_406.html (дата обращения: 29.01.2024).

37. Костюков, В.А. Алгоритмы планирования сглаженных индивидуальных траекторий движения наземных роботов / В.А. Костюков, М.Ю. Медведев, В.Х. Пшихопов // *Мехатроника, автоматизация, управление*. – 2022. – Т. 23, № 11. – С. 585–595.

Поступила в редакцию 16 февраля 2024 г.

Сведения об авторах

Костюков Владимир Александрович – кандидат технических наук, старший научный сотрудник, научно-конструкторское бюро робототехники и систем управления, Южный федеральный университет, г. Таганрог, Российская Федерация, e-mail: wkost-einheit@yandex.ru.

Медведев Илья Михайлович – студент, Южный федеральный университет, г. Таганрог, Российская Федерация, e-mail: imedvede@sfedu.ru.

Медведев Михаил Юрьевич – доктор технических наук, ведущий научный сотрудник НИИ робототехники и процессов управления, Южный федеральный университет, г. Таганрог, Российская Федерация, e-mail: medvmihal@sfedu.ru.

Пшихопов Вячеслав Хасанович – доктор технических наук, профессор, директор НИИ робототехники и процессов управления, Южный федеральный университет, г. Таганрог, Российская Федерация, e-mail: pshichop@rambler.ru.

**SIMULATION OF SWARM ALGORITHMS FOR PATH PLANNING
IN A TWO-DIMENSIONAL NON-MAPPED ENVIRONMENT**

V.A. Kostyukov, I.M. Medvedev, M.Yu. Medvedev, V.Kh. Pshikhopov
Southern Federal University, Taganrog, Russian Federation
E-mail: medvmihal@sfedu.ru

Abstract. This paper examines the effectiveness of swarm path planning algorithms in a two-dimensional unmapped environment. The efficiency criteria are the number of iterations in the path finding process and an assessment of the probability of successfully achieving the goal. During the study, the maximum speed of movement of the swarm and the maximum number of iterations during which it is allowed that the distance to the target does not decrease are changed. It is assumed that each particle can determine the state of the environment in a certain local region. By determining the state we mean determining the presence of an obstacle in a cell of the environment. To solve the problem of local minima, it is proposed to introduce a virtual obstacle at the local minimum point. This approach is generally known. The novelty of this approach lies in the fact that it solves the problem of detecting a local minimum by a swarm of particles. With a single movement, detecting a local minimum is trivial and comes down to checking the movement to previously visited cells. In the group case, a new solution to the problem of detecting a local minimum is required. This article provides a review and analysis of the path planning problem, problem formulation, problem statement, mathematical description of global swarm path planning algorithms with proposed modifications, pseudo-codes of planning algorithms and the results of a numerical study. In the course of numerical studies, the paper presents the criteria for the efficiency of path planning in an environment of 100×100 cells with randomly placed obstacles.

Keywords: *swarm algorithms; two-dimensional environment; local minimum; virtual obstacles; local search; virtual obstacle.*

References

1. Kazakov K.A., Semenov V.A. Reviews of Modern Path Planning Methods. *Proceedings of ISP RAS*, 2016, Vol. 28, no. 4, pp. 241–294.
2. Chakravorty S., Kumar S. Generalized Sampling-Based Motion Planners. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, June 2011, Vol. 41, no. 3, pp. 855–866. DOI: 10.1109/TSMCB.2010.2098438
3. Beloglazov D., Gaiduk A., Kosenko E., Medvedev M., Pshikhopov V., Soloviev V., Titov A., Finaev V., Shapovalov I. *Gruppovoe upravlenie podvizhnymi ob"ektami v neopredelennykh sredakh* (Group Control of Vehicles in Uncertain Environments). Moscow, FIZMATLIT Publ., 2015, 305 p. (in Russ.).
4. Reynolds C. Flocks, Herds, and Schools: A Distributed Behavioral Model. *CACM SIGGRAPH Computer Graphics*, Vol. 21, Iss. 4, p. 25–34. DOI: 10.1145/37402.37406
5. Dorigo M., Maniezzo V., Colomi A. Ant System: Optimization by a Colony of Cooperating Agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 1996, Vol. 26, no. 1, p. 29–41. DOI: 10.1109/3477.484436
6. Karaboga D. *An Idea Based On Honey Bee Swarm for Numerical Optimization. Technical Report-TR06*. Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.
7. Pomerleau D.A. ALVINN: An Autonomous Land Vehicle in a Neural Network. *NeurIPS Proceedings*, 1988, pp. 305–313.
8. Gladkov L.A., Kureychik V.V., Kureychik V.M. *Geneticheskie algoritmy* (Genetic Algorithms), Moscow, Fizmatlit Publ., 2010, 365 p. (in Russ.).

9. Nagoev Z.V., Sundukov Z.A., Pshenokova I.A., Denisenko V.A. Architecture of CAD for Distributed Artificial Intelligence Based on Self-Organizing Neuro-Cognitive Architectures. *News of the Kabardino-Balkarian Scientific Center of the RAS*, 2020, no. 2(94), pp. 40–47. (in Russ.). DOI: 10.35330/1991-6639-2020-2-94-40-47

10. Nagoev Z.V., Bzhikhatlov K.Ch., Pshenokova I.A., Nagoeva O.V., Atalikov B.A., Chechenova N.A., Malyshev D.A. Autonomous Formation of Spatial Ontologies in the Intelligent Decision-Making System of a Mobile Agricultural Robot Based on the Self-Organization of Multi-Agent Neurocognitive Architectures. *News of the Kabardino-Balkarian Scientific Center of the RAS*, 2020, no. 6(98), pp. 68–79. (in Russ.). DOI: 10.35330/1991-6639-2020-6-98-68-79

11. Nagoev Z.V., Shuganov V.M., Zammoev A.U., Bzhikhatlov K.Ch., Ivanov Z.Z. Development of Intelligent Integrated System for “Smart” Agricultural Production. *Izvestiya SFedU. Engineering Sciences*, 2022, no. 1 (225), pp. 81–91. (in Russ.). DOI: 10.18522/2311-3103-2022-1-81-91

12. Nagoev Z.V., Pshenokova I.A., Anchekov M.I., Bzhikhatlov K.Ch., Atalikov B.A., Kankulov S.A., Enes A.Z. Classification and Conditions of Application of Algorithms for Automatic Ontologization of the State Space of a General Artificial Intelligence Agent under the Control of Neurocognitive Architecture. *News of the Kabardino-Balkarian Scientific Center of the RAS*, 2023, no. 6(116), pp. 210–225. (in Russ.). DOI: 10.35330/1991-6639-2023-6-116-210-225

13. Anchekov M.I., Apshev A.Z., Bzhikhatlov K.Ch., Kankulov S.A., Nagoev Z.V., Nagoeva O.V., Pshenokova I.A., Khamov A.A., Enes A.Z. Formal Genome Model of a General Artificial Intelligence Agent Based on Multi-Agent Neurocognitive Architectures. *News of the Kabardino-Balkarian Scientific Center of the RAS*, 2023, no. 5(115), pp. 11–24. (in Russ.). DOI: 10.35330/1991-6639-2023-5-115-11-24

14. Gaiduk A.R., Martjanov O.V., Medvedev M.Yu., Pshikhopov V.Kh., Hamdan N., Farhood A. Neural Network Based Control System for Robots Group Operating in 2-d Uncertain Environment. *Mekhatronika, avtomatizatsiya, upravlenie*, 2020, Vol. 21, no. 8, pp. 470–479. DOI: 10.17587/mau.21.470-479

15. Bojarski M., Testa D.D., Dworakowski D., Firner B., Flepp B., Goyal P., Jackel L.D., Monfort M., Muller U., Zhang J., Zhang X., Zhao J., Zieba K. End to End Learning for Self-Driving Cars. *arXiv: 1604.07316v1*. DOI: 10.48550/arXiv.1604.07316

16. LeCun Y., Muller U., Ben J., Cosatto E., Flepp B. Off-road obstacle avoidance through end-to-end learning. *Part of Advances in Neural Information Processing Systems 18 (NIPS 2005)*, pp. 739–746.

17. Hawke J., Shen R., Gurau C., Sharma S., Reda D., Nikolov N., Mazur P., Micklethwaite S., Griffiths N., Shah A., and Kendall A. Urban Driving with Conditional Imitation Learning. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, Paris, France, 2020, pp. 251–257. DOI: 10.1109/ICRA40945.2020.9197408.

18. Pankratov I. Genetic Algorithm for Optimizing Energy Costs for Reorienting the Orbital Plane of a Spacecraft. *Mekhatronika avtomatizatsiya upravlenie*, 2022, Vol. 23, no. 5, pp. 256–262. DOI: 10.17587/mau.23.256-262

19. Elshamli A., Abdullah H.A., Areibi S. Genetic Algorithm for Dynamic Path Planning. Canadian Conference on Electrical and Computer Engineering 2004 (IEEE Cat. No.04CH37513), Niagara Falls, ON, Canada, 2004, Vol. 2, pp. 677–680. DOI: 10.1109/CCECE.2004.1345203

20. Filimonov A.B., Filimonov N.B., Nguyen T.K., Pham Q.P. Planning of UAV Flight Routes in the Problems of Group Patrolling of the Extended Territories. *Mekhatronika avtomatizatsiya upravlenie*, 2023, Vol. 24, no. 7, pp. 374–381. DOI: 10.17587/mau.24.374-381.

21. Rodzin S. Current State of Bio Heuristics: Classification, Benchmarking, Application Areas. *Izvestiya SFedU. Engineering sciences*, 2023, no. 2, pp. 280–298. DOI: 10.18522/2311-3103-2023-2-280-298

22. Masehian E., Sedighizadeh D. A Multi-Objective PSO-based Algorithm for Robot Path Planning. *2010 IEEE International Conference on Industrial Technology, Via del Mar, Chile*, 2010, pp. 465–470. DOI: 10.1109/ICIT.2010.5472755

23. Kostyukov V., Medvedev M., Pshikhopov V. An Algorithm for Path Planning in a Two-Dimensional Environment With Polygonal Obstacles on a Class of Piecewise Polyline Trajectories. *Izvestiya SFedU. Engineering sciences*, 2023, no. 5(235), pp. 34–48. DOI: 10.18522/2311-3103-2023-5-34-48

24. Pshikhopov V., Medvedev M., Kostyukov V., Hussein F., Kadim A. Trajectory Planning Algorithms in Two-Dimensional Environment with Obstacles. *Informatics and Automation*, 2022, Iss. 21(3), pp. 459–492. DOI: 10.15622/ia.21.3.1
25. Nandanwar M., Nandanwar A. Implementation and Comparison between PSO and BAT Algorithms for Path Planning with Unknown Environment. *International Journal of Latest Technology in Engineering, Management & Applied Science (IJLTEMAS)*, 2017, Vol. 6, Iss. 8S, pp. 67–72.
26. Adamu P.I., Jegede J.T., Okagbue H.I., Oguntunde P.E. Shortest Path Planning Algorithm – A Particle Swarm Optimization (PSO) Approach. Proc. *World Congress on Engineering 2018 Vol I*, WCE 2018, July 4–6, 2018, pp. 19–24.
27. Nandanwar M.K., Zadagaonkar A.S., Shukla D.A. Path Planning through PSO Algorithm in Complex Environments. *International Journal of Computer Science Trends and Technology (IJCTST)*, 2016, Vol. 4, Iss. 1, pp. 79–86.
28. Cheng X., Li J., Zheng C., Zhang J., Zhao M. An Improved PSO-GWO Algorithm With Chaos and Adaptive Inertial Weight for Robot Path Planning. *Front. Neurobot.*, 2021, Vol. 15, 770361. DOI: 10.3389/fnbot.2021.770361
29. Qu Y., Zhang Y., Zhang Y. A Global Path Planning Algorithm for Fixed-wing UAVs. *J. Intell. Robot. Syst.*, 2018, Vol. 91, pp. 691–707. DOI: 10.1007/s10846-017-0729-9
30. Pshihopov V.H., Sukonkin S.Ya., Naguchev D.Sh., Strakovich V.V., Medvedev M.Ju., Gurenko B.V., Kostukov V.A., Voloshchenko Yu.P. Autonomous Underwater Vehicle “Skat” for Search and Detection Silty Object Tasks. *Izvestiya SfedU*, 2010, no. 3(104), pp. 153–163. (in Russ.).
31. Pshikhopov V.Kh., Medvedev M.Yu., Gaiduk A.R., Fedorenko R.V., Krukhmalev V.A., Gurenko B.V. Position-Trajectory Control System for Unmanned Robotic Airship. *IFAC Proceedings Volumes*, 2014, Vol. 47, Iss. 3, P. 8953–8958. DOI: 10.3182/20140824-6-ZA-1003.00393
32. Yan Z., Li J., Wu Y., Zhang G. A Real-Time Path Planning Algorithm for AUV in Unknown Underwater Environment Based on Combining PSO and Waypoint Guidance. *Sensors*, 2018, Vol. 19, Iss. 1, p. 20. DOI: 10.3390/s19010020
33. Shin J.J., Bang H. UAV Path Planning under Dynamic Threats Using an Improved PSO Algorithm. *International Journal of Aerospace Engineering*, 2020, Article ID 8820284, 17 pages. DOI: 10.1155/2020/8820284
34. Mirshamsi A., Godio S., Nobakhti A., Primatesta S., Dovis F., Guglieri G. A 3D Path Planning Algorithm Based on PSO for Autonomous UAVs Navigation. In: Filipič, B., Minisci, E., Vasile, M. (eds) *Bioinspired Optimization Methods and Their Applications. BIOMA 2020. Lecture Notes in Computer Science*, Vol. 12438, Springer, Cham, 2020. DOI: 10.1007/978-3-030-63710-1_21
35. Phung M.D., Ha Q.P. Safety-Enhanced UAV Path Planning with Spherical Vector-Based Particle Swarm Optimization. *Applied Soft Computing*, 2021, Vol. 107, Article ID 107376. DOI: 10.1016/j.asoc.2021.107376
36. Skobcov Yu., Speransky D. *Evolyutsionnye vychisleniya (Evolutionary Calculations)*. Moscow, National Open University, 2016. URL: https://www.studentlibrary.ru/book/intuit_406.html.
37. Kostjukov V.A., Medvedev M.Y., Pshikhopov V.Kh. Algorithms for Planning Smoothed Individual Trajectories of Ground Robots. *Mekhatronika avtomatizatsiya upravlenie*, 2022, Vol. 23, no. 11, pp. 585–595. DOI: 10.17587/mau.23.585-595

Received February 16, 2024

Information about the authors

Kostyukov Vladimir Aleksandrovich is Cand. Sc. (Engineering), Senior Researcher, Joint-Stock Company “Robotics and Control Systems”, Southern Federal University, Taganrog, Russian Federation, e-mail: wkost-einheit@yandex.ru.

Medvedev Ilya Mikhailovich is student, Southern Federal University, Taganrog, Russian Federation, e-mail: imedvede@sfedu.ru.

Medvedev Mikhail Yurjevich is Dr. Sc. (Engineering), Leading Researcher, R&D Institute of Robotics and Control Systems, Southern Federal University, Taganrog, Russian Federation, e-mail: medvmihal@sfedu.ru.

Pshikhopov Viacheslav Khasanovich is Dr. Sc. (Engineering), Professor, Director, R&D Institute of Robotics and Control Systems, Southern Federal University, Taganrog, Russian Federation, e-mail: pshichop@rambler.ru.