

## Планирование траектории движения мобильного робота в среде с препятствиями на основе модифицированного алгоритма оптимизации роя частиц

Шуай Ян\*

Московский государственный университет имени М.В. Ломоносова,  
физический факультет, кафедра физико-математических методов управления  
Россия, 119991, Москва, Ленинские горы, д.1, стр. 2

(Поступила в редакцию 26.05.2023; подписана в печать 17.10.2023)

В данной статье рассматривается разработка интеллектуальной системы для планирования траектории движения мобильного робота в среде с препятствиями. Для решения поставленной задачи автор предлагает использовать модифицированную версию алгоритма оптимизации роя частиц (Particle Swarm Optimization, PSO). Эта методология включает в себя введение двух новых параметров, функционирующих как детекторы для идентификации частиц, не способных улучшить свой индивидуальный и глобальный оптимум в установленном числе последовательных итераций, и заменяющих их на реконфигурированные; модификацию механизма ограничения скорости, что способствует увеличению разнообразия популяции; а также внедрение переменного параметра  $\omega$ , обеспечивающего баланс между глобальными и локальными стратегиями поиска. Этот подход позволяет предотвратить стагнацию и проблемы локальной оптимизации без потери свойств быстрой сходимости PSO. Компьютерная апробация и анализ полученных данных демонстрируют, что предложенный адаптированный алгоритм PSO представляет собой эффективный метод для планирования траектории движения мобильного робота в среде с препятствиями.

PACS: 02.30.Yy

УДК: 519.7

Ключевые слова: планирование траектории, алгоритм оптимизации роя частиц, модифицированный алгоритм оптимизации роя частиц.

### ВВЕДЕНИЕ

Современные достижения в области робототехники обозначили роботов как интегральный элемент технологической инфраструктуры современного общества. Данные устройства обладают потенциалом для эффективного выполнения многообразных функций в условиях повышенной опасности и сложности, обеспечивая тем самым снижение рисков, ранее связанных с человеческим трудом. Одним из важнейших аспектов инженерии робототехнических систем является разработка методов планирования траектории. Данная технология предполагает создание алгоритмов, позволяющих роботам определять оптимальные траектории перемещения от начальной точки до целевой локации, избегая при этом конфликтных ситуаций с препятствиями или другими объектами в операционной среде [1–3].

С точки зрения оптимизации, решение задачи оптимального планирования двумерной траектории является NP-трудным, и общего решения этой задачи не существует. В современной научной практике планирование траектории обычно выполняется следующими группами алгоритмов:

- Графовые алгоритмы поиска траектории.
- Метод потенциальные поля.
- Случайные и итеративные алгоритмы.

### • Роевые алгоритмы

**Графовые алгоритмы поиска траектории** (grid-based methods), например, A\* (A-star), Dijkstra, Theta\* и др. Специфицированные алгоритмы действуют в рамках графовых структур, в которых вершины символизируют потенциальные локации, а рёбра обозначают возможные пути перехода между ними. Первоочередной целью является нахождение наиболее кратчайшей траектории, соединяющей начальную и конечную точки [4–6].

**Метод потенциальные поля** (potential fields) использует виртуальные «потенциальные поля» для планирования движения робота в пространстве. Целевая точка создает гравитационное поле, направляя робота на себя, а препятствия создают отталкивающие поля, не давая роботу столкнуться с ними. Робот движется в соответствии с градиентом составного поля, созданным суперпозицией всех отдельных полей, стремясь минимизировать потенциальную энергию и достичь целевой точки, избегая препятствий [4–6].

**Случайные и итеративные алгоритмы** (sampling-based methods), на пример, Rapidly-exploring Random Trees (RRT), Probabilistic Roadmaps (PRM) и др. Данные методики преимущественно прибегают к стохастическим процессам при исследовании пространственных доменов, целью которых является обнаружение решений даже в пространствах с высокой размерностью, хотя это может влечь за собой определённое уменьшение оптимальности результата [4–6].

**Роевые алгоритмы** (поведенческие алгоритмы, встречающиеся в живой природе), например, генетический, муравьиный, пчелиный алгоритм, метод роя

\* shuai.yang21@physics.msu.ru

частиц и др. Эти алгоритмы основаны на имитации естественного поведения живой природы и могут решать NP-сложные задачи для создания оптимальных решений [7, 8].

Проанализировав четыре категории методологий планирования траектории, можно заключить, что они демонстрируют заметное временное затруднение выполнения и предрасположенность к локальным минимумам, особенно при реализации движений мобильного робота в условиях множественных ограничений на планирование траектории. В этом контексте роевые алгоритмы, в частности метаэвристические, могут представлять собой наиболее эффективные методики планирования траекторий в двухмерных детерминированных пространствах.

В данной статье автор применяет модифицированный алгоритм оптимизации роя частиц. В рамках данного подхода робот анализируется как материальная точка, что значительно сокращает объем вычислительных операций и способствует достижению оптимизированного решения. Данный алгоритм позволяет системам робототехники определить оптимальную траекторию, минимизируя риски коллизий и принимая во внимание ограничения, связанные с движением.

## 1. ПОСТАНОВКА ЗАДАЧИ

В рамках данного исследования предполагается, что робот осуществляет движение в двухмерном евклидовом пространстве, где присутствуют стационарные препятствия. Основная задача планирования траектории заключается в генерации наиболее оптимальной траектории, связывающей начальную точку ( $SP$ ) и целевое местоположение ( $GP$ ) и исключающей столкновения с препятствиями. В данном контексте робот моделируется как материальная точка, обеспечивающая значительное упрощение компьютерных вычислений при сохранении достоверности прогнозируемых траекторий. Сценарий движения описывает пространство, насыщенное статическими препятствиями, где замкнутые круги символизируют препятствия, в то время как неокрашенные участки указывают на зоны, доступные для свободного перемещения робота.

С целью формирования траектории, следует определить последовательность контрольных точек  $WPI$ ,  $i = 1, \dots, n$ , служащих в качестве ориентиров для построения оптимальной траектории. Реализация гладкой траектории достигается посредством применения сплайн-функций в среде Matlab, базирующихся на установленных контрольных точках. Как иллюстрируется на рис. 1, геометрические фигуры (квадраты и звезды) демонстрируют начальные и конечные точки траектории соответственно.

Отметим, что в контексте данного метода три контрольные точки консолидируются в одну единицу, представляющую собой одну частицу.

Задача планирования траектории заключается

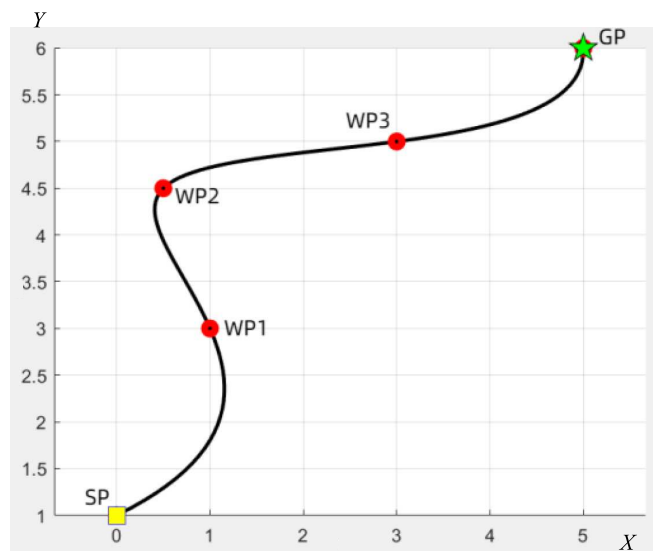


Рис. 1. Траектория, полученная на основе сплайн-функции и контрольных точек

в определении трех ключевых контрольных точек, которые служат основой для создания оптимизированной траектории. Эта траектория позволяет роботу эффективно осуществлять переход из начального положения в целевое с препятствиями. При этом основные параметры оптимизации включают минимизацию расстояния и времени перемещения, оптимизацию энергопотребления и максимизацию безопасности перемещения.

**Объективная функция**, подлежащая минимизации в процессе алгоритмического планирования траектории, разделяется на два ключевых аспекта: первый направлен на обеспечение безопасности траектории путём безопасного расстояния до ближайшего препятствия, в то время как второй фокусируется на минимизации длины траектории для повышения эффективности движения. С учётом безопасности, значение объективной функции увеличивается при уменьшении расчётного расстояния до препятствия до уровня ниже установленного безопасного предела. В связи с этим, во время итерационного процесса отбраковываются частицы с повышенными показателями объективной функции (т.е. частицы, расстояние до препятствия которых меньше безопасного расстояния). С другой стороны, в контексте минимизации длины траектории, после определения трех контрольных точек и построения гладкой траектории производится расчет объективной функции и сохраняется траектория с наименьшим значением объективной функции. Далее следует последовательность итераций, в результате которых находится оптимальная кратчайшая траектория.

Таким образом, при заданных начальной и конечной точках, а также характеристиках препятствия, задача планирования траектории сведется к определению трех оптимальных контрольных точек. Эти точки в дальней-

шем служат основой для формирования гладкой траектории посредством функции сплайна в программной среде MATLAB. В этом контексте каждое возможное решение представляет собой конкретный набор контрольных точек в многомерном пространстве решений, причем начальные наборы этих точек генерируются случайным образом.

## 2. КЛАССИЧЕСКИЙ АЛГОРИТМ ОПТИМИЗАЦИИ РОЯ ЧАСТИЦ КЕННЕДИ И ЭБЕРХАРДА

Алгоритм оптимизации роя частиц (Particle Swarm Optimization, PSO), впервые предложенный Кеннеди и Эберхардом, представляет собой биоинспирированный метод, воспроизводящий социальные аналогии, заметные в коллективном поведении животных. В рамках данного алгоритма, каждое предложенное решение в пространстве поиска интерпретируется как «частица». Алгоритм начинается с создания исходного роя, где каждая частица инициализируется случайным образом в пространстве возможных решений. Адаптация положения каждой частицы в пространстве поиска основывается на психосоциальной динамике, где индивиды стремятся имитировать успешные стратегии других участников группы. В этом контексте рой формируется из многочисленных частиц, каждая из которых представляет потенциальное решение проблемы. Положение каждой частицы модифицируется в соответствии с ее личным опытом и наблюдениями за успешными стратегиями соседей [7–9].

В процессе реализации алгоритма оптимизации роя частиц можно выделить следующие этапы:

- **Фаза Инициализации:** В данной фазе каждой частице, что является аналогом трех контрольных точек, присваиваются начальные значения скорости и позиции в пространстве решений.
- **Фаза Определения Глобального Оптимума:** На этом этапе производится расчет и идентификация глобального оптимального решения, т.е. определение той частицы, у которой значение целевой функции минимально.
- **Фаза Обновления Частиц:** В рамках этой фазы осуществляется коррекция скорости и позиции каждой частицы с применением соответствующего математического уравнения. В фазе обновления алгоритма оптимизации роя частиц, динамические параметры каждой частицы подвергаются коррекции: их скорость обновляется согласно уравнению (1), а координаты положения — согласно уравнению (2). Этот процесс является критически важным для эффективного поиска оптимальной траектории в многомерном пространстве решений, обеспечивая последователь-

ное приближение к глобальному оптимуму.

$$v_{n+1} = \omega v_n + c_1 \cdot \text{rand}_1 \cdot (pBest_n - x_n) + c_2 \cdot \text{rand}_2 \cdot (gBest_n - x_n), \quad (1)$$

$$x_{n+1} = x_n + v_{n+1}. \quad (2)$$

где  $v_{n+1}$  — скорость частицы в момент времени  $n+1$ ,  $v_n$  — скорость частицы в момент времени  $n$ ,  $x_{n+1}$  — положение частицы в момент времени  $n+1$ ,  $x_n$  — положение частицы в момент времени  $n$ ,  $\omega$  — коэффициент инерции, который играет роль поддержания баланса между возможностями глобального и локального поиска. Большой  $\omega$  дает алгоритму сильную способность к глобальному поиску, в то время как меньший концентрируется на локальном поиске.  $c_1$  — когнитивный коэффициент,  $c_2$  — социальный коэффициент.  $\text{rand}_1$  и  $\text{rand}_2$  — независимые случайные величины, равномерно распределенные в  $[0, 1]$ ;  $pBest$  — персональное наилучшее положение, достигнутое частицей, а  $gBest$  — глобально наилучшее положение, достигнутое группой.

- **Фаза Расчета И Оптимизации Решений:** На данном этапе происходит расчет и сохранение как глобального, так и индивидуального оптимальных решений с целью последующей оптимизации процесса поиска.
- **Фаза Итерационного Повторения:** Эта фаза предполагает многократное повторение третьего этапа до тех пор, пока не будет выполнено условие остановки. Критериями остановки алгоритма оптимизации роя частиц (PSO) чаще всего служат достижение максимального предопределённого числа итераций или удовлетворение условию минимального уровня ошибки. Эти критерии обеспечивают баланс между временем вычислений и достигаемой точностью решения, предотвращая возможное переобучение или закливание алгоритма.

## 3. МОДИФИКАЦИЯ КЛАССИЧЕСКОГО АЛГОРИТМА PSO

В целях обеспечения гармонического баланса между глобальным и локальным поисками и минимизации вероятности попадания в локальные оптимумы, в классический алгоритм роя частиц вносятся следующие три модификации:

- Введение локального детектор оптимума.
- Изменение ограничения скорости.
- Введение переменного коэффициент  $\omega$ .

**Введение локального детектор оптимума** (Local Optimum Detector, LOD) представляет собой важный элемент общего процесса оптимизации, служащий для динамического мониторинга и адаптации хода оптимизации, проводимой каждым роботом индивидуально. Когда «личный лучший» результат ( $pBest$ )  $i$ -й частицы остается неизменным на протяжении заданного числа последовательных поколений, это может указывать на то, что соответствующий подпроцесс оптимизации находится в стадии застревания в локальном оптимуме. В таких случаях, значение  $LOD$ , ассоциированное с  $pBest$  (или  $LODpBest$ ), подвергается увеличению. Аналогичный процесс проводится на уровне глобального лучшего решения ( $gBest$ ), с соответствующей коррекцией параметра  $LODgBest$  [10, 11].

Допустим,  $Sp$  и  $Sg$  являются параметрами, конфигурируемыми пользователем. В сценарии, где значение  $pBest$  отдельной частицы не показывает улучшения в течение заранее определенного порогового периода, определенного как  $LODpBest=Sp$ , эта частица подвергается процессу реструктуризации, в результате которой ее  $pBest$  трансформируется в  $pBest\_temp$ , как описано в уравнении (3). Аналогично, в случаях, когда глобальный оптимальный результат ( $gBest$ ) не улучшается до значения  $LODgBest=Sg$ , происходит реструктуризация в состояние  $pBest\_temp$ , как представлено в уравнении (4). Эта динамическая адаптация является средством более эффективного поиска оптимизационного решения, позволяя алгоритму избегать застревания в локальных минимумах и стремиться к обнаружению глобальных оптимумов.

$$pBest\_temp_i = pBest_{i_1} + \frac{i_1}{i_1 + i_2} \times (gBest\_hist_{i_2} - pBest_i), \quad (3)$$

$$gBest\_temp_i = \min(pBest\_temp_i, gBest_i). \quad (4)$$

где  $pBest\_temp_i$  — локальный оптимум перезагрузки,  $gBest\_temp_i$  — глобальный оптимум перезагрузки,  $i_1 = \text{random}(1, M)$ ,  $M$  — размер популяции,  $i_2 = \text{random}(1, \text{size}(gBest\_hist))$ , а  $gBest\_hist$  представляет исторические значения  $gBest$ .

Эта динамическая стратегия обновления  $LOD$  способствует более гибкому и адаптивному поиску оптимальных решений, позволяя алгоритму эффективно балансировать между локальным и глобальным поиском и предотвращая преждевременную сходимость к локальным оптимумам.

**Изменение ограничения скорости.** В процессе выполнения алгоритма возможно возникновение ситуаций, когда скорость частицы может достигать чрезмерно высоких значений. В таких случаях применяется ограничение на скорость, чтобы контролировать траекторию частиц. Определим максимальное допустимое значение скорости как  $Vmax$ . Если обновленная скорость  $Vn + 1 > Vmax$ , тогда  $Vn + 1 = Vmax$ .

Если обновленная скорость  $Vn + 1 < -Vmax$ , тогда  $Vn + 1 = -Vmax$ .

Однако следует отметить, что применение такого ограничения может ограничить возможность алгоритма «вырваться» из локальных оптимумов. В связи с этим введена дополнительная коррекция, при которой скорость может быть установлена на случайное значение в ситуациях, когда она превышает установленное максимальное ограничение. Это позволяет сохранить потенциал для обнаружения более широкого диапазона возможных оптимумов.

Если  $Vn + 1 > Vmax$ , то

$$Vn + 1 = (0.5 + \text{rand}/2) \times Vmax,$$

Если  $Vn + 1 < -Vmax$ , то

$$Vn + 1 = -(0.5 + \text{rand}/2) \times Vmax,$$

где  $\text{rand}$  — случайное число в  $[0,1]$ , то рандомизация скорости усиливает и увеличивает вероятность выхода частиц из локального оптимума.

**Введение переменного коэффициента  $\omega$ .** В процессе выполнения алгоритма оптимизации роя частиц, частицы могут быстро конвергировать к текущему оптимальному решению, обнаруженному одной из частиц. Однако это может привести к тому, что алгоритм оказывается застрявшим в локальном оптимуме, неспособным продолжить поиск в пространстве решений. Чтобы преодолеть эту проблему, предлагается метод адаптивного управления коэффициентом инерции  $\omega$ , в котором  $\omega$  линейно уменьшается по мере увеличения числа итераций.

В рамках настоящего исследования стратегия адаптивного управления коэффициентом инерции конструируется таким образом, чтобы начинаться с максимального значения, последовательно уменьшаться с большого до малого и, в конечном итоге, стабилизироваться около нулевого уровня как минимальная величина. Данный подход обеспечивает оптимальное соотношение между глобальной и локальной поисковой способностью, фактически облегчая нахождение глобального оптимума и предотвращая преждевременную конвергенцию к локальным оптимумам.

Коэффициент инерции обновляется в соответствии с уравнением (5):

$$w = \exp(-20 \times (r/r_{max})^{10}), \quad (5)$$

где  $r$  — количество текущих итераций,  $r_{max}$  — общее количество итераций.

#### 4. КОМПЬЮТЕРНАЯ АПРОБАЦИЯ МОДИФИЦИРОВАННОГО АЛГОРИТМА PSO В MATLAB

В экспериментальной настройке алгоритма оптимизации роя частиц, максимальное установленное количество итераций равно 100. Общая популяция частиц



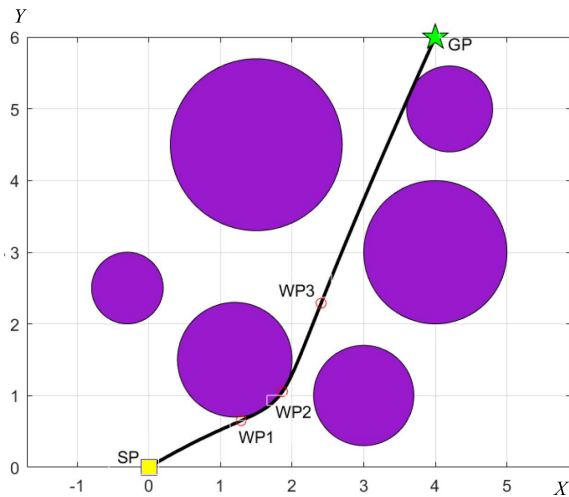


Рис. 2. Траектория движения робота

в роле определена в размере 800 единиц. Исходное приближение к оптимальному решению генерируется случайным образом в допустимом пространстве возможных решений. По завершении 100 итераций, алгоритм выведет оптимальную траекторию, являющуюся наиболее предпочтительным путем, вычисленным на основе процесса оптимизации.

Как демонстрируется на рис. 2, определены координаты оптимальных точек управления, обозначенные как WP1, WP2 и WP3, с числовыми значениями соответственно (1.2874, 0.6484), (1.8655, 1.0547) и (2.4053, 2.2878). Вычисленная оптимальная длина траектории составляет приблизительно 7.5483 единиц.

На рис. 3 проиллюстрирована динамика изменения длины оптимальной траектории в зависимости от числа проведенных итераций. Заметно, что после примерно 20 итераций длина траектории асимптотически при-

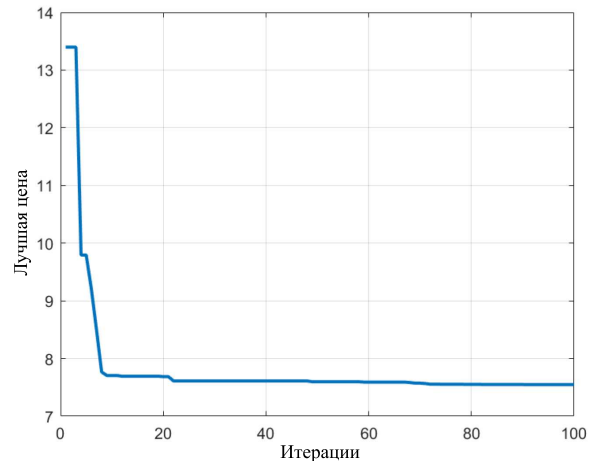


Рис. 3. Изменение значения объективной функции с итерациями

ближается к значению, наблюдаемому после 100 итераций, что свидетельствует о высокой эффективности и скорости сходимости алгоритма оптимизации роя частиц.

## ЗАКЛЮЧЕНИЕ

По результатам численного моделирования была подтверждена высокая эффективность модифицированного алгоритма, представленного в настоящем исследовании. Предложенная модификация способствует повышению генетического разнообразия популяции, обеспечивая оптимальное сочетание глобального и локального поиска. Это позволяет избежать проблем стагнации и локальной оптимизации, при этом сохраняя преимущества быстрой сходимости, присущие классическому алгоритму роя частиц (PSO).

- [1] Чернышев Н.Н., Нижнец Т.В. // Вестник Воронежского государственного технического университета. 2022. – Т. 18. – №. 6. – С. 7-14.
- [2] Дас Н.Ч. и др. Алгоритм планирования траектории мобильного робота. Международная конференция по мягким вычислениям и измерениям. Санкт-Петербургский государственный электротехнический университет ЛЭТИ им. В.И. Ульянова (Ленина). 2021. **1**. 228.
- [3] Qingzhou L., Feng W. // Computer Engineering. **46**. N 4.
- [4] Hamdan N. et al. Study of Path Planning Methods in Two-Dimensional Mapped Environments. International Conference on Interactive Collaborative Robotics. pp. 137-150. Cham: Springer Nature Switzerland, 2023.
- [5] Rajai A.N.A. et al. // Journal of Robotics. 2022. **2022**.
- [6] Ngwenya T., Ayomoh M. // Sensors. 2022. **22**, N 18. 6943.
- [7] Kennedy J., Eberhart R. // Proceedings of ICNN'95-international conference on neural networks. IEEE. 1995. **4**. 1942.
- [8] Yang M., Li C. Path planning and tracking for multi-robot system based on improved PSO algorithm. 2011 International Conference on Mechatronic Science, Electric Engineering and Computer (MEC). IEEE, 2011. pp. 1667-1670.
- [9] Wang Y., Cai F. Application of particle swarm optimization in path planning of mobile robot. AIP Conference Proceedings. **1864**. N 1. 020023. AIP Publishing LLC, 2017.
- [10] Ayari A., Bouamama S. // Robotics and biomimetics. 2017. **4**, N 1. 1.
- [11] Hereford J.M. A distributed particle swarm optimization algorithm for swarm robotic applications. 2006 IEEE International conference on evolutionary computation. IEEE, 2006. pp. 1678-1685.

## Mobile robot trajectory planning in an environment with obstacles based on a modified particle swarm optimization algorithm

Shuai Yang

*Department of Physical and Mathematical Methods of Control, Faculty of Physics, Lomonosov Moscow State University,  
Moscow 119991, Russia  
E-mail: shuai.yang21@physics.msu.ru*

This article discusses the development of an intelligent system for planning the trajectory of a mobile robot in an environment with obstacles. To solve this problem, the authors propose to use a modified particle swarm optimization (PSO) algorithm. The modification involves three aspects. Firstly, the algorithm introduces two pivotal parameters functioning as detectors. These detectors are proficient in identifying particles that fail to augment their respective individual and global optimums within a specified sequence of iterations, thereby replacing them with reconfigured counterparts. Furthermore, introducing a variable parameter  $\omega$ , aims to strike an optimal balance between global and local search strategies. These modifications circumvent the potential pitfalls of stagnation and local optimization dilemmas and retain the rapid convergence attributes innate to the PSO algorithm. Modeling and analysis of the obtained data allow us to conclude that the proposed modified PSO algorithm is effective for planning the trajectory of a mobile robot in an environment with obstacles.

PACS: 02.30.Yy.

*Keywords:* trajectory planning, particle swarm optimization algorithm, modified particle swarm optimization algorithm.  
*Received 26 May 2023.*

### Сведения об авторе

Ян Шуай — аспирант; e-mail: shuai.yang21@physics.msu.ru.