

Роевой алгоритм транспортного потока *

О. П. Бобровская^{1,2}, Т. В. Гавриленко^{1,2}, В. А. Галкин^{1,2}

¹ Сургутский государственный университет, Сургут, Россия

² Сургутский филиал Федерального научного центра Научно-исследовательский институт системных исследований РАН, Сургут, Россия

Аннотация. Решается задача моделирования маршрутов автопилотируемых транспортных средств в транспортном потоке, при котором отсутствуют их столкновения. Предлагается новый роевой алгоритм, основанный на микроскопической модели транспортного потока, обеспечивающий движение агентов без столкновений. Рассматривается изменение в процессе работы алгоритма нескольких критериев оптимальности, таких как: средняя скорость агентов, пропускная способность, количество перестроений. Оцениваются границы эффективных значений гиперпараметров алгоритма. При определенных параметрах плотности и коэффициентах отталкивания/притяжения в транспортном потоке наблюдаются свободный поток и улучшение значений критериев оптимизации.

Ключевые слова: роевой интеллект, роевой алгоритм, микроскопическая модель транспортного потока, агент, алгоритм косяка рыб.

DOI 10.14357/20718594230406

EDN SJCXCN

Введение

Агентный подход, то есть модель, состоящая из множества независимых частиц с одинаковым набором реакций на окружающую среду, а также включающая в себя способы принятия решений, правила обучения, топологию взаимодействия агентов и среды, находит свое применение во многих областях науки и техники [1]: многоагентные системы, позволяющие решать задачи логистики и моделировать социальные структуры; роевые алгоритмы, обеспечивающие численную оптимизацию функций, не являющихся дифференцируемыми и гладкими; клеточные автоматы, описывающие жидкость в виде множества частиц, перемещающихся между узлами дискретной решетки, и обеспечивающие возможность параллельных вычислений; микроскопические модели транспортного потока [2–3], рассматривающие не отдельные

макрохарактеристики потока, а множество реальных автомобилей, взаимодействие которых эти макрохарактеристики формирует. Несмотря на различные термины и степень формализации, используемые в работах по этим направлениям, близка суть подходов, заключающаяся в запуске модели со множеством агентов и предоставлении им возможности взаимодействовать друг с другом.

Роевыми алгоритмами или роевым интеллектом называются алгоритмы, использующие коллективное поведение децентрализованной самоорганизующейся системы, в основном, являющееся обобщением биологических систем. Они используются для численной оптимизации функций в случаях, когда применение классических методов является проблематичным, потому что не предъявляют строгих требований к функции, вроде гладкости, монотонности, дифференцируемости. Роевые алгоритмы являются

* Публикация выполнена в рамках государственного задания ФГУ ФНЦ НИИСИ РАН (Выполнение фундаментальных научных исследований ГП 47) по теме № 0580-2021-0007 «Развитие методов математического моделирования распределенных систем и соответствующих методов вычисления».

эвристическими, то есть направлены на поиск приближенного эффективного решения, когда другие методы не в состоянии найти точное решение при заданных ограничениях. Существует множество роевых алгоритмов, и продолжают создаваться новые. В недавно вышедшей работе [4] приводится список из 511 алгоритмов, среди которых представлены алгоритмы, вдохновленные поведением животных, растений, небологическими природными явлениями, поведением человека и созданными им предметами. Многие из алгоритмов широко исследуются и используются, например, алгоритм роя частиц [5], алгоритм серых волков [6], муравьиный алгоритм [7], пчелиный алгоритм [7], алгоритм летучих мышей [7], алгоритм кукушки [8]. Кроме того, разрабатываются гибридные модели, как в работе [9].

Остановимся на алгоритме косяка рыб, созданном в 2007 году и представленном в работе [10] (эффективность алгоритма исследуется в работе [11]). Можно рассматривать и другие алгоритмы, но выберем этот в качестве примера. Если обратить внимание на источник вдохновения – косяк рыб, то можно отметить, что он представляет собой множество идентичных агентов, которые двинутся без столкновений по сложными траекториям. Что напоминает интересующий нас транспортный поток.

Цель алгоритма – найти глобальный минимум функции. Для этого создается косяк рыб, состоящий из особей с заданными одинаковыми правилами поведения. У каждой рыбы есть вес и положение в пространстве. На каждом шаге вес обновляется на величину, зависящую от изменения значения оптимизируемой функции в результате перемещения на предыдущем шаге. После чего на основе предыдущих координат и веса всего косяка обновляется координата каждой из рыб. Более подробно алгоритм будет приведен ниже.

В данном роевом алгоритме не рассматриваются проблема пересечения траекторий и задача избегания столкновения агентов. В том числе это связано с тем, что агенты ведут себя как материальные точки. Физическое столкновение в данном случае является лишь повторным посещением агентами одной и той же точки пространства, которое может дать новый результат благодаря различным значениям параметра веса различных агентов.

Есть несколько метаэвристических оптимизационных алгоритмов, которые сами основаны на идеях движения автомобилей [12–13].

В области дорожного движения роевые алгоритмы применяются для решения транспортных задач, таких как задача коммивояжера [14] или маршрутизации транспорта [15], управления длительностью сигналов светофора [16]. Применение алгоритмов муравьиной колонии, роя частиц и пчелиной колонии в решении транспортных задач рассматривается в работе [17].

Роевой интеллект успешно используется для решения задачи управления дронами [18–19], но не транспортным потоком. Хотя, как отмечается в работе [20], рой автономных автомобилей может оптимизировать транспортный поток и сокращать заторы посредством координации действий агентов внутри роя и избегания коллизий.

Тем не менее к роевым алгоритмам близка суть подхода микроскопического моделирования транспортного потока. При моделировании движения агентов (косяка рыб, стаи птиц, потока автомобилей, в том числе беспилотных) ставится задача, которую мы планируем решить в данной статье: эффективное в заданных ограничениях перемещение всех агентов из начального положения в конечное таким образом, чтобы в процессе движения они не сталкивались. В данной постановке задачи оптимизируется количество изменений скорости, изменений направления или пропускная способность дороги / канала движения, что будет рассмотрено подробнее в обсуждении. Поскольку роевые алгоритмы являются методами численной оптимизации, в результате решения поставленной задачи может быть предложен роевой алгоритм, основанный на поведении транспортного потока.

Более строго задачу моделирования можно сформулировать следующим образом: построить функции $x(t)$ и $v(t)$, где $x(t)$ – закон изменения координат автомобилей, в момент времени $\tau = 1..N$ значением функции является набор элементов из бесконечного множества $X = \{x | \forall x_{\tau i} \neq x_{\tau j}, i \neq j, i, j = 1..n\}$, $x_i \in X_u$, где X_u – задаваемый пользователем маршрут, $v(t)$ – закон изменения скоростей автомобилей, в момент времени $\tau = 1..N$ значением функции является набор элементов из бесконечного множества $V = \{v | \forall v_i \leq v_{max}, i = 1..n\}$, $v_i \in V_u$, где V_u – возможные значения скорости автомобилей, ограниченные задаваемым пользователем

значением максимальной скорости, $t = 1..N$ – время, N – заданное пользователем количество итераций алгоритма; и рассмотреть значение заданного функционала $p(n, X_u, x_0, v_0, x(t), v(t), v_{max}) = A \in R^3$, где n – количество автомобилей, X_u – маршрут, включающий продольную форму дороги и конфигурацию полос, доступных для перестроения, x_0 – начальные координаты автомобилей, v_0 – начальные скорости автомобилей, $x(t)$ – закон изменения координаты автомобилей, рассмотренный ранее, $v(t)$ – закон изменения скоростей автомобилей, рассмотренный ранее, v_{max} – задаваемое пользователем ограничение скорости, A – набор следующих элементов: 1) количество перестроений; 2) отрицательная величина пропускной способности дороги; 3) отрицательная величина средней скорости автомобиля; который описывает работу алгоритма транспортного потока.

1. Алгоритм косяка рыб

С момента создания алгоритма косяка рыб было предложено множество модификаций, улучшающих его работу. Рассмотрим оригинальную версию алгоритма, представленную на сайте [21].

У каждой рыбы есть вес W_i , положение в пространстве \vec{x}_i . Новая координата на следующем шаге рассчитывается на основании этих значений для всего косяка рыб на предыдущем шаге.

Вес обновляется каждый шаг на величину, зависящую от изменения значения оптимизируемой функции за шаг.

$$W_i(t+1) = W_i(t) + \frac{\Delta f}{\max(|\Delta f|)}$$

$$\Delta f = f[\vec{x}_i(t)] - f[\vec{x}_i(t-1)],$$

где $W_i(t) \in [1, w_scale]$ – вес i -ой рыбы, Δf – изменение целевой функции между предыдущей и новой позицией, $\max(|\Delta f|)$ – максимум из абсолютных значений целевой функции среди всех рыб в косяке.

При запуске алгоритма все рыбы инициализируются со значением веса, равным $\frac{w_scale}{2}$, в случайных координатах в заданном интервале.

Перемещение в пространстве за шаг (скорость рыбы) складывается из 3-х составляющих:

1) индивидуальное движение, задающееся случайно:

$$\vec{x}_i(t+1) = \vec{x}_i(t) + U(-1,1) \text{step}_{\text{ind}}(t)$$

$$\text{step}_{\text{ind}}(t+1) =$$

$$= \text{step}_{\text{ind}}(t) - \frac{\text{step}_{\text{ind}}^{\text{inicial}} - \text{step}_{\text{ind}}^{\text{final}}}{\text{iterations}},$$

где $\vec{x}_i(t)$ и $\vec{x}_i(t+1)$ представляют позиции рыбы до и после применения индивидуального оператора движения, $U(-1,1)$ – равномерное распределение случайных чисел от -1 до 1, $\text{step}_{\text{ind}}(t)$ – это параметр, который определяет максимальное смещение для этого шага. Новая позиция $\vec{x}_i(t+1)$ обновляется, только если изменение позиции приведет к улучшению целевой функции рыбы, иначе остается предыдущее значение $\vec{x}_i(t+1) = \vec{x}_i(t)$. Iterations – это количество взаимодействий / итераций, используемых в моделировании;

2) коллективно-инстинктивное движение, направленное к наиболее тяжелым (успешным) рыбам в косяке:

$$\vec{I}(t) = \frac{\sum_{i=1}^N \Delta \vec{x}_i \Delta f_i}{\sum_{i=1}^N \Delta f_i}$$

$$\Delta \vec{x}_i = \vec{x}_i(t) - \vec{x}_i(t-1).$$

\vec{I} представляет взвешенное среднее смещения (перемещения) каждой рыбы, N – количество рыб. После его вычисления по каждой рыбе обновляется позиция по: $\vec{x}_i(t+1) = \vec{x}_i(t) + \vec{I}(t)$;

3) коллективно-волевое движение, собирающее всех рыб в косяк, если вес косяка увеличивается, и рассредоточивающее косяк по пространству, если вес косяка уменьшается.

$$\text{Барицентр: } \vec{B}(t) = \frac{\sum_{i=1}^N \Delta \vec{x}_i W_i(t)}{\sum_{i=1}^N W_i(t)}$$

$$\vec{x}_i(t+1) = \vec{x}_i(t) + \text{step}_{\text{vol}} U(0,1) \frac{(\vec{x}_i(t) - \vec{B}(t))}{DE(\vec{x}_i(t), \vec{B}(t))}$$

$$\vec{x}_i(t+1) = \vec{x}_i(t) - \text{step}_{\text{vol}} U(0,1) \frac{(\vec{x}_i(t) - \vec{B}(t))}{DE(\vec{x}_i(t), \vec{B}(t))}$$

$$\text{step}_{\text{vol}}(t+1) =$$

$$= \text{step}_{\text{vol}}(t) - \frac{\text{step}_{\text{vol}}^{\text{inicial}} - \text{step}_{\text{vol}}^{\text{final}}}{\text{iterations}},$$

где step_{vol} определяет размер максимального смещения, полученного с использованием этих формул, рекомендуется в два раза больше step_{ind} . $DE(\vec{x}_i(t), \vec{B}(t))$ – евклидово расстояние между позицией рыбы i -ой и барицентром.

Возьмем данный алгоритм за основу нового алгоритма транспортного потока.

2. Алгоритм транспортного потока

В работе [22] была предложена модель транспортного потока, основанная на потенциале действия. Автомобили были представлены частицами, притягивающимися или отталкивающимися друг от друга в зависимости от расстояния между ними. Ограничившись рассмотрением только автопилотируемых транспортных средств, поведение которых строго задано правилами, обобщим эту модель на трехмерное пространство и представим в виде алгоритма.

Гиперпараметры – это параметры алгоритма, задаваемые перед запуском и неизменные в процессе его работы, в отличие от такого параметра, как скорость каждого агента, меняющегося на каждом шаге. Гиперпараметрами алгоритма транспортного потока являются: n – количество агентов; R – радиус кругового трека; V_{\max} – ограничение скорости; ΔT – шаг времени; $d_{\text{станд.}}$ – стандартная дистанция; $k_{\text{отт.}}$ – коэффициент отталкивания; $k_{\text{прит.}}$ – коэффициент притяжения, γ – коэффициент, уменьшающий влияние позади идущего автомобиля. Зависимость от формы или протяженности маршрута в алгоритме не учитывается.

Пусть даны агенты, с начальной скоростью, положением, движущиеся по одному маршруту. Агент – это некоторая сущность, система, которая обладает состоянием и поведением (имеет параметры и функции, их изменяющие), а также свойствами существовать и объединять, необходимыми для взаимодействия с внешней средой. Маршрут – набор коридоров одинаковой формы (для простоты используются параллелепипеды), перестроение возможно только в соседние 4 коридора (Рис. 1): левый, верхний, правый и нижний. Перестроение происходит за расчетный шаг.

Каждый агент имеет следующие параметры: x – координата; v – скорость; i_+ – впереди идущий агент по коридору и i_- – позади идущий, а также впереди и позади идущие агенты для смежных коридоров, на которые возможно перестроение. На каждом шаге рассчитывается ускорение (1), (2), в соответствии с которым изменяется скорость (3), а затем координата агента по формуле $x_{i,\tau+1} = x_{i,\tau} + v_{i,\tau+1}\Delta t$.

Формулы для расчета ускорения на каждом новом шаге:

$$a_{i-} = \begin{cases} \frac{k_{\text{отт.}}}{r l_{i-}}, & \text{если } l_{i-} < d_{\text{станд.}}; \\ 0, & \text{если } l_{i-} \geq d_{\text{станд.}}; \end{cases} \quad (1)$$

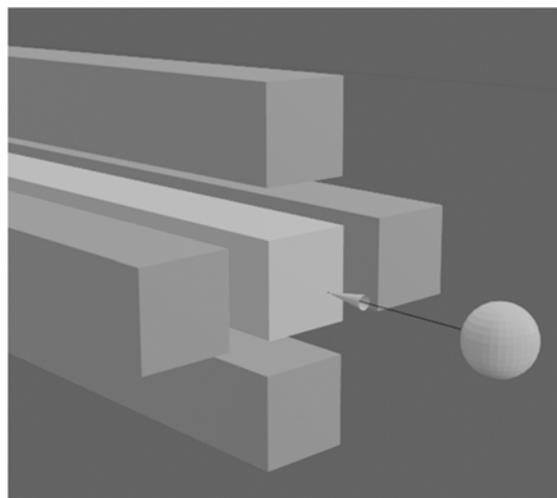


Рис. 1. Коридоры, доступные агенту для перестроения

$$a_{i+} = \begin{cases} \frac{k_{\text{отт.}}}{l_{i+}}, & \text{если } l_{i+} < d_{\text{станд.}}; \\ k_{\text{прит.}} * l_{i+}, & \text{если } l_{i+} \geq d_{\text{станд.}}, \end{cases} \quad (2)$$

где a_{i-} – ускорение i -го агента, которое вызывает позади идущий агент;

$k_{\text{отт.}}$ – коэффициент отталкивания;

l_{i-} – расстояние до позади идущего агента для i -ого агента;

γ – коэффициент, уменьшающий влияние позади идущего автомобиля;

$d_{\text{станд.}}$ – стандартная дистанция;

a_{i+} – ускорение i -го агента, которое вызывает впереди идущий агент;

l_{i+} – расстояние до впереди идущего агента для i -ого агента;

$k_{\text{прит.}}$ – коэффициент притяжения.

На Рис. 2 наглядно представлены индексы приведенных в уравнениях переменных.

Формула расчета скорости агента на новом шаге:

$$v_{i,\tau+1} = \min(v_{i,\tau} + (a_{i-} + a_{i+})\Delta t, v_{\max}), \quad (3)$$

где $v_{i,\tau+1}$ – скорость i -ого агента в момент времени $\tau + 1$;

$$\min(a, b) = \begin{cases} a, & \text{если } a < b; \\ b, & \text{если } a \geq b; \end{cases}$$

$v_{i,\tau}$ – скорость i -ого агента в момент времени τ ;

v_{\max} – ограничение скорости.

Всем агентам при создании задается начальная скорость и нулевое ускорение. Также агенту задается начальное положение. Так как агенты представляют собой окружности (сферы) с заданным радиусом, то при начальном размещении (как и в дальнейшем) обеспечивается непересечение

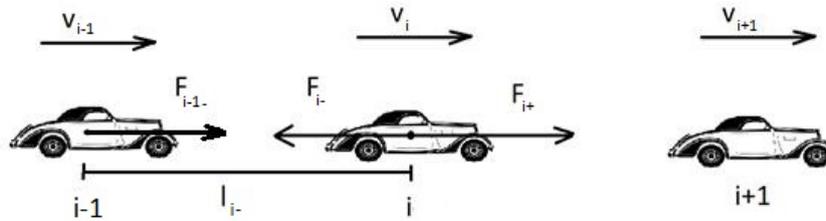


Рис. 2. Действие сил в рамках одной полосы

границ других агентов. Исходный рой размещается в заданном месте с заданным количеством агентов, обеспечивающим нужную плотность.

Алгоритм, по которому изменяются координаты агентов:

1. Расчет скорости для каждого агента:
 - a. Расчет ускорения, из предположения, что агенты будут продолжать двигаться по своей полосе.
 - b. Если ускорение отрицательное, проверка ближайших полос на возможность перестроения на них.
 - c. Расчет скорости для следующего шага.
 - d. Расчет новых положений агентов.
2. Проверка столкновений.
3. Изменение положения агентов.

Условием останова является выполнение заданного пользователем количества итераций. Подробнее алгоритм представлен на Рис. 3.

Как можно видеть из блок-схемы, в алгоритме есть цикл по j до N , в который вложен цикл по i до n . В последнем цикле все операции требуют $c \cdot n$ времени, где c – константа, которой можно пренебречь. Следовательно, асимптотическая сложность алгоритма $O(n^2N)$, где n – количество особей, N – количество итераций.

Проверка возможности перестроения и его осуществление описаны на Рис. 4.

Выбор полосы из возможных свободных для перестроения осуществляется последовательно. Кроме того, само перестроение происходит с вероятностью 30 % (в случае отрицательного ускорения) и после этого на фиксированное количество времени для перестроившегося агента действует запрет на дальнейшие перестроения.

Изменение положения агентов за один шаг алгоритма представлено на Рис. 5.

Сферы, изображающие агентов роевого алгоритма, движутся вдоль оси x . У однотонных агентов одинаковая скорость, перестроений не произошло. У отмеченного сеткой агента скорость больше, чем у впереди идущего агента, и

имелось свободное пространство в смежном левом коридоре, в результате чего произошло перестроение.

3. Обсуждение

Описан алгоритм, цель которого – в задании такого поведения агентов, при котором отсутствуют столкновения. Представлена имитационная модель, поскольку аналитическое решение затруднительно.

Агентом является движущееся по заданному маршруту автопилотируемое транспортное средство или летательный аппарат, у которого есть ограниченный набор возможных действий: ускорение, замедление, перестроение между полосами (коридорами). Поведение агента основывается на поступающей из окружающей среды информации о расстоянии до соседних агентов и скорости впереди идущих агентов на полосах (коридорах), на которые возможно перестроение.

Обычно цель у роевого интеллекта – численная оптимизация, но в нашем случае цель транспортного потока – это движение без столкновений. Рассмотрим несколько возможных критериев оптимизации для описанной задачи движения:

1. Минимальное количество перестроений.
2. Максимальная пропускная способность дороги (коридоров).
3. Максимальная средняя скорость агента.
4. Минимальное количество (резких) изменений скорости – торможений и остановок.

Приведем примеры работы роевого алгоритма, демонстрирующие изменения данных критериев. При использовании полученных в работе [22] коэффициентов притяжения и отталкивания, подходящих для моделирования автомобилей на дороге с несколькими полосами в реальном мире, будет наблюдаться следующее поведение. Если плотность позволяет, то агенты распределяются по пространству таким образом, чтобы дистанция

между ними не вызывала торможения, и достигают максимальной скорости. В противном случае

(если плотность велика) агенты непрерывно перестраиваются, стараясь избежать торможения.

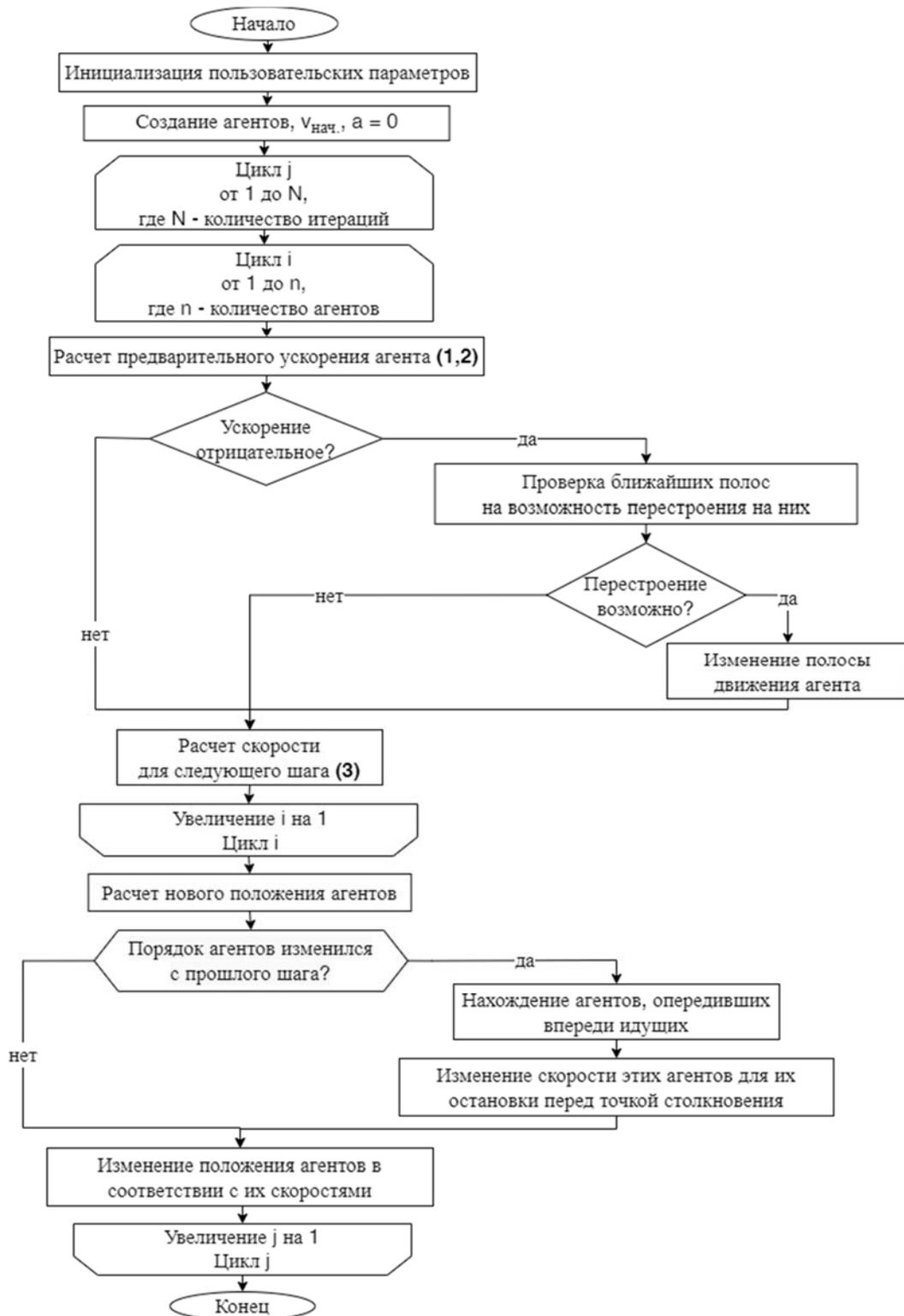


Рис. 3. Роевой алгоритм транспортного потока

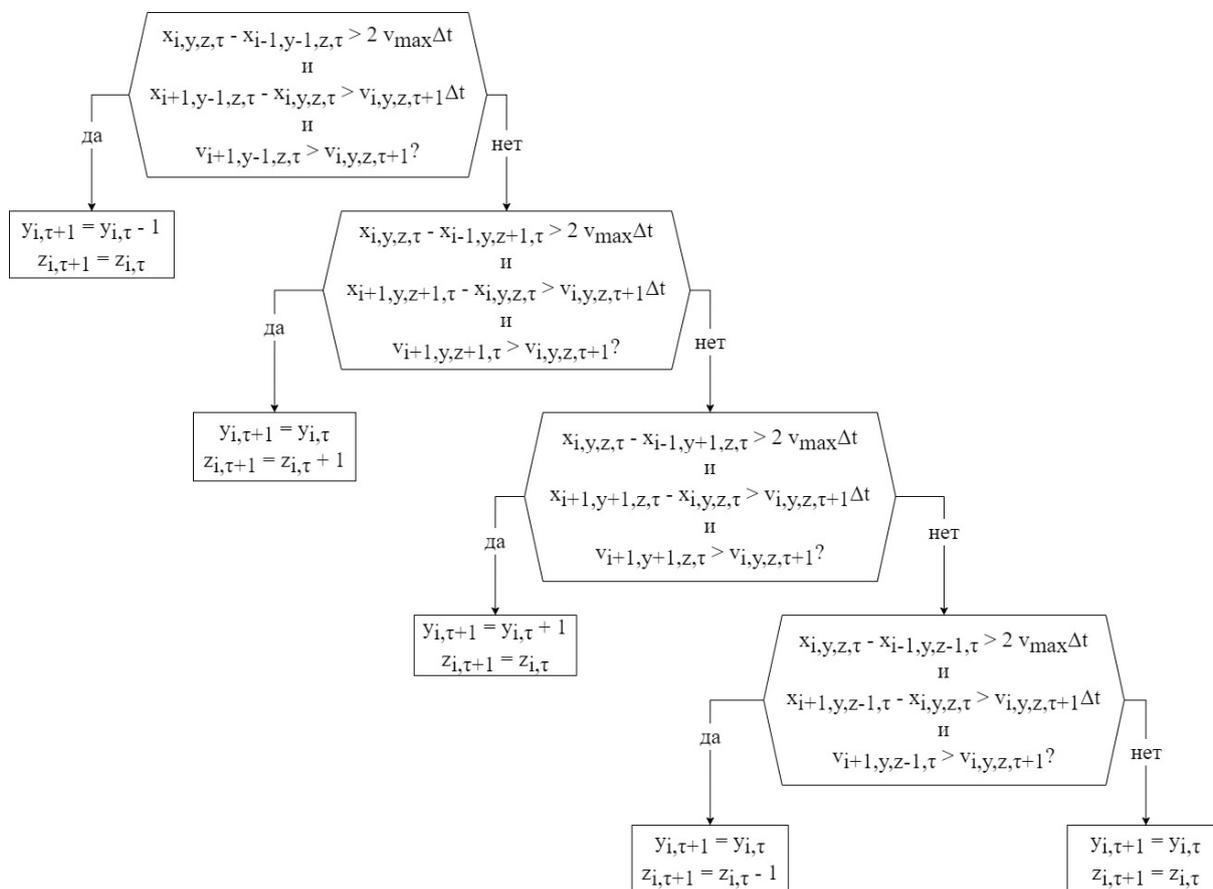


Рис. 4. Определение новой координаты в результате перестроения

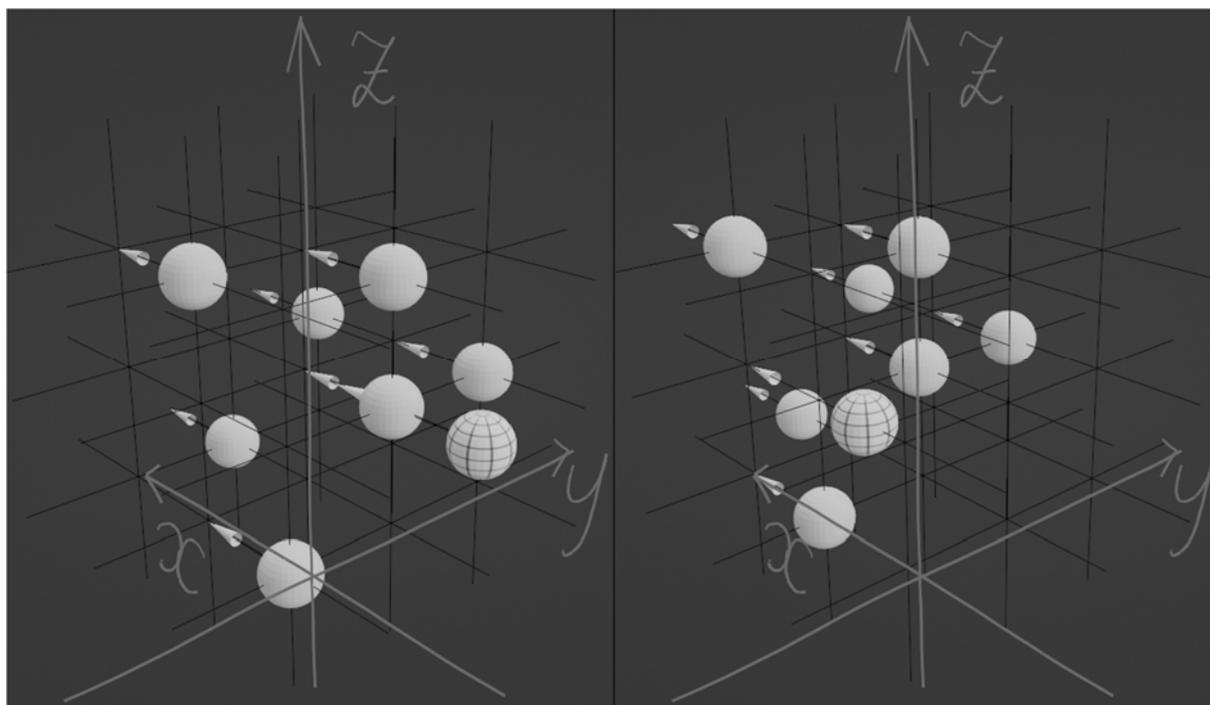


Рис. 5. Перемещение агентов в пространстве с перестроением

Свободный поток при большой плотности и заданных параметрах невозможен исходя из правил алгоритма: если агентов расставить на равных расстояниях и запустить их движение с максимальной скоростью, то из-за малых дистанций они начнут снижать скорость, а затем двигаться с заторами или вовсе остановятся. Избегать снижения скорости на малой дистанции опасно из-за возможных непредвиденных обстоятельств, вроде случайно появившегося на дороге пешехода.

Для описанного частного случая модели были проведены эксперименты для оценки изменения рассматриваемых параметров в результате взаимодействия агентов. Модель транспортного потока была реализована в виде кода на языке Python3 и запускалась на компьютере с 2 процессорами Intel(R) Xeon(R) Silver 4214R CPU 2.40 GHz, оперативной памятью 124 Гб, операционной системой Windows 10 Pro в WSL2 – подсистеме Windows для Linux.

В данном случае набор коридоров (маршрут) представлен двумя полосами, составляющими окружность радиусом 360 м; количество агентов – 220; максимальная скорость 6 м/с; коэффициент притяжения 0.301, отталкивания – 10.6

(коэффициенты, найденные экспериментально [22] для моделирования реального транспортного потока). Радиус агентов равен 2 м. Начальная скорость агентов 6 м/с (около 20 км/ч). Время запрета на дальнейшие перестроения после совершенного перестроения равно 3 с. Коэффициент, уменьшающий влияние позади идущего автомобиля, r – 100. Результаты представлены на графиках на Рис. 6.

Количество перестроений заметно падает через 3 минуты после начала моделирования, и затем его значение сохраняется около 20 перестроений в секунду. Фундаментальная диаграмма показывает интенсивность потока, которая при практически постоянной плотности составляет 6 автомобилей в секунду (количество агентов, проезжающих за ед. времени, – это и есть пропускная способность). Средняя максимальная скорость равна заданной максимальной допустимой скорости.

Рассмотрим, как поведет себя модель, если отойти от реально возможных для автомобилей скоростей. Изменения тех же параметров с ограничением максимальной скорости 60 м/с представлены на Рис. 7.

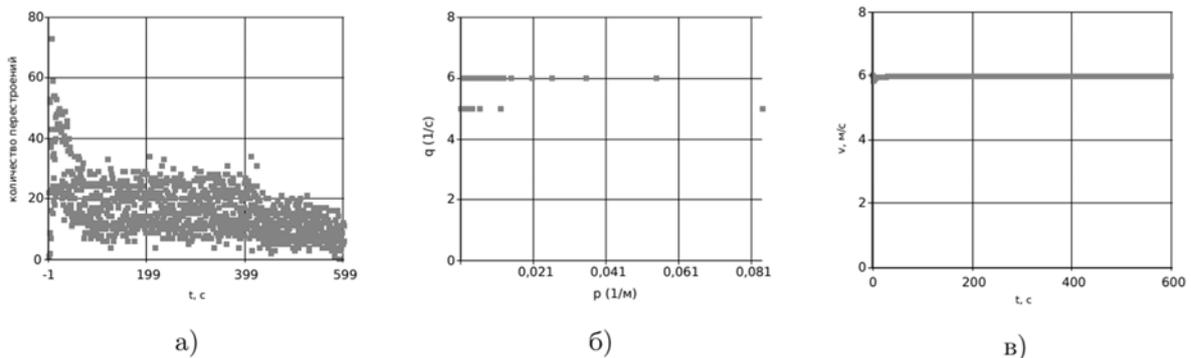


Рис. 6. Экспериментальный расчет. 220 агентов, ограничение скорости 6 м/с
 а) количество перестроений, б) фундаментальная диаграмма, в) средняя скорость

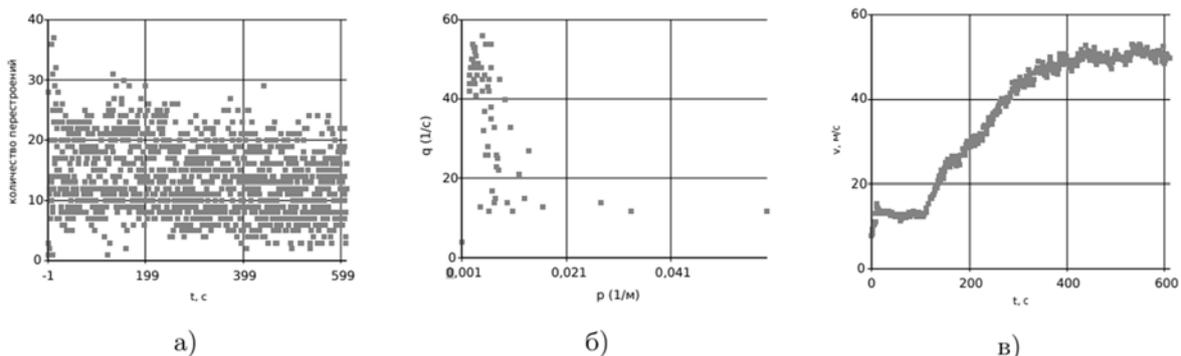


Рис. 7. Экспериментальный расчет. 220 агентов, ограничение скорости 60 м/с
 а) количество перестроений, б) фундаментальная диаграмма, в) средняя скорость

Количество перестроений между полосами – от 5 до 25 за 1 с. На фундаментальной диаграмме при малой плотности наблюдается свободный поток, а затем правее появляются точки, отмечающие заторное движение (при увеличивающейся плотности наблюдается одинаковая интенсивность). Значение средней скорости агентов возрастает до 55 м/с за 7 мин, после чего сохраняется постоянным.

Рассмотрев результаты увеличения ограничения максимальной скорости, перейдем к изменению следующих гиперпараметров – коэффициентов притяжения и отталкивания. Можно добиться максимизации пропускной способности, определив v_{max} и $k_{прит.}$ максимальными значениями, $k_{отт.}$ – минимальным и оставив обработку столкновений на экстренный вариант. В таком случае адаптивность к возмущениям в реальности будет несколько проблематична. Параметры запуска модели, представленной на Рис. 8: $v_{max} = 60$ м/с, $R = 360$, $n = 460$, $k_{прит.} = 3000$, $k_{отт.} = 0.0001$, $r = 100$.

Количество перестроений меньше 5 за секунду. Фундаментальная диаграмма так же демонстрирует свободный поток и заторное движение, но максимальная интенсивность, как и максимальная плотность, выше. Пропускная способность, даже при движении с пробками, выше, чем на графике (Рис. 7, б) при свободном движении. Скорость за 3 мин поднимается до максимального значения.

У транспортного потока есть 2 качественно наблюдаемых состояния: свободный поток и заторное движение (пробки). Кроме того, в нашей модели, над которой проводятся вычислительные эксперименты (замкнутый круг), возможна полная остановка. Определим, при каких значениях параметров (коэффициентов притяжения, отталкивания и плотности) транспортный поток

будет находиться в том или ином состоянии (Рис. 9). Это позволит определить рекомендуемые границы параметров при дальнейшем использовании роевого алгоритма. Экспериментальные расчеты проводились с двухполосным движением агентов по кругу в течение 10 мин, после чего снимались показания, характеризующие состояние транспортного потока.

Транспортный поток продолжает движение до тех пор, пока плотность не станет равной единице. Когда коэффициент отталкивания равен 0, наблюдается свободный поток (возможно, нулевая скорость после экстренного торможения не успевает фиксироваться). Чем выше коэффициент отталкивания, тем раньше при меньших значениях плотности появляется заторный поток. На близком, но не равном нулю коэффициенте притяжения свободный поток наблюдается при большей плотности, чем при большем значении коэффициента притяжения. После увеличения значения коэффициента отталкивания выше сотни изменения коэффициента притяжения перестают влиять на результат.

Если коэффициенты отталкивания и притяжения сильно отличаются от тех, что вызывают подобное реальности движение, то нагрузка избегания столкновений переходит с потенциала действия на экстренный механизм. Это может помочь добиться высоких показателей эффективности потока агентов, но требует выполнения большего количества операций с большей затратой ресурсов и практически свидетельствует о столкновениях в реальном мире.

Более подробное рассмотрение поведения модели, изменения значений критериев при различных значениях параметров требует дополнительных исследований и выходит за рамки данной статьи.

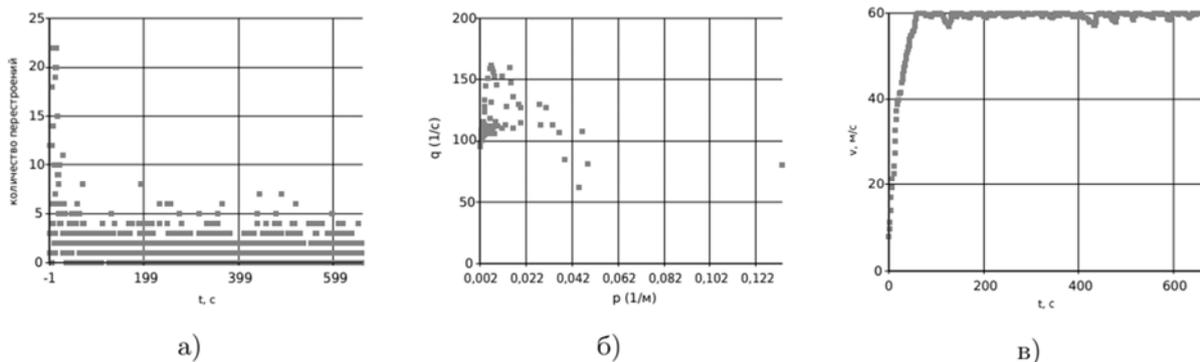


Рис. 8. Экспериментальный запуск. 460 агентов, ограничение скорости 60 м/с
 а) количество перестроений, б) фундаментальная диаграмма, в) средняя скорость

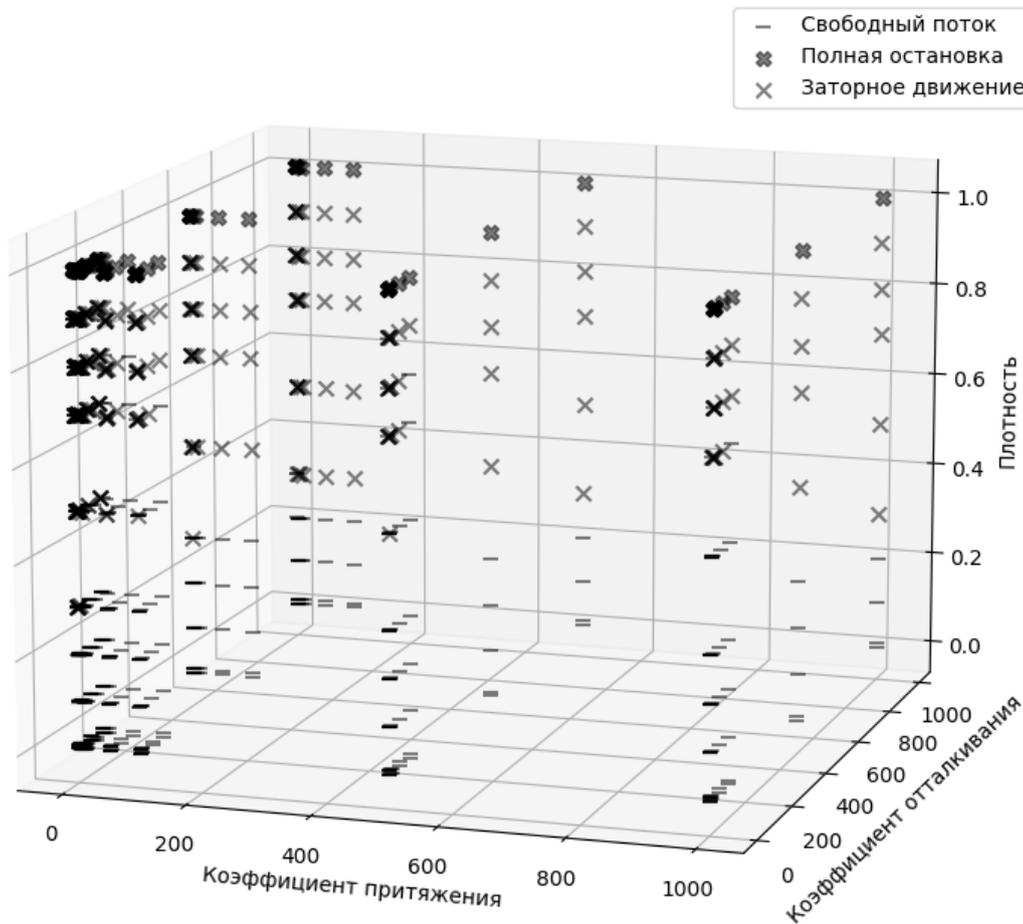


Рис. 9. Границы гиперпараметров. 1000 точек по выбранным значениям

Из сказанного выше следует, что через какое-то время после запуска модели, если позволяют плотность и коэффициенты отталкивания / притяжения, образуется стационарный свободный поток, который либо основан на притяжении и отталкивании, либо на экстренных торможениях. В противном случае, при отсутствии возможности свободного потока будут наблюдаться движение с пробками или полная остановка. В процессе работы алгоритма, если движение потока сохраняется, значения рассматриваемых критериев оптимизации изменяются от худшего к лучшему.

Можно встраивать этот алгоритм как улучшение в другие алгоритмы для избегания повторений расчетов. Вопрос, будут ли использование коэффициентов отталкивания и притяжения и проверка на столкновение проще, чем лишние шаги повторяющихся (или сохраняемых в памяти) расчетов для агента, требует дальнейшего рассмотрения.

Заключение

В роевых алгоритмах не рассматривается вопрос столкновений агентов, поскольку размещение нескольких агентов одновременно в одной и той же точке пространства несущественно для численной оптимизации. Кроме того, в некоторых алгоритмах у них могут быть различные дополнительные параметры, что позволяет получать новые значения для каждого из агентов. Но, так как физические объекты имеют объем и не могут проходить друг сквозь друга, решение задачи движения без столкновений для роя агентов является имеющим смысл, если возникнет необходимость запустить численную оптимизацию в реальности.

Предложен роевой алгоритм, позволяющий обеспечить движение агентов без столкновений. Описанный алгоритм может использоваться для решения задачи перемещения из пункта А в пункт Б по заданному маршруту. В процессе

агенты будут избегать возможных столкновений, следуя простым правилам ускорения или торможения в зависимости от расстояния до впереди идущего агента, или, в особых случаях, заранее сбрасывая скорость.

Предложенный новый оптимизационный подход может найти практическое применение в создании новых алгоритмов управления транспортными потоками, с минимальной нагрузкой на вычислительные системы и отдельные элементы системы.

В дальнейшем планируется разработать серию нейронных сетей, которая будет включать в себя разработанный алгоритм. Многоагентное представление хорошо распараллеливается в вычислительных системах (агенты просты и независимы), поэтому может быть получен набор простейших нейронных сетей, реализующих поведение транспортного потока.

Литература

- Кузнецов А.В. Краткий обзор многоагентных моделей // Управление большими системами: сборник трудов. 2018. № 71. С. 6–44.
- Кравченко П.С., Омарова Г.А. Микроскопические математические модели транспортных потоков. Аналитический обзор // Проблемы информатики. 2014. № 1 (22). С. 24–31.
- Четверушкин Б.Н. и др. Макро- и микроскопические модели для описания движения автотранспорта на многополосных магистралях // Труды МФТИ. 2010. № 4.
- Ma Z. at al. Performance assessment and exhaustive listing of 500+ nature-inspired metaheuristic algorithms // Swarm and Evolutionary Computation. 2023. V. 77. P. 101248.
- Himani K., Girdhar A. Swarm Intelligence and Flocking Behavior // IJCA Proceedings on International Conference on Advancements in Engineering and Technology (ICAET 2015). 2015. V. 10. P. 9–12.
- Кулиев Э.В. и др. Адаптивный алгоритм стаи серых волков для решения задач проектирования // Известия ЮФУ. Технические науки. 2017. № 7 (192). С. 28–38.
- Дюличева Ю.Ю. Алгоритмы роевого интеллекта и их применение для анализа образовательных данных // Открытое образование. 2019. № 23 (5). С. 33–43.
- Алалван А.Р.Д., Лихута В. И., Кислов Л. С. Некоторые аспекты использования алгоритма кукушки в обучении и управлении роботами // Наука. Информатизация. Технологии. Образование : Материалы XII международной научно-практической конференции. 2019. С. 402–408.
- Лебедев Б.К., Лебедев О.Б., Лебедева Е.М. Распределение ресурсов на основе гибридных моделей роевого интеллекта // Научно-технический вестник информационных технологий, механики и оптики. 2017. № 17(6). С. 1063–1073.
- Carmelo J. A. at al. A novel search algorithm based on fish school behavior // 2008 IEEE International Conference on Systems, Man and Cybernetics. 2008. P. 2646–2651.
- Частикова В.А., Дружинина М.А., Кекало А.С. Исследование эффективности алгоритма поиска косяком рыб в задаче глобальной оптимизации // Современные проблемы науки и образования. 2014. № 4.
- Chen J., Cai H., Wang W. A new metaheuristic algorithm: car tracking optimization algorithm // Soft Comput. 2018. V. 22. P. 3857–3878.
- Savsani P., Savsani V. Passing vehicle search (PVS): A novel metaheuristic algorithm // Applied Mathematical Modelling. 2016. V. 40. No 5–6. P. 3951–3978.
- Zhong Y., Lin J., Wang L., Zhang H. Discrete comprehensive learning particle swarm optimization algorithm with Metropolis acceptance criterion for traveling salesman problem // Swarm and Evolutionary Computation. 2018. V. 42. P. 77–88.
- Мартынова Ю.А., Мартынов Я.А. Формализация задачи организации маршрутных сетей городского пассажирского транспорта // Интернет-журнал «НАУКОВЕДЕНИЕ». 2014. № 6.
- Santwana P., Anjali M. P., Kedar H. Traffic Management Using Swarm Intelligence and Route Selection Using Android Application // International Journal of Engineering and Innovative Technology (IJEIT). 2015. V. 5. No 6. P. 59–63.
- Teodorović D. Swarm intelligence systems for transportation engineering: Principles and applications // Transportation Research Part C. 2008. V. 16. No 6. P. 651–667.
- Walker P., Saman A. A., Chakraborty N., Lewis M., Sycara K. Human Control of Robot Swarms with Dynamic Leaders // IEEE International Conference on Intelligent Robots and Systems. 2014. P. 1108–1113.
- Virágh C. at al. Flocking algorithm for autonomous flying robots // Bioinspiration & Biomimetics. 2014. V. 9. No. 2. P. 025012.
- Altshuler Y. Recent Developments in the Theory and Applicability of Swarm Search // Entropy. 2023. V. 25. No 5. P. 710.
- Operators and functioning of FSS // Fish School Search (FSS Algorithm): сайт. – URL: <https://fbln.me/fss/operators/operators/> (дата обращения: 21.04.2023).
- Бобровская О.П., Гавриленко Т.В., Галкин В.А. Модель транспортного потока, основанная на взаимодействии частиц с потенциалом действия // Вестник КРАУНЦ. Физ.-мат. науки. 2022. Т. 40. № 3. С. 72–87.

Бобровская Ольга Павловна. Инженер, Сургутский филиал Федерального научного центра «Научно-исследовательский институт системных исследований РАН». Ассистент, Сургутский государственный университет. Области исследований: вычислительная математика, информационные технологии. E-mail: o-bobrovskaya@mail.ru (Ответственный за переписку).

Гавриленко Тарас Владимирович. Кандидат технических наук, доцент. Доцент кафедры автоматизированных систем обработки информации и управления, Сургутский государственный университет. Заместитель директора, Сургутский филиал Федерального научного центра «Научно-исследовательский институт системных исследований РАН». Области исследований: вычислительная математика, информационные технологии. E-mail: taras.gavrilenko@gmail.com.

Галкин Валерий Алексеевич. Доктор физико-математических наук, профессор. Профессор кафедры прикладной математики, БУ ВО «Сургутский государственный университет». Директор, Сургутский филиал Федерального научного центра «Научно-исследовательский институт системных исследований РАН». Области исследований: вычислительная математика, информационные технологии. E-mail: val-gal@yandex.ru.

Swarm Intelligence Algorithm of Traffic

O. P. Bobrovskaya^{1,2}, T. V. Gavrilenko^{1,2}, V. A. Galkin^{1,2}

¹ Surgut State University, Surgut, Russian Federation

² Surgut Branch of Federal Scientific Center "Research Institute for System Research of RAS", Surgut, Russia

Abstract. The problem of modeling the routes of self-driving vehicles in a traffic flow in which there are no collisions is being solved. A new swarm algorithm based on a microscopic model of traffic flow is proposed, which ensures the movement of agents without collisions. Changes in several optimality criteria during the operation of the algorithm are considered, such as: average speed of agents, throughput, number of lane changes. The boundaries of the effective values of the hyperparameters of the algorithm are estimated. At certain density parameters and push/pull coefficients in the traffic flow, free flow and an improvement in the values of the optimization criteria are observed.

Keywords: swarm intelligence, swarm algorithm, microscopic model of traffic flow, agent, fish school algorithm.

DOI 10.14357/20718594230406 EDN SJCXCN

References

1. Kuznecov A.V. Kratkij obzor mnogoagentnyh modelej [Brief overview of multi-agent models]. *Upravlenie bol'shimi sistemami: sbornik trudov* [Large Systems Management: Proceedings]. 2018. V. 71. P. 6–44.
2. Kravchenko P.S., Omarova G.A. Mikroskopicheskie matematicheskie modeli transportnyh potokov. Analiticheskij obzor [Microscopic mathematical models of traffic flows. Analytical review] // *Problemy informatiki* [Problems of Informatics]. 2014. V. 1. No 22. P. 24–31. URL: <https://cyberleninka.ru/article/n/mikroskopicheskie-matematicheskie-modeli-transportnyh-potokov-analiticheskij-obzor> (accessed January 24, 2023).
3. Chetverushkin B. N., Trapeznikova M. A., Furmanov I. R., Churbanov N. G. Makro- i mikroskopicheskiye modeli dlya opisaniya dvizheniya avtotransporta na mnogopolosnykh magistralyakh [Macro- and microscopic models for describing the movement of vehicles on multi-lane highways] // *Trudy MFTI* [Trudy MIPT]. 2010. V. 4. URL: <https://cyberleninka.ru/article/n/makro-i-mikroskopicheskie-modeli-dlya-opisaniya-dvizheniya-avtotransporta-na-mnogopolosnykh-magistralyakh> (accessed January 31, 2023).
4. Ma Z. at al. Performance assessment and exhaustive listing of 500+ nature-inspired metaheuristic algorithms. *Swarm and Evolutionary Computation*. 2023. V. 77. P. 101248.
5. Himani K., Girdhar A. Swarm Intelligence and Flocking Behavior. *IJCA Proceedings on International Conference on Advancements in Engineering and Technology (ICAET 2015)*. 2015. V. 10. P. 9–12.
6. Kuliev E.V., Shcheglov S.N., Pantelyuk E.A., Kulieva N.V. Adaptivnyj algoritm stai seryh volkov dlya resheniya zadach proektirovaniya [An adaptive algorithm for a pack of gray wolves for solving design problems] // *Izvestiya YUFU. Tekhnicheskie nauki* [Izvestiya SFU. Technical science]. 2017. V. 7. No. 192. URL: <https://cyberleninka.ru/article/n/adaptivnyy-algoritm-stai-seryh-volkov-dlya-resheniya-zadach-proektirovaniya> (accessed January 31, 2023).
7. Dyulichева Yu.Yu. Algoritmy roevogo intellekta i ih primeneniye dlya analiza obrazovatel'nyh dannyh [Swarm intelligence algorithms and their application for the analysis of educational data] // *Otkrytoe obrazovanie* [Open education]. 2019. V. 23. No 5. P. 33-43.
8. Alalvan A.R.D., Lihuta V. I., Kislov L. S. Nekotorye aspekty ispol'zovaniya algoritma kukushki v obuchenii i upravlenii robotami [Some Aspects of Using the Cuckoo Algorithm in Teaching and Controlling Robots] // *Nauka. Informatizaciya. Tekhnologii. Obrazovanie* : Materialy XII mezhdunarodnoj nauchno-prakticheskoy konferencii [Science. Informatization. Technologies. Education: Proceedings of the XII International Scientific and Practical Conference]. 2019. P. 402–408.
9. Lebedev B.K., Lebedev O.B., Lebedeva E.M. Raspredeleeniye resursov na osnove gibridnyh modelej roevogo intellekta [Resource distribution based on hybrid models of swarm intelligence] // *Nauchno-tekhnicheskij vestnik informacionnyh tekhnologij, mekhaniki i optiki* [Scientific and technical bulletin of information technologies, mechanics and optics]. 2017. V. 17(6). P. 1063-1073.
10. Carmelo J. A. at al. A novel search algorithm based on fish school behavior. 2008 IEEE International Conference on Systems, Man and Cybernetics, Singapore. 2008. P. 2646–2651.

11. Chastikova V.A., Druzhinina M.A., Kekalo A.S. Issledovanie effektivnosti algoritma poiska kosyakom ryb v zadache global'noj optimizacii [Investigation of the efficiency of the fish school search algorithm in the problem of global optimization] // *Sovremennye problemy nauki i obrazovaniya* [Modern problems of science and education]. 2014. V. 4. URL: <https://science-education.ru/ru/article/view?id=14142> (accessed October 21, 2022).
12. Chen J., Cai H., Wang W. A new metaheuristic algorithm: car tracking optimization algorithm. *Soft Comput.* 2018. V. 22. P. 3857–3878.
13. Savsani P., Savsani V. Passing vehicle search (PVS): A novel metaheuristic algorithm. *Applied Mathematical Modelling.* 2016. V. 40. No 5–6. P. 3951–3978.
14. Zhong Y., Lin J., Wang L., Zhang H. Discrete comprehensive learning particle swarm optimization algorithm with Metropolis acceptance criterion for traveling salesman problem. *Swarm and Evolutionary Computation.* 2018. V. 42. P. 77–88.
15. Martynova Yu.A., Martynov Ya.A. Formalizaciya zadachi organizacii marshrutnyh setej gorodskogo passazhirskogo transporta [Formalization of the task of organizing route networks of urban passenger transport]. *Internet- journal «NAUKOVEDENIE».* 2014. V. 6.
16. Santwana P., Anjali M. P., Kedar H. Traffic Management Using Swarm Intelligence and Route Selection Using Android Application. *International Journal of Engineering and Innovative Technology (IJEIT).* 2015. V. 5. No 6. P. 59–63.
17. Teodorović D. Swarm intelligence systems for transportation engineering: Principles and applications. *Transportation Research Part C.* 2008. V. 16. No 6. P. 651–667.
18. Walker P., Saman A. A., Chakraborty N., Lewis M., Sycara K. Human Control of Robot Swarms with Dynamic Leaders. *IEEE International Conference on Intelligent Robots and Systems.* 2014. P. 1108–1113.
19. Virágh C. et al. Flocking algorithm for autonomous flying robots. *Bioinspiration & Biomimetics.* 2014. V.9. No 2. P. 025012.
20. Altshuler Y. Recent Developments in the Theory and Applicability of Swarm Search. *Entropy.* 2023 V. 25. No 5. P. 710.
21. Operators and functioning of FSS // Fish School Search (FSS Algorithm): сайт. – URL: <https://fbln.me/fss/operators/operators/> (accessed April 21, 2023).
22. Bobrovskaya O. P., Gavrilenko T.V., Galkin V. A. Model' transportnogo potoka, osnovannaya na vzaimodeystvii chastits s potentsialom deystviya [Transport flow model based on interaction of particles with action potential] // *Vestnik KRAUNC. Fiz.-mat. Nauki* [Bulletin of KRAUNC. Phys.-math. Sciences]. 2022. V. 40. No 3. P. 72–87.

Bobrovskaya Olga P. Engineer, Surgut Branch of Federal Scientific Center “Research Institute for System Research of RAS”. Teaching assistant, Surgut State University. Research areas: computational mathematics, information technology. E-mail: o-bobrovskaya@mail.ru.

Gavrilenko Taras V. Candidate of Technical Sciences, docent. Assistant professor, Department of Automated Information Processing and Management Systems, Surgut State University. Deputy Director, Surgut Branch of Federal Scientific Center “Research Institute for System Research of RAS”. Research areas: computational mathematics, information technology. E-mail: taras.gavrilenko@gmail.com.

Galkin Valeriy A. Doctor of Physical and Mathematical Sciences, Professor. Professor, Department of Applied Mathematics, Surgut State University. Director, Surgut Branch of Federal Scientific Center “Research Institute for System Research of RAS”. Research areas: computational mathematics, information technology. Email: val-gal@yandex.ru.