ИПМ им.М.В.Келдыша РАН • Электронная библиотека

Препринты ИПМ • Препринт № 23 за 2024 г.

ISSN 2071-2898 (Print)
ISSN 2071-2901 (Online)

А.А. Чечина

Различные аспекты создания алгоритмов для модели транспортных потоков на основе теории клеточных автоматов

Статья доступна по лицензии
Creative Commons Attribution 4.0 International

Рекомендуемая форма библиографической ссылки: Чечина А.А. Различные аспекты создания алгоритмов для модели транспортных потоков на основе теории клеточных автоматов // Препринты ИПМ им. М.В.Келдыша. 2024. № 23. 31 с. https://doi.org/10.20948/prepr-2024-23
https://library.keldysh.ru/preprint.asp?id=2024-23

О р д е н а Л е н и н а
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ
имени М.В. Келдыша
Р о с с и й с к о й а к а д е м и и н а у к

А.А. Чечина

Различные аспекты создания алгоритмов для модели транспортных потоков на основе теории клеточных автоматов

Москва — 2024

Чечина А.А.

Различные аспекты создания алгоритмов для модели транспортных потоков на основе теории клеточных автоматов

Препринт посвящен описанию полного набора алгоритмов для модели на основе клеточных автоматов, разрабатываемой коллективом авторов из ИПМ им. М.В. Келдыша РАН последние несколько лет.

В работе представлены алгоритмы движения автомобилей для различных ситуаций: перестроение, объезд препятствия, проезд перекрестков различных типов, въезды/съезды и т.д. Представлены алгоритмы поведения водителей и их влияние на дорожную ситуацию.

Ключевые слова: моделирование транспортных потоков, модель клеточных автоматов

Chechina Antonina Aleksandrovna

Various aspects of development of algorithms for the traffic flow model based on cellular automata theory

The preprint is devoted to the description of the complete set of algorithms for the model based on cellular automata, developed by the team of authors from KIAM RAS for the last few years.

The paper presents algorithms of vehicle movement in various situations: changing lanes, avoiding obstacles, driving through different types of intersections, entrances/exits, etc. Driver behavior algorithms and their impact on traffic situation are presented.

Key words: traffic flow modeling, cellular automata model

Введение

Клеточные автоматы – подход, хорошо зарекомендовавший себя в самых разных областях науки [1]. Метод зарекомендовал себя как удобный инструмент для описания больших разнородных систем со многими взаимодействующими элементами, поскольку оперирует правилами (условиями вида «если – то»), а не уравнениями. Так как система автомобилей в транспортном потоке и дорожная инфраструктура как раз являются частью сложной и разнородной системы, клеточные автоматы остаются одним из популярных инструментов при моделировании транспортных потоков (см., например, [2-5]).

Препринт посвящен описанию полного набора алгоритмов для модели на основе клеточных автоматов, разрабатываемой коллективом авторов из ИПМ им. М.В. Келдыша РАН последние несколько лет.

В работе представлены алгоритмы движения автомобилей для различных ситуаций: перестроение, объезд препятствия, проезд перекрестков различных типов, въезды/съезды и т.д. Представлены алгоритмы поведения водителей и их влияние на дорожную ситуацию.

Алгоритмы реализованы в виде составной части программного комплекса для высокопроизводительных вычислительных систем [6]. Тестовые расчеты, верифицирующие созданную модель, представлены в [7, 8].

Построение многополосной модели на основе теории клеточных автоматов

В основу разработанной многополосной модели положена одномерная модель Нагеля-Шрекенберга [9], использующая идеологию клеточных автоматов.

Ранее этот подход представлялся наиболее перспективным для детального описания локальных дорожных ситуаций на небольших расстояниях, так как модели являются достаточно гибкими благодаря возможности реализовать взаимодействие водителей как с дорожно-транспортной инфраструктурой, так и между собой без существенных алгоритмических затрат. Однако благодаря существующей в настоящее время вычислительной технике сверхвысокой производительности модели данного типа могут с успехом применяться и для моделирования движения на больших дорожных сетях.

Трасса в этом случае представляет собой двумерную решетку, где количество ячеек в поперечном направлении соответствует числу полос трассы, как показано на рис. 1.

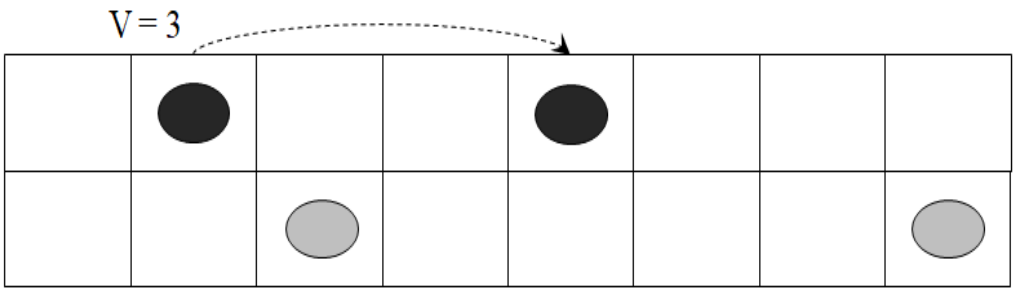


Рис. 1. Расчетная область в обобщенной двумерной микроскопической модели

В многополосной модели разрешены перестроения машин из полосы в полосу и обгоны. Процесс обновления состояний ячеек делится на два шага:

- смена полосы при необходимости и наличии возможности;
- движение вперед по правилам однополосного движения.

Размер ячейки расчетной области равен 7,5 метрам в длину и 3,5 метрам в ширину. Эти параметры соответствуют средней площади, занимаемой легковым автомобилем, стоящим на дороге, с учетом минимального необходимого расстояния до соседних автотранспортных средств – АТС. Ширина ячейки соответствует стандартной ширине полосы автомобильной дороги обычного типа (нескоростная дорога).

В двумерную микроскопическую модель в качестве параметров входят две вероятностные характеристики: вероятность спонтанного снижения скорости из модели Нагеля-Шрекенберга и вероятность остаться в своей полосе в случае, когда перестроение возможно и выгодно водителю. Первый параметр соответствует ситуациям, когда водитель, например, посчитал состояние дорожного полотна потенциально опасным или вынужден отвлечься на что-то во время движения и по этой причине принимает решение временно снизить скорость. Второй параметр характеризует наличие «спокойных» водителей, не склонных лишний раз менять полосу движения, даже если это позволит ехать быстрее. Другой причиной отказа от перестроения может быть нерешительность водителя или отсутствие опыта.

Каждому автомобилю приписывается уникальный номер (ID), по которому можно отследить его положение в системе. Как и в случае одномерного движения, АТС характеризуется текущей и максимальной скоростью. Скорость принимает целые значения от 0 до 4, а при высоких скоростных режимах и более. Максимальная скорость равна $V_{max} = 4\frac{ячейки}{шаг\ по\ времени}$. Если шаг по времени равен одной секунде, то дискретному значению скорости 4 соответствует скорость 108 километров в час, то есть скорость свободного движения автомобиля по автомагистрали. Так как правилами дорожного движения в России скорость ограничена 110 км/ч, нам нет смысла рассматривать $V_{max} = 5\frac{ячеек}{шаг\ по\ времени} = 135\ км/ч$, как в модели Нагеля-Шрекенберга.

Ускорение в модели – это число единиц, на которое автомобиль может поменять скорость за один временной шаг. Важно сопоставлять это значение с реальным возможным ускорением автомобиля при движении. С этой точки зрения для более адекватного описания физического движения автомобиля при заданных параметрах ячеек и шага по времени не имеет смысла менять скорость автомобиля более чем на одну единицу за один шаг по времени.

Модель обеспечивает возможность достижения автомобилями цели. Целями автомобилей при многополосном движении могут быть, например, съезд с дороги или поворот в определенную сторону на светофоре. В обоих случаях машины начиная с определенного момента времени стремятся перестроиться в целевую полосу, игнорируя значения плотности и скорости на ней. Таким образом, в описываемой модели для каждой машины необходимо хранить параметр цели. Некоторые автомобили не достигают цели, если не смогли вовремя перестроиться в полосу, из которой разрешен поворот или съезд. Это происходит из-за того, что нужная полоса занята другими транспортными средствами. Модель предусматривает гибкий подход к достижению автомобилем его цели: число автомобилей, которые не достигли цели, может уменьшаться за счет разрешения поворотов из большего числа полос, усложнения алгоритма перестроения или за счет реализации алгоритма «вежливый водитель». Алгоритм заставляет водителей в целевой полосе пропускать машины, которые хотят перестроиться.

Каждый автомобиль в модели имеет набор параметров, таких как:

- уникальный номер (ID),
- текущая скорость,
- максимальная скорость,
- конечная цель.

Имеются также вспомогательные параметры, упрощающие выполнение логических операций. О них будет сказано далее.

В модели, однако, мы оперируем именно с ячейками, а не с автомобилями. Наличие автомобиля и набор его характеристик – это свойство данной ячейки на данном шаге по времени. Все параметры, кроме скорости, в пустых ячейках равны значениям по умолчанию, мы их не используем. Если скорость в ячейке с автомобилем принимает значение от 0 до 4 и соответствует скорости автомобиля, то параметр скорости пустой ячейки полагается равным -1, ячейки с неподвижным препятствием -10. Эти значения используются для построения алгоритмов движения.

Далее приведем основные алгоритмы, включенные в модель и позволяющие описывать движение в различных обстоятельствах. Алгоритмы будут приведены в виде блок-схем и/или псевдокода и дополнены словесным описанием.

Алгоритмы движения автомобилей

1. Общий алгоритм движения

Рассмотрим сначала самый простой и самый общий случай – движение по участку дороги без перекрестков, светофоров, сужений/расширений и т.д. Итак, на каждом шаге по времени обновление состояния ячеек в модели происходит в два этапа:

А) Для каждого АТС выясняется возможность и необходимость смены полосы. Производится смена полосы.

Б) Производится движение вперед по выбранной полосе по правилам однополосного движения.

В самом общем виде алгоритм смены полосы (А) можно записать так:

1.1 Алгоритм а1- общий алгоритм смены полосы

1. Проверка четности номера шага по времени. На четных шагах происходит смена полосы направо, на нечетных – налево: если *step*%2=0 *wmr*=true, *wml*=false; иначе *wmr*=false, *wml*=true.

Здесь *wml(r)*=true/false – переменная, обозначающая, состоится или не состоится смена полосы налево/направо соответственно. Изначально ее значение – true (перестроение состоится). Дальнейшая последовательная проверка условий переопределяет значение переменной (или может оставлять без изменений).

2. Выгодно ли перестроение: если $d_{m,k\pm1} \leq d_{m,k}$ или $v_{m,k\pm1} < v_{m,k}$, то *wml(r)*=false (здесь $d_{m,k}$ – расстояние до а/м в данной полосе, $d_{m,k\pm1}$ – расстояние до а/м в целевой полосе, $v_{m,k}$ и $v_{m,k\pm1}$ – скорости этих автомобилей, m – номер ячейки, k – номер полосы).

3. Есть ли препятствие на этой полосе: если существует m: $i < m \leq i+10$, $v_{m,k}$=-10 то *wml(r)*=true *(m* – номер ячейки впереди, $v_{m,k}=-10$ соответствует неподвижному препятствию).

4. Свободна ли целевая ячейка: если $v_{i,k\pm1} \neq -1$, то *wml(r)*=false ($v_{i,k\pm1} = -1$ соответствует пустой ячейке).

5. Условие безопасности: если $d^{prev}_{m,k\pm1} < V$, то *wml(r)*=false ($d^{prev}_{m,k\pm1}$ – расстояние до предыдущего автомобиля в целевой полосе, $v^{prev}_{i,k\pm1}$ – его скорость, V – величина, выбираемая исходя из стиля вождения данного водителя. $V=v_{max}$ или $V=v^{prev}_{i,k\pm1}$ для «осторожных» или «агрессивных» водителей соответственно, более подробно об этом будет рассказано ниже).

6. Стохастический элемент: если $p<rand()$ то *wml(r)*=false (p – заданная вероятность, *rand()* – случайная величина).

Как и в оригинальной модели Нагеля-Шрекенберга, считаем, что скорость и расстояние измеряются в ячейках, а время безразмерно и эти величины можно складывать, вычитать и сравнивать друг с другом.

Шаг 1 отвечает за проверку четности номера шага. Перестроение направо на четных шагах, налево на нечетных шагах выполняется с целью избежания конфликтов при числе полос больше двух, когда автомобили, расположенные «через ряд» друг от друга, могут претендовать на одну и ту же ячейку в ряду посередине. Согласно правилам дорожного движения (ПДД), автомобили должны руководствоваться правилом «помеха справа», то есть автомобиль из ряда левее должен пропустить автомобиль из ряда правее при таком конфликте. Однако это требует усложнения взаимодействия между автомобилями из разных рядов, то есть увеличения числа соседей для конкретной ячейки в терминах клеточных автоматов. Такое усложнение возможно в будущих версиях модели.

Шаг 2 определяет желание водителей двигаться в более быстром и/или более свободном ряду.

Шаг 3 проверяет, есть ли препятствие впереди на текущей полосе (если оно есть, водитель принимает решение в пользу перестроения, даже если целевой ряд более медленный). Также происходит проверка наличия препятствия в целевой полосе. При наличии препятствия запускается алгоритм объезда широкого препятствия, более подробно о котором будет рассказано далее.

Шаг 4 определяет, свободна ли целевая ячейка.

Шаг 5 проверяет условие безопасности. Для безопасного перестроения расстояние между целевой ячейкой и автомобилем, движущимся по целевой полосе и подъезжающим к этой ячейке, должно быть достаточным. Достаточность расстояния определяется скоростью автомобиля в целевой полосе и стилем вождения управляющего перестраивающимся транспортным средством. Соответственно, расстояние, которое водитель считает безопасным, может отличаться.

На шаге 6 перестроение может с некоторой вероятностью отмениться, даже если выполнены все условия для смены полосы.

Для наглядности общий алгоритм смены полосы представлен на рис. 2 в виде блок-схемы.

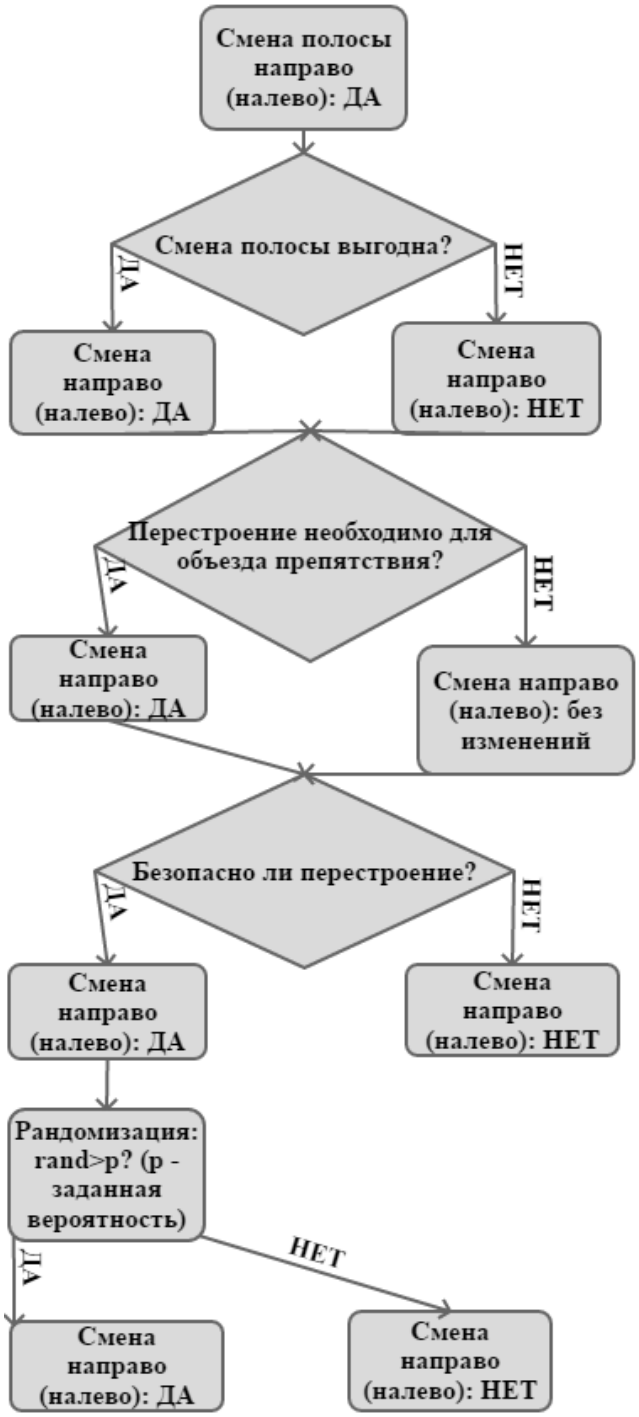


Рис. 2. Общий алгоритм смены полосы

Подшаг (Б) (алгоритм *a2*) выглядит так же, как в модели Нагеля-Шрекенберга:

1.2 Алгоритм a2 – общий алгоритм движения вперед

1. Ускорение. Скорость автомобиля увеличивается на единицу, если максимальная разрешенная скорость не достигнута:
   $v_{i,k} \to \min(v_{i,k} + 1, V_{max})$.

Это отображает желание водителя двигаться с максимальной разрешенной скоростью.

2. Торможение. Скорость автомобиля уменьшается на единицу, если есть угроза столкновения с впереди идущим автомобилем:
   $v_{i,k} \to \min(v_{i,k}, d_{m,k} - 1)$, где $d_{m,k}$ – расстояние до впереди идущего автомобиля. Данное условие – условие безопасности.
3. Случайные возмущения. Если скорость автомобиля положительна, то она может быть уменьшена на единицу с некоторой вероятностью:
   $v_{i,k} \to \max(v_{i,k} - 1, 0)$ с вероятностью p.
4. Движение. Выполнив пп. 1-3, мы определили скорость автомобиля на текущем шаге по времени и можем теперь переставить его по системе ячеек на расстояние, соответствующее данной скорости. Каждый автомобиль продвигается вперед на данное количество ячеек:
   $x_{i,k} \to x_{i,k} + v_{i,k}$.

2. Достижение цели

Рассмотрим теперь более сложные случаи. Если элемент улично-дорожной сети не просто прямой участок дороги, а, например, перекресток или разворот, у каждого автомобиля появляется несколько вариантов выезда с такого элемента.

Модель предусматривает достижение автомобилем цели. Цель определяется в рамках одного элемента улично-дорожной сети и соответствует номеру дороги, по которой машина должна выехать из расчетной области. Для обеспечения движения по сложным элементам УДС каждый такой элемент в модели в общем случае состоит из четырех дорог (направлений движения), моделирование на каждом из них происходит независимо от других. Взаимодействие между дорогами описывается отдельными алгоритмами, отличными для каждого конкретного элемента.

3. Движение на элементах улично-дорожной сети

Для созданной модели разработаны алгоритмы движения на элементах улично-дорожной сети с управляющей инфраструктурой. На рис. 3 представлены примеры моделирования движения на таких элементах.

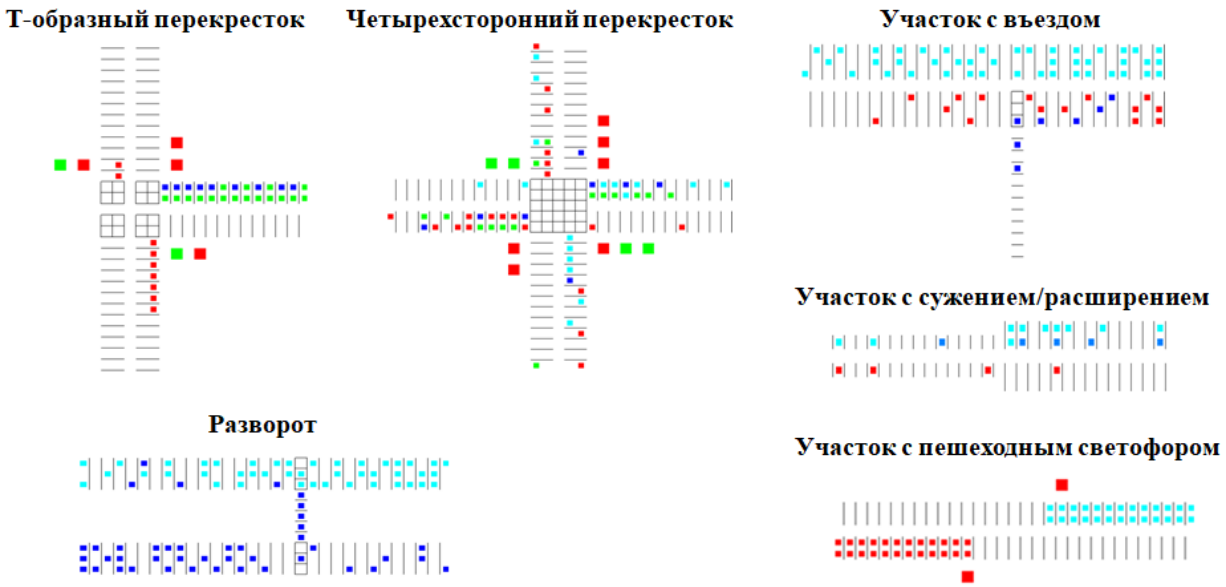


Рис. 3. Элементы УДС

Здесь и далее мелкие квадраты представляют автомобили, разные цвета отвечают разным целям. Более крупные квадраты – это светофоры. Соответственно, зеленым цветом обозначен разрешающий сигнал, красным – запрещающий сигнал в данном направлении. Представленные на рис. 3 изображения – это стоп-кадры визуализации расчетов, выполненных при помощи созданной модели CAM-2D.

Рассмотрим алгоритмы движения на участках УДС более подробно.

3.1 Алгоритм а3 перестроения с достижением цели (движение по участку дороги с выездом)

Рассмотрим теперь простейший случай, включающий достижение цели: участок дороги с выездом. Наличие выезда обозначает, что у автомобилей теперь есть две возможные цели: проехать участок прямо или воспользоваться съездом. В данном случае подшаг (А) общего алгоритма движения модифицируется.

1. Проверка четности номера шага по времени. На четных шагах происходит смена полосы направо, на нечетных – налево: если *step*%2=0 *wmr*=true, *wml*=false; иначе *wmr*=false, *wml*=true.

2. Выгодно ли перестроение: если $d_{m,k\pm1} \leq d_{m,k}$ или $v_{m,k\pm1} < v_{m,k}$, то *wml(r)*=false (здесь $d_{m,k}$ – расстояние до а/м в данной полосе, $d_{m,k\pm1}$ – расстояние до а/м в целевой полосе, $v_{m,k}$ и $v_{m,k\pm1}$ – скорости этих автомобилей, m – номер ячейки, k – номер полосы).

3. Есть ли препятствие на этой полосе: если существует m: $i < m \leq i+10$, $v_{m,k}$=-10, то *wml(r)*=true (m – номер ячейки впереди, $v_{m,k}$=-10 соответствует неподвижному препятствию).

4. Перестроение в нужный ряд вблизи цели: если $d_{i,k} < D$ (где $d_{i,k}$ – расстояние до цели, а D – фиксированное расстояние – параметр моделирования) и целевая полоса находится слева (справа), то *wml(r)*=true, если $d_{i,k} \geq D$ *wml(r)= wml(r)*.

5. Свободна ли целевая ячейка: если $v_{i,k\pm1} \neq -1$, то *wml(r)*=false ($v_{i,k\pm1} = -1$ соответствует пустой ячейке)

6. Условие безопасности: если $d^{prev}_{m,k\pm1} < V$, то *wml(r)*=false ($d^{prev}_{m,k\pm1}$ – расстояние до предыдущего автомобиля в целевой полосе, $v^{prev}_{i,k\pm1}$ – его скорость, V – величина, выбираемая исходя из стиля вождения данного водителя. $V=v_{max}$ или $V=v^{prev}_{i,k\pm1}$ для «осторожных» или «агрессивных» водителей соответственно, более подробно об этом будет рассказано ниже).

7. Стохастический элемент: если *p<rand()*, то *wml(r)*=false (p – заданная вероятность).

Отличие от общего алгоритма перестроения – в пункте 4. Вблизи от цели, при расстоянии до цели $d_{i,k} < D$, где D – фиксированное расстояние (параметр моделирования, зависящий от типа дороги и режима движения: чем больше скорость автомобилей, тем больше D), в алгоритм добавляется ветвь, переопределяющая значение переменных *wmr/wml* в зависимости от того, нужно ли перестроение для достижения цели.

На рис. 4 представлена блок-схема алгоритма перестроения автомобилей при наличии цели.

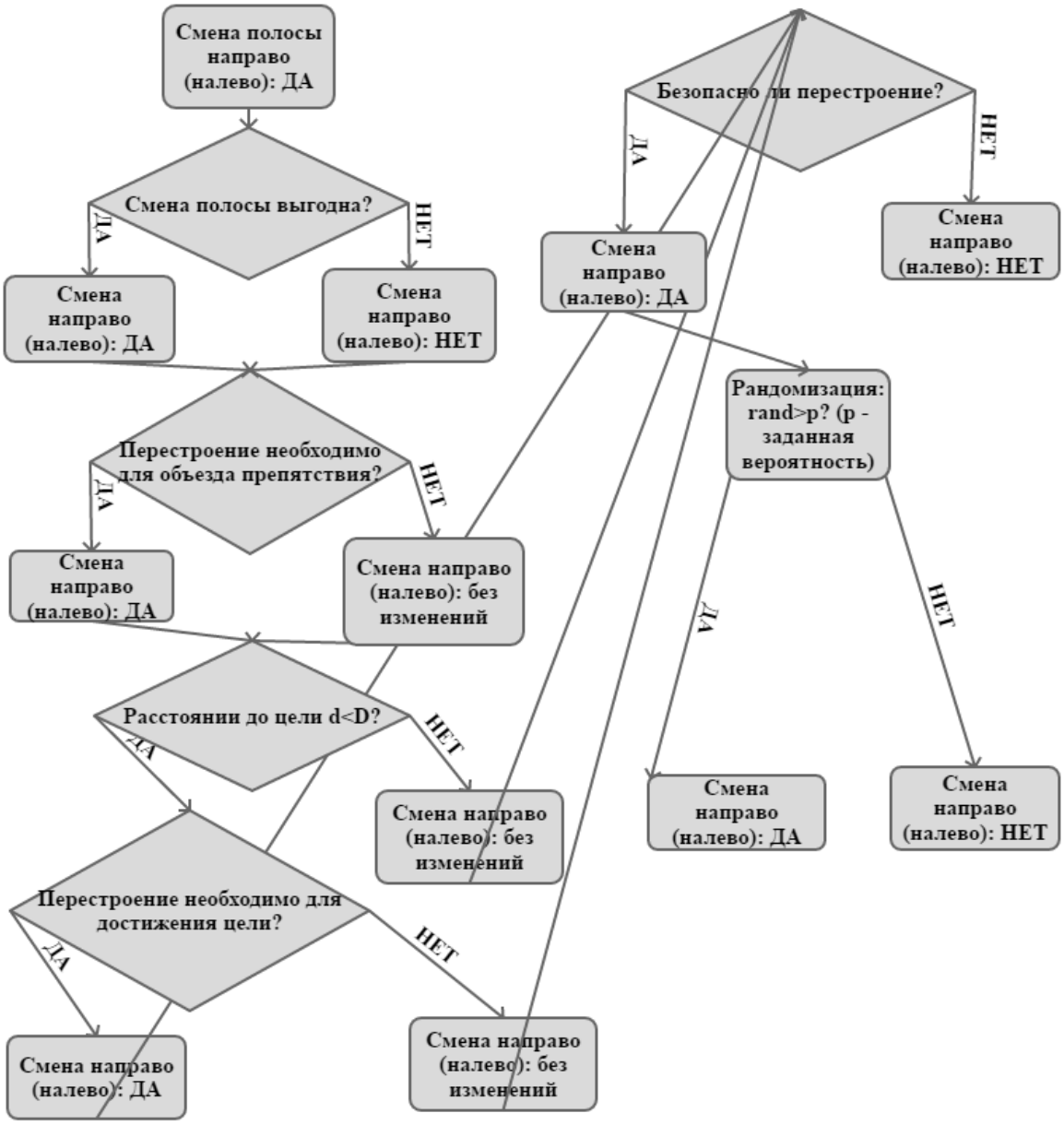


Рис. 4. Блок-схема алгоритма перестроения автомобилей при наличии цели

Подшаг (Б) – движение вдоль дороги – реализуется так же, как в общем алгоритме перестроения (алгоритм *a2*).

3.2 Алгоритм a4 прохождения четырехстороннего перекрестка со светофором

При прохождении регулируемого четырехстороннего перекрестка смена полосы происходит по алгоритму *a3* (перестроение с достижением цели). Его шаги 1-7 полностью повторяются.

Движение вперед происходит согласно алгоритму *a2,* однако при подъезде к перекрестку в него добавляются новые условия. Следующие правила применяются для того, чтобы автомобиль мог замедляться перед перекрестком и останавливаться на светофоре, а также поворачивать:

1. Замедление за D_2 ячеек перед поворотом: если $d_{i,k,j} < D_2$, то $v_{i,k,j}$= min($v_{i,k,j}$ -1, 1) ($d_{i,k,j}$ – расстояние до перекрестка, D_2 – фиксированное расстояние, $v_{i,k,j}$ – скорость автомобиля, i – номер ячейки, k – номер полосы, j – номер дороги).

2. Остановка в точке поворота: если $d_{i,k,j}$=0, то $v_{i,k,j}$= 0.

3. Замедление и остановка на красный: если reg=R, то при $d_{i,k,j}$>0 $v_{i,k,j}$=min($v_{i,k,j}$ -1, 0), при $d_{i,k,j}$=0, то $v_{i,k,j}$= 0 (reg – режим светофора, R соответствует красному сигналу для данного направления).

4. Поворот/смена дороги j: если $v_{i,k,j}$= 0 и $reg \neq R$, и $k = k^{goal}$, и $v_{i*,k*,j*}$ =-1 (целевая ячейка свободна), то $v_{i,k,j} \rightarrow v_{i*,k*,j*}$ (здесь k^{goal} – целевая полоса, из которой разрешен поворот).

5. Движение прямо, если текущая полоса не предполагает поворот и целевая ячейка свободна: если $v_{i,k,j}$= 0 и $reg \neq R$, и $k \neq k^{goal}$, и $v_{i+1,k,j}$ =-1, то $v_{i,k,j} \rightarrow v_{i+1,k,j}$.

Если цель машины – поворот на перекрестке, и она находится в пределах D_2 клеток от поворота, то ее скорость уменьшается, пока не достигнет 1. При достижении точки поворота машина останавливается.

Если горит красный сигнал светофора, автомобиль, подъезжающий к перекрестку, замедляется, в ячейке перед перекрестком он останавливается. Когда загорается зеленый, автомобиль снова начинает движение.

Если цель автомобиля – поворот на перекрестке (номер дороги, соответствующий цели, предполагает поворот), и он находится в полосе, из которой данный поворот разрешен, он поворачивает. Это означает, что он попадает в соответствующую ячейку другой дороги, проверив, что она не занята.

3.3 Алгоритм a5 – обеспечение безопасности на перекрестке

Рассмотрим реализацию поворотов более подробно. Как уже упоминалось ранее, расчетная область всех элементов УДС состоит из четырех независимых участков. Для четырехстороннего перекрестка эти участки – направления движения (обозначены цифрами 0-3 на рисунке 7). Обозначения направлений следующие: движение снизу вверх – j=0, сверху вниз – j=1, справа налево – j=2, слева направо – j=3.

Каждое направление, или дорога, может иметь любое количество полос.

Подчеркнем, что термин «дорога» здесь соответствует именно одному направлению движения, в отличие от обычного употребления этого слова, когда дорогой называют оба направления (если движение не одностороннее). В обычном понимании дорога – это совокупность направлений 0 и 1 или 2 и 3 на рис. 5. Мы, однако, при упоминании номера дороги будем всегда иметь в виду номер направления.

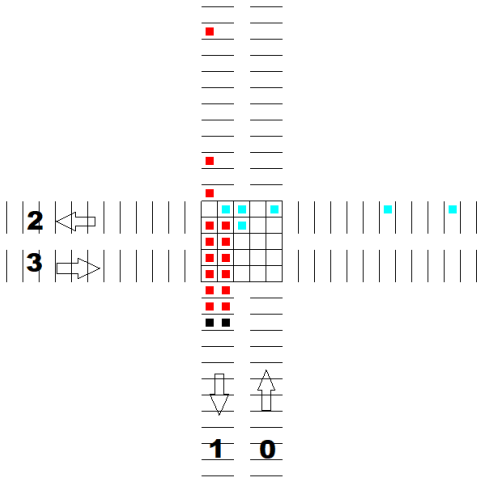


Рис. 5. Схема движения на четырехстороннем перекрестке

Автомобили, как упоминалось ранее, обозначены цветными квадратами, где цвет определяет направление. Черные квадраты обозначают неподвижное препятствие, необходимое в этом расчете для верификации движения на перекрестке, а именно: для того чтобы автомобили направления 1 остались в пределах области пересечения дорог. Это позволяет увидеть, как происходит движение автомобилей по направлению 2 по перекрестку с другими автомобилями.

Если моделируется, скажем, прямой участок УДС, то мы «выключаем» лишние направления (просто взяв входящие потоки по данным направлениям равными нулю) и взаимодействие этих направлений с актуальными.

При описании пересечений дорог различных типов необходимо, разумеется, описать взаимодействие пересекающихся направлений. В данном случае при расчете движения на четырехстороннем перекрестке у направлений 0-3, 0-2, 1-3, 1-2 имеются общие области. Они вместе с «разделительными полосами» в одну клетку, имеющимися между направлениями 0-1 и 2-3, образуют область перекрестка.

Для учета нахождения автомобилей на перекрестке создается матрица $cross[x][y]=v_{j1} + v_{j2}$, где v_{j1} и v_{j2} – значения скоростей в соответствующих ячейках по направлениям $j1$ и $j2$, x и y – переменные, нумерующие элементы матрицы перекрестка. Если ячейка расчетной области пустая, значение скорости в ней равно -1. Таким образом, элемент матрицы равен -2 тогда и только тогда, когда ячейка перекрестка пустая.

При подъезде автомобиля к границе перекрестка происходит обращение к соответствующему элементу матрицы *cross[x][y]* для проверки наличия автомобилей. Если целевая ячейка занята, автомобиль останавливается и ждет, когда она освободится. На следующем шаге он может попробовать перестроиться в другой ряд и повторить попытку проехать.

3.4 Алгоритм а6 прохождения Т-образного перекрестка со светофором

Этот алгоритм реализуется как частный случай алгоритма *а4*. Для этого используется входящая в общий алгоритм перестроения возможность объезда неподвижного препятствия. Ячейки расчетной области, занятые препятствием, обозначены при помощи параметра скорости, равного -10. «Лишние» ячейки расчетной области заполняются неподвижными препятствиями.

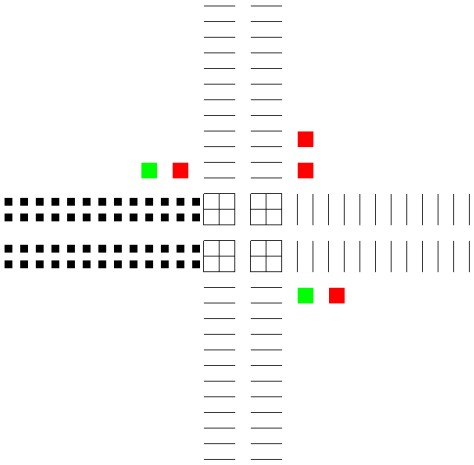


Рис. 6. Т-образный перекресток как вариант четырехстороннего

На рис. 6 показано, как с помощью препятствий, обозначенных черными квадратами, происходит перекрытие незадействованных областей перекрестка. Теперь автомобили не могут попасть в запрещенные ячейки.

Для Т-образного перекрестка также адаптирован алгоритм достижения цели. Так как по направлению j=2 (справа налево) нельзя проехать в прямом направлении, возможна ситуация, при которой автомобили, которым не удалось перестроиться в целевой ряд для поворота, а также те, для кого задана несуществующая цель j=2, могут «застрять» на перекрестке. Для того чтобы этого не происходило, все автомобили должны повернуть направо или налево, даже если не оказались в целевом ряду. Чтобы не нарушать ПДД, они поворачивают туда, куда разрешен поворот из ряда, в котором они оказались.

3.5 Алгоритм а7 прохождения участка с сужением/расширением

Рассмотрим участок дороги с сужением и расширением, как показано на рис. 7.

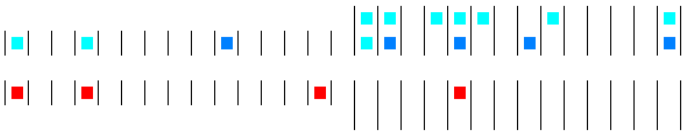


Рис. 7. Участок УДС с сужением/расширением

При движении слева направо на дороге имеется расширение, а справа налево – сужение. На практике мы в данном случае имеем дело с участком, имеющем по две полосы в каждом направлении, но расчетная область в левой части рисунка, там, где дороги однополосные, ограничена при помощи

заполнения ячеек неподвижными препятствиями, аналогично Т-образному перекрестку.

Проезд участка с сужением обеспечивается общим алгоритмом перестроения *a1*, который предполагает обнаружение препятствия в текущей полосе при приближении к нему (за 10 ячеек) и перестроение при первой возможности (*a1* п.3).

Проезд участка с расширением также не требует дополнительной адаптации алгоритмов. Перестроение в появившуюся полосу происходит за счет желания водителя двигаться по более свободному ряду (*a1* п.2).

3.6 Алгоритм a8 въезда с второстепенной дороги на главную

С точки зрения расчетной области, въезд на главную дорогу – вариант Т-образного перекрестка, лишние ячейки, на которые не должны заезжать автомобили, обозначаются при помощи параметра скорости v=-10.

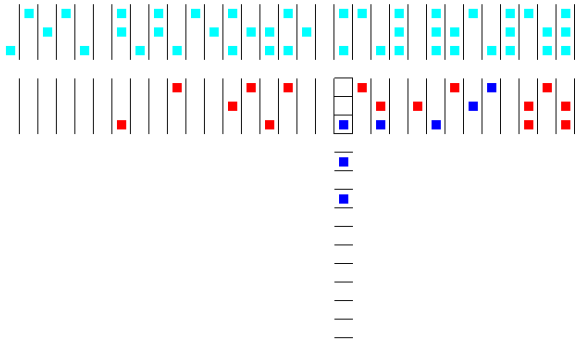


Рис. 8. Участок УДС с въездом

На рис. 8 представлен участок УДС, состоящий из трехполосной дороги с двусторонним движением и однополосного въезда. Широкая дорога является главной.

Алгоритм въезда, обеспечивающий приоритет автомобилям, движущимся по главной дороге, выглядит следующим образом.

1. Замедление за D_2 ячеек перед поворотом: если $d_{i,k,j} < D_2$, то $v_{i,k,j}$= min($v_{i,k,j}$-1, 1), $d_{i,k,j}$ – расстояние до перекрестка, D_2 – фиксированное расстояние, $v_{i,k,j}$ – скорость автомобиля, i – номер ячейки, k – номер полосы, j – номер дороги.

2. Задается переменная, обозначающая, состоится ли въезд *we* (want to enter), по умолчанию *we*=false (въезд невозможен).

3. Если расстояние до машины, подъезжающей к точке въезда по главной дороге $d^{prev}_{i,k,j*}>V^*$,то *we*=true. V^* может принимать разные значения в зависимости от стиля вождения, $v^{prev}_{i,k,j*}<V^*\leq V_{max}$, где $v^{prev}_{i,k,j*}$ – скорость автомобиля на главной дороге, V_{max} – максимальная скорость.

4. Поворот/смена дороги j: если *we*=true и целевая ячейка свободна ($v_{i*,k*,j*}$=-1), то $v_{i,k,j} \rightarrow v_{i*,k*,j*}$.

3.7 Алгоритм a9 проезда участка с разворотом

Участок с разворотом (рис. 9) – это комбинация участка с выездом и участка с въездом.

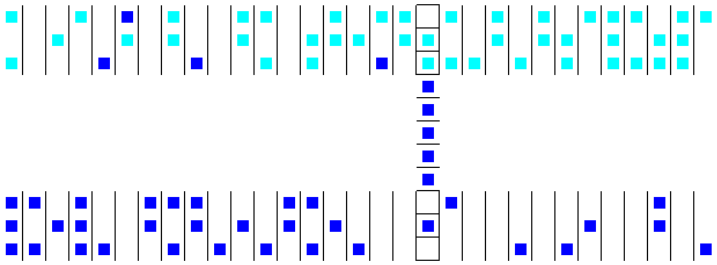


Рис. 9. Участок с разворотом

Расчетная область, как всегда, реализована при помощи запрета движения на неиспользуемых участках четырехстороннего перекрестка. На рисунке 11 движение по нижней дороге происходит слева направо (j=3), автомобили, въехавшие по данному направлению, обозначены темными квадратами. Часть автомобилей проезжает участок прямо, а часть едет на разворот. Движение по верхней дороге происходит, соответственно, справа налево (j=2) и автомобили, заехавшие на этот участок справа, более светлые. На верхней дороге в левой части видно несколько темных автомобилей, которые попали на это направление с полосы для разворота.

Данный пример расчета выполнен для широкой разделительной полосы между направлениями 2 и 3 (для этого потребовалась небольшая дополнительная модификация расчетной области, помимо заполнения недоступных ячеек). Длина полосы для разворота может быть любой, в том числе и равной нулю.

С точки зрения алгоритмов, здесь используется комбинация *a3* и *a8* (выезда и въезда с второстепенной дороги на главную).

3.8 Алгоритм a10 проезда участка с пешеходным светофором

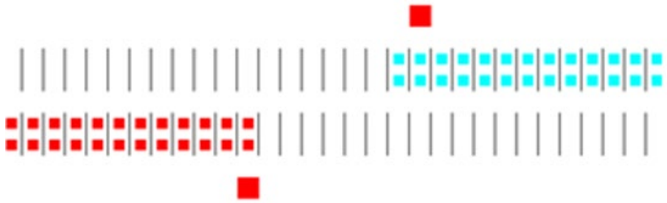


Рис. 10. Участок с пешеходным светофором

Элемент с пешеходным светофором представлен на рис. 10. Алгоритм проезда светофора – упрощенный вариант алгоритма *a4* без поворотов. Реализуются только части, отвечающие за светофорное регулирование.

3.9 Алгоритм a11 объезда широкого неподвижного препятствия

Следующий алгоритм – усложненный вариант алгоритма перестроения *a1* для случая, когда на многополосной дороге имеется препятствие, занимающее

более одной полосы (рис. 11). Препятствие обозначено прямоугольниками с крестиками. Светлые и темный квадраты обозначают автомобили. Темный автомобиль должен принять решение, с какой стороны объезжать препятствие – в данном случае более выгодным является перестроение направо (в нижние полосы на данном рисунке).

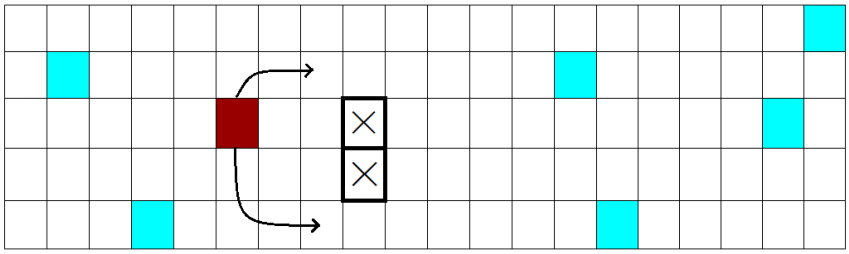


Рис. 11. Участок с широким препятствием

Этот случай реализуется, например, при ремонте дорожного покрытия или при крупном ДТП. Водитель, подъезжающий к такому препятствию, должен принять решение, с какой стороны его объехать, выбрав более короткий путь. Для этого он проверяет, сколько ячеек поперек дороги занимает препятствие слева и справа от него. Если для объезда препятствия перестраиваться в данную сторону невыгодно и проще объехать его с другой стороны, смена полосы отменяется.

Формальная запись алгоритма перестроения для случая с широким препятствием выглядит следующим образом:

1. Проверка четности номера шага по времени: если *step*%2=0 *wmr*=true, *wml*=false; иначе *wmr*=false, *wml*=true.

2. Выгодно ли перестроение: если $d_{m,k\pm1} \leq d_{m,k}$ или $v_{m,k\pm1} < v_{m,k}$, то *wml(r)*=false.

3. Есть ли препятствие на этой полосе:

   - если есть m: $i < m \leq i+10$, $v_{m,k}=-10$, то есть ли препятствие на целевой полосе? ($v_{m,k\pm1}=-10$?),
     - если нет, *wml(r)*=true,
     - если да, находим значения *NLl* и *NLr* – число полос, занятых препятствием справа и слева от текущего положения автомобиля. При *NLl*<*NLr* *wml*=true, *wmr*=*wmr*; иначе *wml*=*wml*, *wmr*=true,
   - если препятствия на этой полосе нет, *wml(r)*= *wml(r)*.

4. Свободна ли целевая ячейка: если $v_{i,k\pm1} \neq -1$, то *wml(r)*=false.

5. Условие безопасности: если $d^{prev}_{m,k\pm1} < V$, то *wml(r)*=false.

6. Стохастический элемент: если *p*<*rand()*, то *wml(r)*=false.

Шаги 1-2 и 4-6 такие же, как в *a1*, обозначения те же. Шаг 3 отличается.

Для верификации алгоритма была решена тестовая задача. На рис. 12 показан некоторый момент времени выполнения численного эксперимента. Черные квадраты представляют широкое неподвижное препятствие, цветные квадраты – автомобили. Движение происходит слева направо.

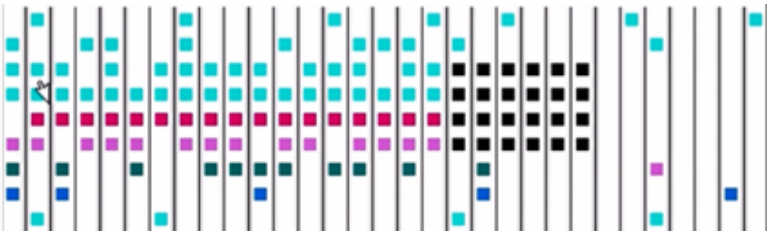


Рис. 12. Участок с широким препятствием (результат моделирования)

Цель эксперимента – проверить, что автомобили выбирают нужное направление объезда. Для наглядности автомобили, которые заезжают в расчетную область по левым рядам и должны объезжать препятствие слева, имеют голубой цвет, а те, что заезжают по правым рядам и должны объезжать справа – разноцветные (цвет здесь отвечает полосе въезда и служит для проверки перестроения при расчете). Выполнен ряд расчетов с разным расположением и шириной препятствия. Расчеты показывают, что водители верно выбирают направление объезда; это видно на рис. 12. Можно сделать вывод, что алгоритм работает верно.

4. Различные стратегии поведения водителей

В модели реализованы две стратегии перестроения водителей: «агрессивные» и «осторожные». Эти качества проявляются в манере перестроения и при въезде с второстепенной дороги на главную.

- «Осторожные» водители для смены полосы или въезда выбирают большой интервал до следующего автотранспортного средства в целевой полосе/дороге. Расстояние до следующего автомобиля должно быть больше максимально разрешенной скорости в системе.

- «Агрессивные» водители оценивают текущую скорость следующего автомобиля, движущегося по целевой полосе/дороге, и перестраиваются вплотную, сохраняя лишь минимальный зазор, необходимый для обеспечения безопасности.

Реализация этих стратегий входит во все описанные выше алгоритмы перестроения и въезда:

✓ Условие безопасности: если $d^{prev}_{m,k\pm1} < V$, то $wml(r)$=false ($d^{prev}_{m,k\pm1}$ – расстояние до предыдущего автомобиля в целевой полосе, $v^{prev}_{i,k\pm1}$ – его скорость, V – величина, выбираемая исходя из стиля вождения данного водителя).

Если водитель «осторожный», то $V=v_{max}$ или больше. Перестроение произойдет, только если число ячеек, отделяющее его от автомобиля в целевой полосе, больше разрешенной максимальной скорости в системе.

Если водитель «агрессивный», то $V=v^{prev}_{i,k\pm1}$. Водитель сравнивает расстояние до автомобиля в целевой полосе с его скоростью, может перестроиться «вплотную» к подъезжающему по целевой полосе автомобилю.

Количество тех и других водителей в системе, а также условия на скорость можно варьировать.

Разумеется, при реализации модели «агрессивные» водители быстрее объезжают препятствия, и в результате вероятность образования затора меньше. Для верификации алгоритма были выполнены тестовые расчеты по моделированию объезда ДТП. На рис. 13(а), 13(б) представлены результаты этих расчетов.

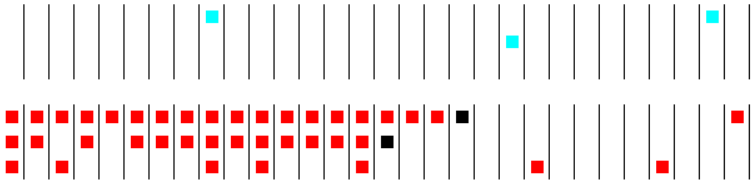


Рис. 13(а). Объезд ДТП, «осторожные» водители

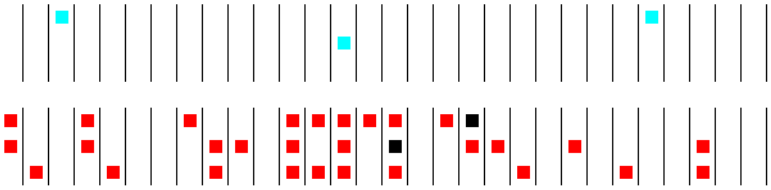


Рис. 13(б). Объезд ДТП, «агрессивные» водители

На обоих рисунках изображен участок трехполосной дороги, в верхней части направление движения справа налево, в нижней части – слева направо. Квадратами обозначены автомобили. Черным цветом обозначены стоящие автомобили – участники ДТП. В расчете они играют роль неподвижного препятствия. Цветные – движущиеся автомобили.

Расчет, представленный на рисунке 13(а), выполнен для ситуации, когда все водители являются «осторожными», то есть для перестроения ждут образования зазора на целевой полосе, равного как минимум максимально разрешенной скорости в системе. На рисунке 13(б) водители перестраиваются более смело, сравнивая имеющийся промежуток на целевой полосе с текущей скоростью следующего автомобиля. Как видно из рисунков, при «агрессивной» стратегии перестроения водителям удается преодолеть возникшее препятствие более эффективно: плотность автомобилей перед препятствием во втором случае меньше, итоговая фактическая пропускная способность участка больше.

Часть водителей в системе могут при определенных условиях стать «вежливыми», то есть пропустить автомобиль из соседнего ряда, который хочет перестроиться, но не имеет такой возможности по соображениям безопасности. «Вежливым» может быть как «агрессивный», так и «осторожный» водитель.

Целью данного алгоритма является описание кооперации водителей для прохождения сужений или неожиданных препятствий на дорогах. Как известно, по правилам дорожного движения автомобили из полосы, где находится препятствие, могут перестроиться в соседнюю полосу для объезда только в том случае, если они не мешают водителям из свободной полосы. Однако на практике в ситуации плотного потока или затора это бы привело к тому, что такие автомобили были бы вынуждены ждать освобождения целевой полосы очень долго. На этот случай существует негласное правило проезда «елочкой»: по очереди, один автомобиль из полосы с препятствием – один автомобиль из полосы без препятствия. Конечно, не все водители поголовно применяют это правило. Недостаток водителей, готовых пропускать других добровольно, однако, восполняется наличием более агрессивных, которые «вклиниваются» в полосу, не оставляя выбора, кроме как остановиться и пропустить их. Оба варианта описываются созданным алгоритмом.

4.1 Алгоритм a12 «вежливый водитель» для смены полосы

Алгоритм «вежливый водитель» («cooperative driver») состоит из двух частей. Первая часть – модифицированный алгоритм смены полосы при наличии в системе вежливых водителей (*a12.1*, рис. 14). Вторая часть (*a12.2*) – присвоение водителю статуса «вежливый».

Каждая часть, относящаяся к алгоритму a12, обозначена флагом *cdonoff* (cooperative driver on/off). Если *cdonoff* принимает значение true или 1, модули «cooperative driver» включаются в расчет и меняют общую схему принятия решения при смене полосы.

Вежливые водители в системе отмечены при помощи дополнительного параметра, описывающего состояние каждой ячейки, – *cd*, который принимает значения да/нет (1 или 0). Напомним, что каждая ячейка имеет набор параметров, позволяющих сохранить всю необходимую информацию об автомобиле: его скорость, максимальная скорость, идентификационный номер ID, точка назначения (номер выезда), время въезда в расчетную область, а также ряд других вспомогательных параметров. Среди этих параметров есть индикаторы желания перестроиться направо/налево (*lchr(l),* lane change right/left), которые не учитывают условия безопасности, то есть, они определяют не окончательное решение о перестроении, а лишь намерение поменять полосу. Эти параметры в модели выполняют ту же роль, что и «поворотники» в настоящем автомобиле. Другие водители могут их видеть и при определенных обстоятельствах принимать во внимание и замедляться, или пропускать автомобиль, который хочет перестроиться, а могут и игнорировать.

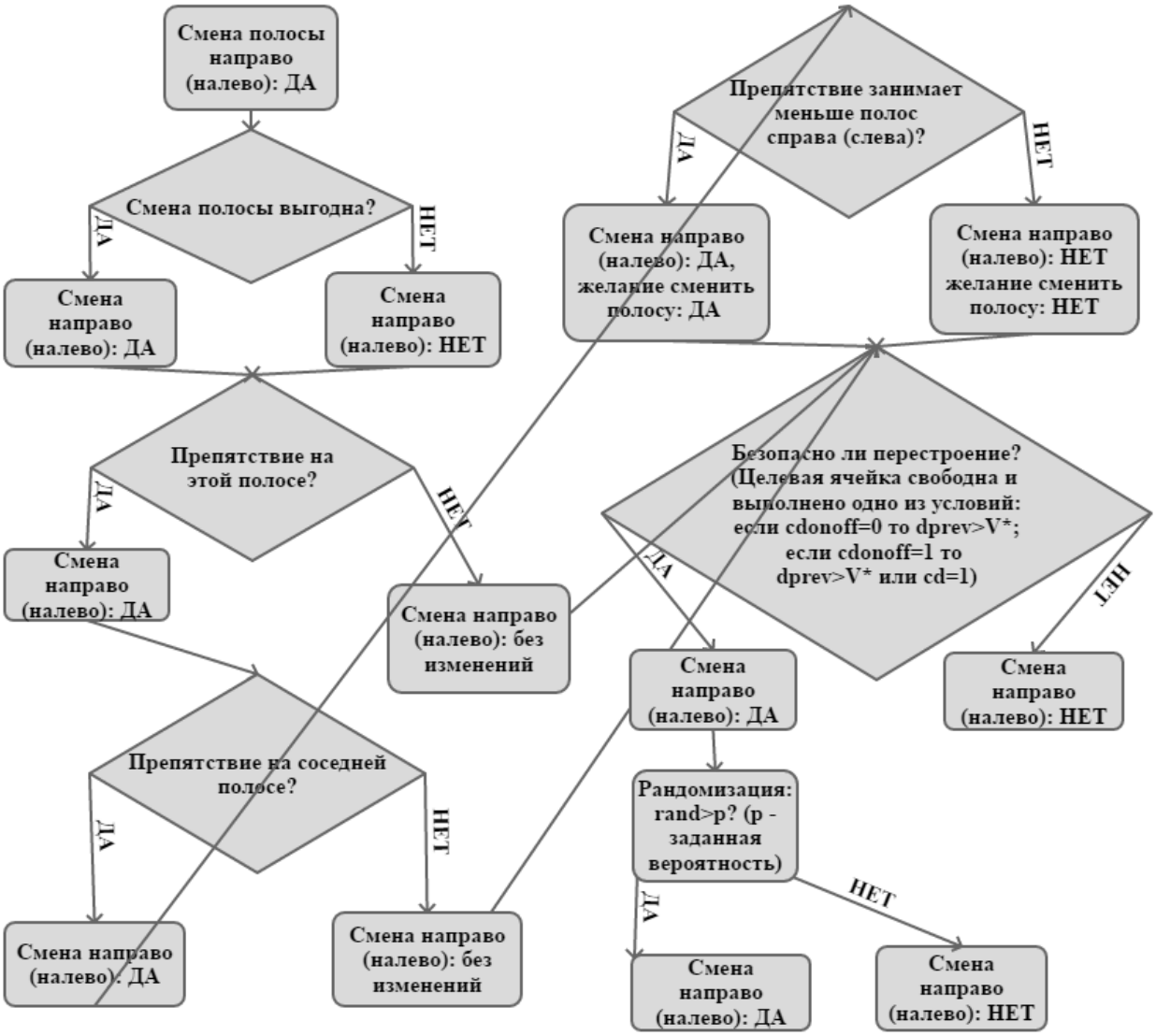


Рис. 14. Алгоритм *a12.1* перестроения при наличии «вежливых» водителей

Первая часть алгоритма, представленного на рисунке 14, уже была описана ранее (*a11*, пп.1-3). Отличие начинается после проверки, занимает ли препятствие меньше полос с той стороны, куда предполагается перестроение. В данном месте параметр *wmr/wml* сохраняется в переменную, обозначающую желание перестроиться. Если в результате указанных действий параметр имеет значение true, автомобиль «включает поворотник».

Далее происходит проверка безопасности перестроения. Она выглядит по-разному для расчета с «вежливыми» водителями (*cdonoff*=1) и без них (*cdonoff*=0). Если в системе нет «вежливых» водителей, то проверяется условие $d^{prev}>V^*$, где d^{prev} – расстояние до предыдущего автомобиля в целевой полосе, а V^* – скорость для сравнения, зависящая от особенностей стратегий водителей («агрессивных» и «осторожных»). При наличии в системе «вежливых» водителей (*cdonoff*=1) условие безопасности $d^{prev}>V^*$ уже может быть избыточно жестким, так как вежливые водители специально тормозят и останавливаются, чтобы пропустить других, и водитель с включенным

«поворотником» может безопасно перестроиться вплотную перед стоящим автомобилем с «вежливым» водителем. Таким образом, совместно с условием безопасности проверяется, является ли водитель предыдущего автомобиля в целевой полосе «вежливым» (*cd*=1) на данный момент. То, как водители в системе становятся «вежливыми», будет изложено далее.

Соответственно, если условие безопасности не выполнено, смена полосы отменяется.

Следующим шагом в алгоритме является рандомизация. Водитель, желающий поменять полосу и имеющий такую возможность, тем не менее может передумать перестраиваться с некоторой заданной вероятностью *p*. Затем, если водитель не передумал, происходит перестроение.

Далее рассмотрим, в каких условиях водителю присваивается статус «вежливый» (алгоритм *a12.2*). Это переменная величина, которая переопределяется каждый шаг по времени, однако можно задать определенное правило, по которому только часть водителей может стать вежливыми в процессе расчета. В нашем случае в качестве определяющего параметра служит идентификационный номер (ID), присваиваемый при добавлении автомобиля в систему. Половина водителей (с четными ID) могут стать «вежливыми», а другая половина – нет.

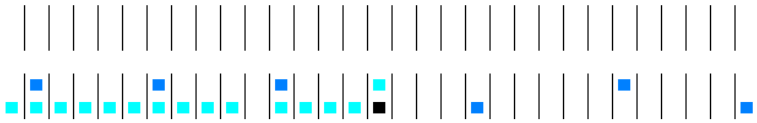
Присвоение параметра *cd*=1 происходит непосредственно перед сменой полосы, после выполнения всех шагов рис. 1. Таким образом, использоваться этот параметр в алгоритме будет только на следующем шаге по времени. Для случая перестроения условия для того, чтобы водитель стал «вежливым», следующие:

- *step*%2=0 для *wmr* (*step*%2=1 для *wml*).
- «Поворотник» автомобиля в полосе с препятствием включен: *lchr(l)*=true.
- Скорость автомобиля в полосе с препятствием $v^{prev}_{i,k\pm l}=0$. В полосе с препятствием уже образовался затор.
- Скорость данного автомобиля $-1< v_{i,k}\leq 1$. Он либо уже остановился, либо его скорость достаточно мала для полной остановки.
- Номер данного автомобиля в системе *ID*%2=0 (или другое условие, определяющее соотношение «вежливых» водителей и всех остальных).
- $cd\neq-1$ (*cd*=-1 после того, как водитель пропустил автомобиль из соседнего ряда и тот перестроился).

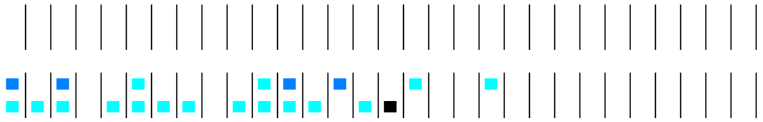
Последнее условие служит для того, чтобы один и тот же водитель, став вежливым, не пропускал сразу много машин из соседнего ряда, а сам начинал движение, пропустив только одну. Таким образом, моделируется так называемое перестроение «елочкой» (zipper merge в англоязычной литературе), когда водители из свободного ряда пропускают по одному автомобилю из ряда с затрудненным движением. Практика показывает, что такой подход проезда участка с неподвижным препятствием наиболее эффективен.

Для верификации алгоритма рассмотрим задачу об объезде неподвижного препятствия. Для расчетов был выбран случай двухполосной дороги, одну

полосу которой заблокировал неподвижный автомобиль (рис. 15).



15(а)



15(б)

Рис. 16. Объезд препятствия – результат моделирования. Сверху (*15(а)*) – без «вежливых водителей», снизу (*15(б)*) – с «вежливыми водителями»

Здесь автомобили представлены квадратами, черный квадрат – неподвижное препятствие, голубые – автомобили, заехавшие в расчетную область по полосе с препятствием, синие – те, кто въехал по свободной полосе. Движение происходит слева направо. При расчетах видно, что в случае (а) – без «вежливых» водителей – движение в правой полосе, где находится препятствие, сильно затруднено, а в случае (б) – с «вежливыми» водителями – затруднено меньше. Однако это не настолько очевидно, особенно из рисунков, поэтому была выполнена серия расчетов среднего времени нахождения автомобиля в системе при разных условиях. Для наглядности также отдельно анализировалось среднее время нахождения в системе автомобилей из полосы с препятствием. Расчеты показывают, что, в зависимости от выбранных параметров моделирования и значений потока, наличие «вежливых» водителей может снизить среднее время нахождения в системе автомобилей из полосы с препятствием в два раза и более.

4.2 Алгоритм a13 «вежливый водитель» для въезда с второстепенной дороги на главную

На рис. 17 представлена блок-схема алгоритма принятия решения о въезде с второстепенной дороги на главную (*a13.1*). Задается булева переменная *we* (want to enter), обозначающая, будет ли решение о въезде положительным, изначально она имеет значение false. Далее следует проверка безопасности, которая, как и в блок-схеме *a12.1*, различается для случая с «вежливыми водителями» и без них.

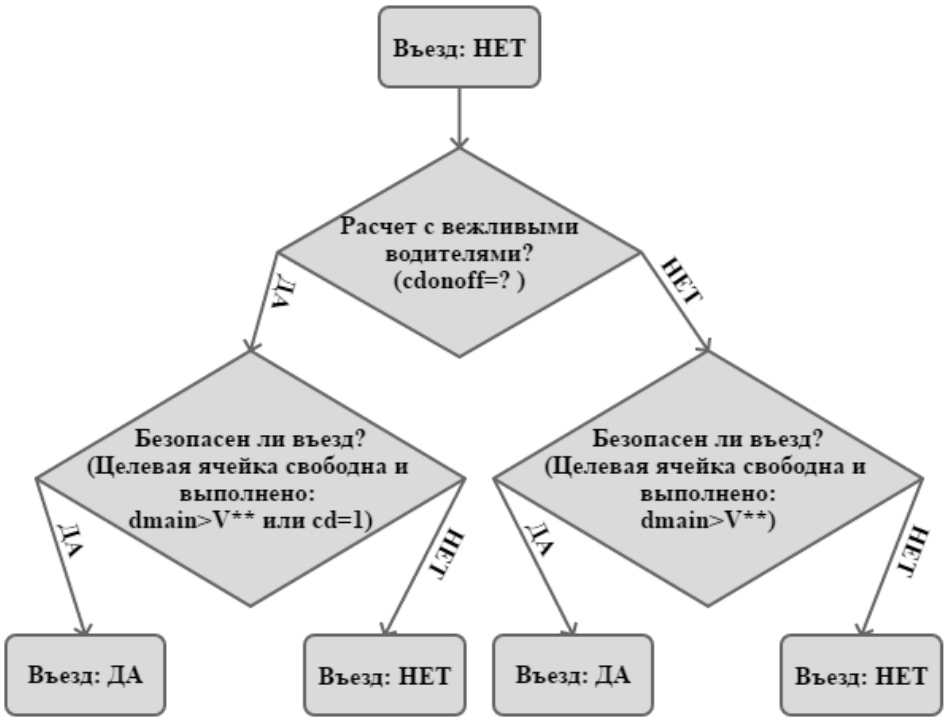


Рис. 16. Алгоритм *a13.1* въезда на главную дорогу с второстепенной при наличии «вежливых» водителей

В случае, когда *cdonoff*=1, должно выполниться одно из двух условий: *dmain*>*V*** или *cd*=1. Здесь *dmain* – расстояние от ближайшего автомобиля, едущего по главной дороге, до точки въезда, а *V*** – некоторая скорость, выбираемая в зависимости от желаемых особенностей водительских стратегий. Как и в случае с *V**, ее можно задать по-разному. *cd*=1, аналогично, обозначает, что водитель, движущийся по главной дороге, «вежливый» и готов остановиться, чтобы уступить дорогу въезжающему.

Соответственно, если *cdonoff*=0, то действует только условие *dmain*>*V***.

Рассмотрим условия задания *cd*=1 для этого случая. Водители, движущиеся по главной дороге, могут пропускать тех, кто едет по второстепенной дороге, если там образовался затор. Водитель автомобиля А на главной дороге, подъехавший к точке пересечения с второстепенной, становится «вежливым» (*cd*=1) и пропускает автомобиль В, если выполнены все перечисленные условия (*a13.2*):

Для автомобиля В –

- $i^*=i^*_{entr}$. Автомобиль В находится у въезда.
- $<v^*_{i,k}>=0$. На второстепенной дороге, кроме В, есть еще несколько стоящих в очереди на въезд автомобилей.

Для автомобиля А –

- Скорость А $-1< v_{i,k}\leq 1$. Он либо уже остановился, либо его скорость достаточно мала для полной остановки.

- *ID*%2=0 (или другое условие, определяющее соотношение «вежливых» водителей и всех остальных).
- *cd*≠-1. На этом шаге по времени водитель А еще не пропускал другие автомобили (*cd*=-1 после того, как водитель пропустил автомобиль из соседнего ряда и тот перестроился).

Далее в расчетном модуле следует набор шагов из модели Нагеля-Шрекенберга (алгоритм *a2*) с дополнительными условиями для обеспечения работы алгоритмов «вежливый водитель». Так, на первом шаге, отвечающем за ускорение, при включенном режиме *cdonoff*=1 проверяется условие: если водитель вежливый (*cd*=1) и его скорость равна нулю, то данный автомобиль не ускоряется, его скорость остается равной нулю. На втором шаге, описывающем замедление, автомобиль с *cd*=1 замедляется до остановки.

Как только «вежливый» водитель пропустил другого, ему присваивается *cd*=-1, что означает, что на данном шаге по времени он уже не может снова стать «вежливым».

Рассмотрим задачу о въезде на главную дорогу (рис. 17) для верификации алгоритма *a13*. По главной дороге поток автомобилей равен *Q1*, по второстепенной дороге – *Q2*.

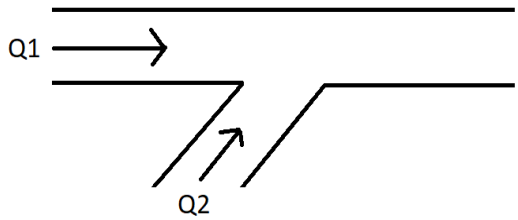
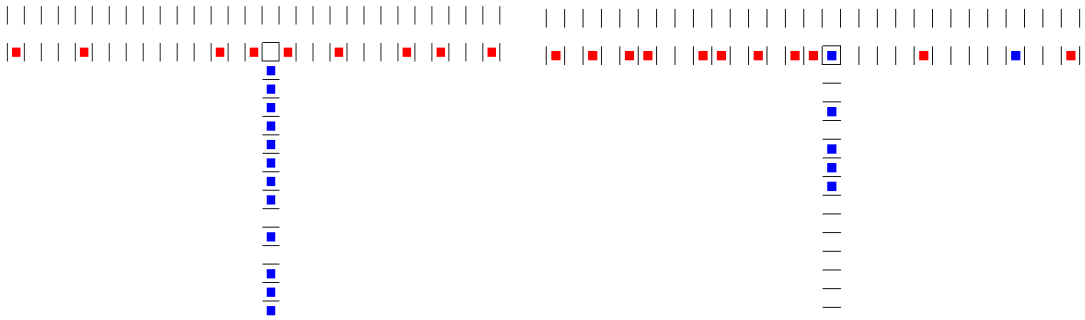


Рис. 17. Въезд на главную дорогу с второстепенной

В зависимости от выбора стратегии поведения водителей в модели, а именно, насколько большое расстояние до следующей машины на главной дороге им комфортно иметь при заезде, можно получить разные картины при расчетах. Если водители «агрессивные» и эффективно перестраиваются, при одних и тех же значениях потоков *Q1* и *Q2* получим снижение среднего времени нахождения автомобиля в расчетной области. Однако при достаточно больших значениях потока Q1 водителям с второстепенной дороги, независимо от их стратегии, заехать на главную тяжело. В результате на второстепенной дороге образуется затор, а среднее время нахождения автомобилей, попадающих в расчетную область с второстепенной дороги, в системе стремится к общему времени расчета. Однако в реальности даже при интенсивном потоке по главной дороге водители находят способ заехать на нее и не стоят на въезде бесконечно.

При включении алгоритма «вежливый водитель» картина меняется. Результат моделирования представлен на рис. 18. Квадратами обозначены автомобили; красные – те, что въехали по главной дороге, синие – те, что въехали по второстепенной. Движение происходит слева направо.



18(а) 18(б)

Рис. 18. Въезд на главную дорогу – результат моделирования. Слева (*18(а)*) – без «вежливых водителей», справа (*18(б)*) – с «вежливыми водителями»

Как видно из рисунков, затор на въезде при включении алгоритма становится меньше. Также это подтверждает расчет среднего времени нахождения автомобилей с второстепенной дороги в системе.

Возникает вопрос о том, снижает ли наличие «вежливых» водителей общее среднее время нахождения автомобилей в системе (всех, а не только тех, кто должен уступать дорогу). Расчеты показывают, что не всегда – это зависит от выбора параметров моделирования, а конкретно: скоростей V^*, V^{**}. Это означает, что данные параметры необходимо тщательно подбирать при калибровке модели под реальные условия при решении практических задач моделирования.

4.3 Алгоритм a14 снижения скорости при наличии затора

Для того чтобы объезд препятствий происходил эффективнее, при скоплении автомобилей в ряду с препятствием водители из других рядов реагируют на это, снижая скорость, упрощая тем самым перестроение. Такое поведение можно наблюдать на дороге: видя аварию или какой-либо неподвижный объект на дороге, водители замедляются из соображений безопасности.

Данный алгоритм встроен в алгоритм *a2*.

Задаются счетчики *jcl* и *jcr* (jam count left, jam count right), которые изначально равны нулю. Для проверки необходимости снизить скорость выполняются следующие шаги.

1. Вычисление количества неподвижных автомобилей в соседних рядах справа и слева $k\pm1$: для всех m: $1+i \le m < \min(i+10, size)$, если $v_{m,k\pm1,j}=0$, то $jcl(r)=jcl(r)+1$ (здесь i – номер ячейки, k – номер ряда, j – номер дороги, *size* – длина расчетного участка в ячейках).
2. Вычисление расстояния до препятствия в соседних рядах (если оно есть): для всех m: $1+i \le m < \min(i+10+V_{max}, size)$, если $v_{m,k\pm1,j}=-10$, то расстояние до препятствия $d_{obst}l(r)=m-i$.

3. Если $d_{obst}l(r)\leq 10$ и $jcl(r)\geq 9$, то параметр, обозначающий наличие пробки перед препятствием *jam*=true.
4. На шаге 2 *a2* дополнительное условие снижения скорости: если *jam*=true, $v_{i,k,j}= \min (v_{i,k,j}-1, 0)$.

5. Модель транспортных потоков с «медленным стартом»

Модели с «медленным стартом» («slow-to-start») – класс моделей, где автомобили, начинающие движение, стартуют только при условии, что перед ними имеется более одной свободной ячейки (см., например, [66]). На первый взгляд, это условие не следует очевидно и однозначно из наблюдения за поведением водителей. Оно было включено для того, чтобы автомобили не разъезжались слишком быстро от места затора, что позволяет, в свою очередь, воспроизвести эффект гистерезиса при переходе от свободного к синхронизированному потоку: при одной и той же величине потока движение может быть как свободным, так и стесненным, в зависимости от случайных процессов в транспортном потоке.

5.1 Алгоритм a15 «медленный старт»

Данный алгоритм также встроен в общий алгоритм движения вперед *a2*.

Шаг 1 в *a2* дополняется условиями на медленный старт и выглядит так:

- Если $v_{ik}>0$, то $v_{ik}=\min(v_{ik}+1, V_{max})$;
- Если $v_{ik}=0$, то ищем расстояние до ближайшего автомобиля впереди d. Если $d>1$, то $v_{ik}=v_{ik}+1$;
  если $d=1$, то $v_{ik}=v_{ik}$.

Остальные шаги *a16* повторяют шаги 2-4 *a2*.

Данный алгоритм также можно применять при калибровке модели для наилучшего совпадения данных тестовых расчетов с экспериментальными данными.

6. Граничные условия в модели

Задание граничных условий при реализации модели требует отдельного внимания, поскольку перед добавлением автомобилей в расчетную область следует убедиться в том, что начальные ячейки свободны и на участке нет затора.

6.1 Алгоритм a16 задания граничных условий с возможностью образования очереди на въезд в расчетную область

На рис. 19 представлена блок-схема алгоритма задания граничных условий.

Рис. 19. Блок-схема алгоритма очереди на въезд в расчетную область

Поддержание необходимого значения входящего потока обеспечивается добавлением автомобилей каждый *n*-й шаг по времени, где *n* вычисляется в зависимости от заданного потока (на данной дороге, с данной целью). При скоплении автомобилей в расчетной области может возникнуть ситуация, когда на нужном шаге по времени автомобиль не может быть добавлен в расчетную область. Для учета таких автомобилей служит набор переменных «очередь» (q_{j1j2}, где *j1* обозначает дорогу, по которой въезжает автомобиль, а *j2* – его целевую дорогу). Соответственно, если при въезде первая ячейка оказалась занята, автомобиль попадает в очередь (соответствующее значение q_{j1j2} увеличивается на 1: $q_{j1j2} = q_{j1j2}+1$) На каждом шаге по времени проверяется наличие автомобилей в очереди ($q_{j1j2}>0$?). Если $q_{j1j2}>0$, автомобиль добавляется

в расчетную область, а значение q_{j1j2} уменьшается на 1 ($q_{j1j2} = q_{j1j2}$-1).

Заключение

В препринте представлен полный набор алгоритмов, вошедших в представленную модель на основе теории клеточных автоматов для описания дорожного движения на городской улично-дорожной сети. Модель разрабатывалась и усложнялась в течение нескольких лет автором препринта совместно с коллегами из ИПМ им. М.В. Келдыша РАН. Данный препринт подводит итог этой работе.

Модель реализована в качестве комплекса программ CAM-2D для высокопроизводительных вычислительных систем и апробирована при помощи тестовых расчетов и сравнения с экспериментальными данными.

Данная модель может служить основой для интеллектуальной транспортной системы мегаполиса.

Библиографический список

1. *S. Wolfram.* A new kind of science. Wolfram Media, 2005, 1197 p.
2. *Maerivoet S., De Moor B.* Cellular automata models of road traffic // Physics Reports, 2016, V. 419, pp. 1-64.
3. *B. Kerner, S. Klenov, M. Schreckenberg.* Simple cellular automaton model for traffic breakdown, highway capacity, and synchronized flow // Phys. Rev. E, 2011, v. 84, p. 046110.
4. *B. Kerner, S. Klenov, G. Hermanns, M. Schreckenberg.* Effect of driver over-acceleration on traffic breakdown in three-phase cellular automaton traffic flow models // Physica A: Statistical Mechanics and its Applications, 2013, v. 392, No.18, p. 4083-4105.
5. *T.P.E. Vranken, M. Schreckenberg.* Cellular Automata Intersection Model // Collective Dynamics, 2020, v. 5, p. 1-25.
6. *Чечина А.А., Герман М.С., Ермаков А.В., Трапезникова М.А., Чурбанова Н.Г.* Моделирование и визуализация потоков автотранспорта на элементах улично-дорожной сети с использованием комплекса программ CAM-2D // Препринты ИПМ им. М.В. Келдыша, 2016, № 124, 17 с.
7. *А.А. Чечина, Н.Г. Чурбанова, М.А. Трапезникова*. Сравнение воспроизведения пространственно-временных структур транспортных потоков при использовании различных способов осреднения данных // Математическое моделирование, 2021, т. 33, №1, с. 25-31.

   M.A. Trapeznikova, A.A. Chechina, N.G. Churbanova. Sravnenie vosproizvedeniia prostranstvenno-vremennykh struktur transportnykh potokov pri ispol’zovanii razlichnykh sposobov osrednenia dannykh // Matematicheskoe modelirovanie, 2021, v. 33, No.1, p. 25-31.

8. *Chechina A., Churbanova N., Trapeznikova M.* Modelling traffic on road junctions on parallel computing systems using cellular automata approach. // Int. Journal of Interactive Mobile Technologies (iJIM), 2020, Vol. 14, No.10, eISSN: 1865-7923. https://doi.org/10.3991/ijim.v14i10.14549. 2021
9. *K. Nagel, M. Schreckenberg* A Cellular automaton model for freeway traffic // J. Phys. I France, 1992, V. 2, pp. 2221-2229.

Оглавление

Введение 3
Построение многополосной модели на основе теории клеточных автоматов 3
Алгоритмы движения автомобилей 6
1. Общий алгоритм движения 6
1.1 Алгоритм a1- общий алгоритм смены полосы 6
1.2 Алгоритм a2 - общий алгоритм движения вперед 8
2. Достижение цели 9
3. Движение на элементах улично-дорожной сети 9
3.1 Алгоритм a3 перестроения с достижением цели (движение по участку дороги с выездом) 10
3.2 Алгоритм a4 прохождения четырехстороннего перекрестка со светофором 12
3.3 Алгоритм a5 – обеспечение безопасности на перекрестке 13
3.4 Алгоритм a6 прохождения Т-образного перекрестка со светофором ... 15
3.5 Алгоритм a7 прохождения участка с сужением/расширением 15
3.6 Алгоритм a8 въезда с второстепенной дороги на главную 16
3.7 Алгоритм a9 проезда участка с разворотом 17
3.8 Алгоритм a10 проезда участка с пешеходным светофором 17
3.9 Алгоритм a11 объезда широкого неподвижного препятствия 17
4. Различные стратегии поведения водителей 19
4.1 Алгоритм a12 «вежливый водитель» для смены полосы 21
4.2 Алгоритм a13 «вежливый водитель» для въезда с второстепенной дороги на главную 24
4.3 Алгоритм a14 снижения скорости при наличии затора 27
5. Модель транспортных потоков с «медленным стартом» 28
5.1 Алгоритм a15 «медленный старт» 28
6. Граничные условия в модели 28
6.1 Алгоритм a16 задания граничных условий с возможностью образования очереди на въезд в расчетную область 28
Заключение 30
Библиографический список 30