

ПРИМЕНЕНИЕ КЛЕТОЧНЫХ АВТОМАТОВ ДЛЯ МОДЕЛИРОВАНИЯ ТРАНСПОРТНЫХ ПОТОКОВ

Г. А. Омарова, Г. Ю. Казанцев

Институт вычислительной математики и математической геофизики СО РАН,
630090, Россия, Новосибирск

УДК 519.179.2+512.23

Рассмотрена модель Нагеля–Шрекенберга. Разработана схема работы клеточных автоматов для Т-образного перекрестка.

Ключевые слова: модель, клеточный автомат, расстояние, скорость, ускорение, регулярная решетка.

Nagel–Schrekenberg's model is considered. The formalized diagram of operation of cellular automatic machines for the T-shaped intersection is developed.

Key words: Model, cellular automata, distance, speed, acceleration, regular grid.

Введение. Процесс моделирования транспортных потоков позволяет получить информацию, которая помогает принимать решение о дальнейшем развитии и управлении транспортной системой. В частности, моделирование позволяет определить потребности в постройке новых или расширении старых дорог, транспортных систем и терминалов, размещении новых светофоров и дорожных знаков или изменении расположения уже имеющихся. Грамотное планирование развития дорожной сети на основе полученных данных позволяет снизить нагрузку на транспортную сеть, избежать пробок и уменьшить среднее время в пути для водителей. Существует ряд проектов, направленных не только на моделирование транспортных потоков, но и на предсказание возможных ситуаций на дороге и минимизацию нагрузок на транспортную сеть [1].

Исследование и моделирование транспортных потоков с помощью микро- и макромоделей позволяет определить объем загрузки и интенсивность потоков [2]. Современные тенденции в этой области направлены на создание специализированных информационных систем, управляющих конкретными городскими регионами и учитывающих его специфику. В мировой практике для решения подобных проблем применяются методы математического моделирования различной сложности и создаются компьютерные стенды, демонстрирующие полученные решения на макетах реальных дорог и перекрестков. Специализированные информационные системы современных городских мегаполисов способны, например, решать такие задачи: оценивать интенсивность потоков транспорта на магистралях и управлять скоростью движения с учетом времени суток, погодных условий и времен года; оценивать экологическую обстановку в районе управления и давать рекомендации по ее улучшению. Построение подобных информационно-управляющих систем требует разработки соответствующих аналитических и программных моделей для оценки принимаемых решений.

Существует множество моделей транспортных потоков, основанных на различных подходах [1–3]. Одним из них является использование клеточных автоматов [4–6, 8]. В предла-

гаемом исследовании рассматривается клеточно-автоматная модель, отличающаяся тем, что длины клеток различны и рассматривается Т-образный перекресток.

Дальнейшее изложение организовано следующим образом. В первом параграфе даются общие сведения о клеточных автоматах, во втором рассматривается модель Нагеля–Шрекенберга, в третьем рассматривается построение клеточного автомата для Т-образного перекрестка и в последнем параграфе описание реализации на языке программирования C++.

1. Клеточные автоматы. Концепция клеточных автоматов (Cellular Automata, CA) была предложена Джоном фон Нейманом в начале 50-х годов при разработке теории самовоспроизводящихся систем. Классический CA — это двумерная сетка произвольного размера. Состояние сетки обновляется с течением времени, а состояние каждой конкретной ячейки в следующий момент времени зависит от состояния ее ближайших соседей (шаблон соседства, то есть набор клеток, считающихся соседними для клетки, может быть различным в разных моделях) и, возможно, от ее собственного состояния на текущей итерации. Количество возможных состояний клетки конечно, а обновление состояния сетки происходит одновременно [4].

Свойства CA:

1) *Однородность системы.* Ни одна область неотличима от другой по каким-либо особенностям правил. Впрочем, так как в практической реализации количество ячеек ограничено объемом памяти, то ячейки на краю области могут иметь отличающиеся правила из-за отсутствия некоторых соседей, либо иметь измененный шаблон соседства.

2) *Локальность правил.* Только клетки окрестности и сама клетка могут влиять на состояние клетки.

3) *Конечность множества возможных состояний клетки.* Данное условие необходимо, чтобы для получения нового состояния требовалось конечное число операций.

4) *Одновременный переход в новое состояние для всех клеток.* Значения во всех клетках меняются одновременно в конце очередной итерации, а не по мере вычисления. В противном случае от порядка вычислений зависит результат итерации.

Впрочем, для решения конкретных практических задач некоторые из свойств могут быть отброшены. В модели фон Неймана клетки составляют бесконечную прямоугольную решетку, у каждой клетки соседями считаются только непосредственно прилегающие клетки. Изменения состояний клеток происходят одновременно, и правила всех клеток одинаковы.

2. Моделирование транспортного потока. Впервые идея применения клеточных автоматов для моделирования транспортных потоков была предложена в работе [5]. Однако активная разработка и исследования в данном направлении начались только после публикации К. Нагеля и М. Шрекенберга (Nagel–Schreckenberg) [6].

Пусть x_n и v_n — координата и скорость n -го автомобиля, $d_n = x_{n-1} - x_n$ — дистанция до лидирующего автомобиля. В CA-модели на каждом шаге $m \rightarrow m + 1$ состояние всех автомобилей в системе обновляется в соответствии со следующими правилами:

1) *Ускорение.* Выражает стремление к увеличению скорости, но в то же время без превышения максимально допустимой скорости v_{max}

$$v_n(m+1) = \min(v_n(m) + 1, v_{max}).$$

2) *Торможение.* Условие, позволяющее избежать столкновений с впереди идущим автомобилем:

$$v_n(m+1) = \min(v_n(m), x_n(m) - x_{n-1}(m) - d),$$

где d — расстояние между соседними автомобилями.

3) *Случайные возмущения.* Данное условие характеризует случайные различия в поведении водителей.

$$v_n(m+1) = \begin{cases} \max(v_n(m) - 1, 0), & p, \\ v_n(m), & 1 - p. \end{cases}$$

4) *Движение.*

$$x_n(m+1) = x_n(m) + v_n(m).$$

Приведенный набор правил является минимальным набором, необходимым для воспроизведения базовых свойств транспортного потока. Численные эксперименты показывают, что поток является устойчивым при малых плотностях и теряет устойчивость при высоких плотностях [3]. Также необходимо заметить, что поток остается устойчивым для всех значений плотности при $p = 0$ [6, 9]. Это обстоятельство можно рассматривать как теоретический недостаток клеточно-автоматных моделей по сравнению с рассмотренными ранее моделями, в которых флуктуации играют роль начального толчка, а дальнейшее развитие затора объясняется неустойчивостью равновесного решения. К несомненным достоинствам клеточных автоматов можно отнести сравнительную простоту моделирования транспортных потоков, что является немаловажным при переходе к многорядным моделям, т. к. такой переход значительно усложняет конструкцию модели. Что касается приведенных выше недостатков, то после [6, 9] был создан ряд моделей клеточных автоматов, которые исправляют этот и другие недостатки модели.

2.1. *Расширение модели клеточных автоматов для двумерного случая.* Многорядные микромодели позволяют получить более реалистичную картину при моделировании транспортных потоков [1]. Обобщение однорядной модели клеточных автоматов на двумерный случай для возможности моделирования многополосного движения. В такой модели трасса представляет собой двумерную решетку, в которой количество ячеек в поперечном направлении соответствует числу полос трассы. В модели разрешены перестроения машин из полосы в полосу и обгоны. Процесс обновления состояний ячеек делится на два подшага:

- 1) для каждой машины выясняется возможность и необходимость смены полосы. Производится смена полосы. Этот подшаг выполняется параллельно для всех машин;
- 2) производится движение вперед по каждой полосе по правилам однополосного движения.

Смена полос должна происходить за один временной шаг. Если в одном направлении существуют больше, чем две полосы, то может возникнуть конфликт, когда две машины с крайних полос желают сместиться в среднюю и занять одну и ту же ячейку. Такой конфликт легко преодолеть, если разрешить перестроение вправо только на четных шагах, а влево — только на нечетных.

Для смены полос существуют несколько причин: на соседней полосе выше скорость движения либо меньше плотность, перестроение на соседнюю полосу необходимо для успешного достижения цели движения. С другой стороны, перед сменой машиной полосы необходимо проверить, выполнены ли условия безопасности. При выполнении всех этих условий машина совершает смену полосы с некоторой заданной вероятностью [1].

Добавление полосы встречного движения несильно влияет на модели, основные изменения касаются правил перестроения с выездом на встречную полосу. В работе [8] рассматривается модель встречного движения с подобными правилами и демонстрируется ее адекватность эмпирическими данными.

2.2. Модель Т-образного перекрестка.

Постановка задачи. Необходимо разработать модели СА для моделирования организации движения на Т-образном перекрестке согласно заданному набору условий проезда перекрестка (знаки приоритета, разрешенные направления движения).

Рассмотрим участок главной дороги, имеющей по одной полосе в каждом направлении, и участок второстепенной, примыкающей к главной, и тоже имеющей по одной полосе в каждом направлении. Для упрощения модели считаем, что все машины движутся с одной скоростью. В искомой модели клетки имеют разные размеры. Большая часть клеток имеет размер одной машины, но непосредственно на самом перекрестке клетки имеют размер в половину машины.

Определим СА как структуру $CA = \langle A, R, X, T \rangle$, где $A = \{0,1\}$ — множество состояний клетки, R — множество правил изменения своего состояния или сохранения прежнего для каждой клетки автомата в зависимости от ее месторасположения, $X = \{1,2,...a\}$ — множество транспортных средств, T — множество тактов времени. Таким образом, каждый СА имеет свой набор правил (и эти правила тем сложнее, чем больше конфликтных направлений, определенных входным набором правил проезда перекрестка), может использовать данную клетку.

Каждая клетка либо пуста, либо содержит машину. На перекрестке также важной является принадлежность двух смежных непустых клеток одной машине. Для этого каждой машине присвоен уникальный идентификатор. Правила всех клеток имеют общий вид — если клетка не пуста, проверяются важные клетки, и, в случае выполнения условий, машина убирается из текущей клетки и помещается в следующие. На клетках половинного размера размещаются основная (передняя) и дополнительная (задняя) части машины. Каждая часть использует различные наборы правил. Правила дополнительной части не зависят от направления, важными являются все смежные клетки, а условием является отсутствие непустой клетки с идентификатором той же машины. Клеток, которые являются следующими, в правилах нет, и новых заполненных клеток они не порождают.

У всех остальных клеток условием является пустота важных клеток. Значения клеток $\{0,1\}$ — пустая или занятая клетки соответственно, дополнительно клетка может быть отмечена как x — важная клетка, n — следующая.

Пример выполнения правил приведен на рис. 1, так как данный маневр сложный, поэтому показано несколько ходов поворота.

3. Реализация. Клеточная структура в программе представляет собой массивы и несколько отдельных переменных, являющихся объектами класса `cell`.

```
class cell
{
public:
    cell();
    cell(vector <rule> r);
    virtual ~cell();
    string get_machineid();
    void add_rules(vector <rule> r);
```

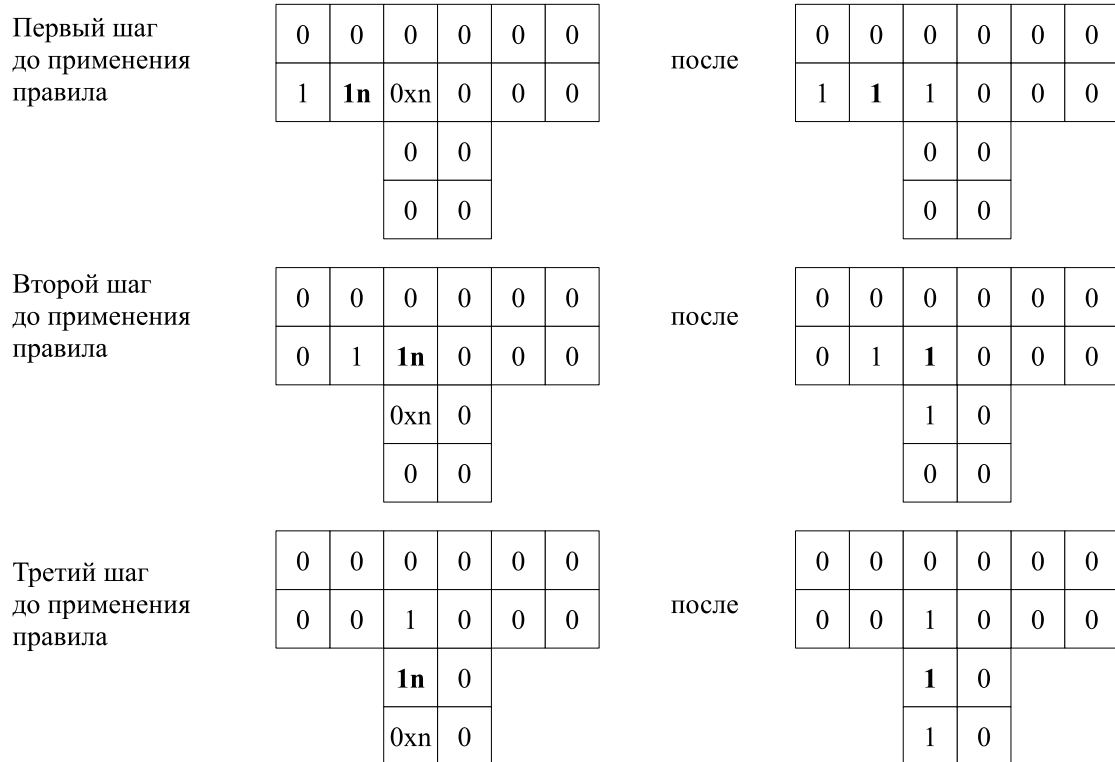


Рис. 1. Поворот направо с главной дороги

```

void place_machine(string m, int i);
void move();
bool empty();
void del();
void move_back();
private:
    string machine;
    vector<rule>rules;
};

```

Из представленного объявления класса видно, что у каждой клетки есть два поля. Поле `machine` может содержать „Empty“ для пустой машины и слово вида „id“, „d“, „dest“. При этом „id“ имеет вид „&“ или „&B“, а „dest“ — W/S/E или WB/SB/EB для основной или дополнительной части машины. Поле `rules` — это массив, хранящий правила функционирования клетки. Каждое правило — это объект типа `rule`, имеющий следующий вид:

```

struct rule
{
    string destination;
    vector<cell*>important;
    vector<cell*>next;
};

```

Поля `important` и `next` — это важные и следующие клетки правила соответственно, определяющие, как функционирует правило. Поле `destination` — это одна или две буквы,

данное поле используется для выбора правила, соответствующего направлению движения машины, и для использования правила необходимо, чтобы его поле `destination` совпало с „`dest`“ машины, находящейся в этой клетке. Методы `place_machine` и `del` позволяют помещать машину в клетку и убирать ее оттуда в функциях, не принадлежащих классу `cell`.

Метод `add_rules` используется для добавления правил в уже созданную клетку. При этом проверяется, было ли у клетки правило с совпадающим полем `destination`, это запрещено, так как создает неоднозначность в поведении модели, поэтому попытка добавления такого правила приводит к исключению, сообщаящему, какое именно правило конфликтует.

Методы `get_machineid` и `empty` предоставляют информацию о состоянии клетки для объектов, не принадлежащих классу `cell`. Метод `empty` просто сообщает, является ли клетка пустой, в то время как метод `get_machineid` возвращает „`id`“ машины или „`Empty`“.

Основными методами для функционирования являются `move` и `move_back`. Метод `move` ищет среди правил клетки то, чье направление совпадает с направлением движения машины, после чего проверяет его и, если оно выполнено (общий вид правил и условие выполнения были указаны выше), перемещает машину в следующие клетки. Метод вызывается для всех клеток модели, он может аварийно завершиться, если в клетке недостаточно правил и отсутствует правило, соответствующее направлению движения машины. При аварийном завершении выбрасывается исключение, содержащее информацию о том, какое правило ожидалось машиной.

Метод `move_back` похож на метод `move` и с одними и теми же правилами работает одинаково, но в отличие от него применяет только правила для дополнительных частей. По этой причине его нет необходимости вызывать для больших клеток.

В классе `roads` хранится и инициализируется клеточная структура. Для клеток на прямых участках дорог правила задаются автоматически благодаря их простоте, для клеток перекрестка, имеющих половинный размер, правила были написаны вручную. Также класс `roads` имеет метод `Motion`, выполняющий шаг итерации и вызывающий в описанном выше порядке в клетках метод `move`, а после его вызова для всех клеток перекрестка вызывающий метод `move_back`.

Заключение. В ходе работы проведено подробное исследование клеточных автоматов для моделирования транспортных потоков. Разработана и построена формализованная схема работы клеточных автоматов для перекрестка. Данная модель удовлетворяет правилам дорожного движения в идеальных условиях, когда все машины движутся с фиксированной скоростью и все водители соблюдают правила. Для визуализации модели на разных шагах итерации используется графический интерфейс, созданный с использованием библиотеки WinAPI.

Список литературы

1. КРАВЧЕНКО П. С., ОМАРОВА Г. А. Микроскопические математические модели транспортных потоков. Аналитический обзор // Проблемы информатики. 2014. № 1. С. 71–78.
2. ГАСНИКОВ А. В. и др. Введение в математическое моделирование транспортных потоков. Москва: МФТИ, 2010.
3. ШВЕЦОВ В. И. Математическое моделирование транспортных потоков // Автоматика и Телемеханика 2003. № 11. С. 3–46.
4. ДЖ. ФОН НЕЙМАН. Теория самовоспроизводящихся автоматов. М.: „Мир“. 1971.

5. CREMER M., LUDWIG J. A fast simulation model for traffic flow on the basis of Boolean operations // Math. Comp Simul. 1986. V. 28. P. 297–303.
6. NAGEL K., SCHRECKENBERG M. A cellular automation model for freeway traffic // Phys. I France. 1992. V. 2. P. 2221–2229.
7. CHANDLER R. E., HERMAN R., MONTROLL E. W. Traffic dynamics: Studies in car following // Operations Research. 1958. V. 6. P. 165–184.
8. SIMON P. M., GUTOWITZ H. A. A Cellular Automaton Model for Bi-Directional Traffic // Phys. Rev. 1998. E. 57, 2441.
9. NAGEL K., HERMANN H. J. Deterministic models for traffic jams // Physica A. 1993. V. 199. P. 254–269.

*Омарова Гульзира Алимовна — канд. физ.-мат. наук, науч. сотр. Института
вычислительной математики и математической геофизики СО РАН;
e-mail: gulzira@rav.sccc.ru*

*Казанцев Григорий Юрьевич — студент НГУ;
e-mail: kazancev.grigorij@gmail.com*

Дата поступления — 25.06.2015