

---

---

**ИСКУССТВЕННЫЙ  
ИНТЕЛЛЕКТ**

---

---

УДК 004.8

**ИНТЕЛЛЕКТУАЛЬНАЯ СИСТЕМА ДЛЯ ПОШАГОВОГО  
УПРАВЛЕНИЯ ПРИ ОПЕРАТИВНОМ  
ПЕРЕСТРОЕНИИ РАСПИСАНИЯ**

© 2021 г. Р. А. Горбачев<sup>a,\*</sup>, Е. М. Захарова<sup>a,\*\*</sup>,  
И. С. Макаров<sup>b,\*\*\*</sup>, В. И. Цурков<sup>b,\*\*\*\*</sup>

<sup>a</sup> Федеральное государственное автономное образовательное учреждение высшего образования  
“Московский физико-технический институт (национальный исследовательский ун-т)” МФТИ,  
Долгопрудный, МО, Россия

<sup>b</sup> Федеральный исследовательский центр “Информатика и управление” РАН (ФИЦ ИУ РАН),  
Москва, Россия

\*e-mail: gorbachev.ra@mipt.ru

\*\*e-mail: zakharova.em@mipt.ru

\*\*\*e-mail: i.s.m.mipt@yandex.ru

\*\*\*\*e-mail: tsur@ccas.ru

Поступила в редакцию 04.02.2021 г.

После доработки 05.02.2021 г.

Принята к публикации 29.03.2021 г.

Приведено описание применения искусственного интеллекта при разработке систем поддержки принятия решения, которые применяются для различных типов транспорта. Эти системы направлены для перестройки расписания движения объектов по причине непредвиденных отклонений заранее запланированных графиков. Используется машинное обучение нейросети с помощью генетического алгоритма. При этом минимизируется функционал, который характеризует отклонение от заданного расписания.

DOI: 10.31857/S0002338821050103

**Введение.** С каждым годом транспортное движение становится все более интенсивным. Увеличивается плотность перевозок как в грузовом, так и в пассажирском движениях. Вследствие этого повышаются требования к исполнению запланированного плана движения, и любое отклонение от него может привести к значительным неустойкам, например, увеличению задержки поездов, их отмене и т.д. [1]. Поэтому разработка систем управления в области движения является актуальным и перспективным направлением. В данной сфере существуют высокие требования к безопасности движения, которые приводят к дополнительным трудностям.

В работе приведено применение искусственного интеллекта, в частности нейронных сетей с использованием генетического алгоритма для обучения.

Основная концепция сводится к задаче управления взаимодействием агентов в динамической среде. Решение является пошаговым. В нем участвует главный агент (диспетчер), взаимодействующий с группой других агентов (движущихся объектов) в динамической среде [2]. Главная решаемая задача — достижение всеми агентами конечных точек их маршрутов с учетом заданных ограничений. При этом отклонение по времени от заданного плана движения для всех агентов должно быть минимальным.

**1. Анализ текущих разработок.** В настоящее время для решения подобных задач разрабатываются алгоритмы, в том числе и с использованием машинного обучения. Такой подход позволяет искать оптимальное решение на несколько шагов вперед.

Теперь несколько слов о библиографии в связи с автоматизацией транспортного диспетчера.

Построение расписаний, схем движения и поиск вариантов решения в случае возникновения конфликтных ситуаций происходит во многом опытно-экспертным путем. В настоящее время разработка и внедрение систем для автоматизации данных процессов с помощью искусственного интеллекта только получает свое развитие. Одной из последних публикаций по внедрению

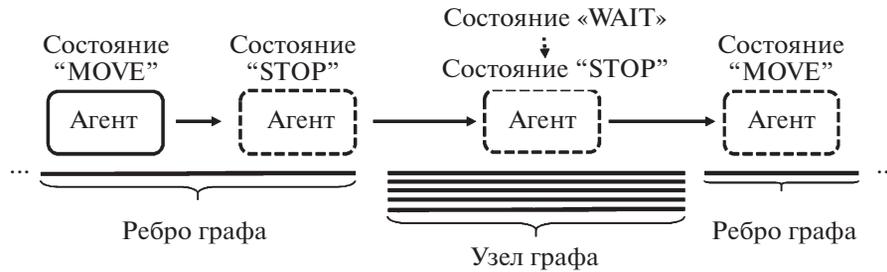


Рис. 1. Математическая модель движения

подхода глубокого обучения для изменения расписаний при сбоях в высокоскоростном движении является работа [3]. Подход, представленный в данной статье, позволяет работать только со скоростным движением. В то же время существует несколько типов движения, отличающихся приоритетом при оперативном пропуске объектов, а также технологическими особенностями движения.

Подход, описанный в [4], позволяет увеличить пропускную способность за счет использования при построении графика генетического алгоритма. Данное решение применяется как плановый график движения. Однако такой подход не предназначен для оперативного разрешения конфликтов. К тому же при их разрешении наиболее приоритетным направлением поиска решения является в первую очередь именно приближение графика исполненного движения к плановому с минимальными отклонениями от него, а не создание нового графика.

В [3] представлен подход, наиболее полно приближенный к поставленной задаче диспетчерского управления. Однако, она так же, как и [5], решена только для одного типа движения. Подобное ограничение упрощает поиск решения, так как в такой модели не существует правил и ограничений по различным видам движения, например приоритет пропуска поездов или ограничения на максимально разрешенную скорость в зависимости от типа движения.

Подход, описываемый ниже, позволяет работать с несколькими видами движения и соответствующими ограничениями для них.

**2. Постановка задачи и основные конструкции.** Для поиска решений при возникновении конфликтных ситуаций необходимо реализовать модель этого движения.

Граф, по которому должны двигаться агенты, представляется как совокупность сегментов-ребер и узлов, которые являются совокупностью параллельных сегментов-ребер. На одном сегменте в один момент времени может находиться только один агент. У сегмента есть следующие параметры: уникальный идентификатор, имя, длина и массив идентификаторов смежных сегментов. Агент задается как объект, имеющий следующие параметры: уникальный идентификатор, тип движения, длина, скорость, приоритет, координаты и маршрут. Маршрут в свою очередь содержит информацию о временах прибытия и стоянки на ребрах узлов и список всех узлов, на которых должен останавливаться текущий агент. Объекты могут иметь следующие статусы: "Move" (агент движется), "Wait" (агент стоит), "Stop Move" (агент закончил движение) или "Stop Wait" (агент закончил стоянку). Агент может получать следующие команды: "Move", "Continue", "Wait". Команды "Move" и "Wait" могут быть выполнены только в том случае, если агент находится в состоянии "Stop". Состояние "Stop" подразумевает либо окончание плановой стоянки агента, либо окончание движения агента по перегону.

На рис. 1 представлено графическое описание предложенной модели.

В данном подходе принятие решений по управлению движением агентов, движущихся в дискретном пространстве в виде планарного графа на участке, будет реализовано на основе интеллектуального управления, для чего была построена полносвязная искусственная нейронная сеть, обученная с использованием генетического алгоритма. Нейронная сеть состоит из трех слоев – входного, выходного и скрытого. Размерность входного слоя составляет порядка 392 нейронов, размерность выходного слоя – 102 нейронов. Количество нейронов в скрытом слое – 392.

На рис. 2 представлена архитектура предполагаемой нейронной сети.

На ее вход передается информация о сегментах графа участка, а также агенты и их маршруты. Результатом работы нейронной сети является вектор команд ожидающим агентам.

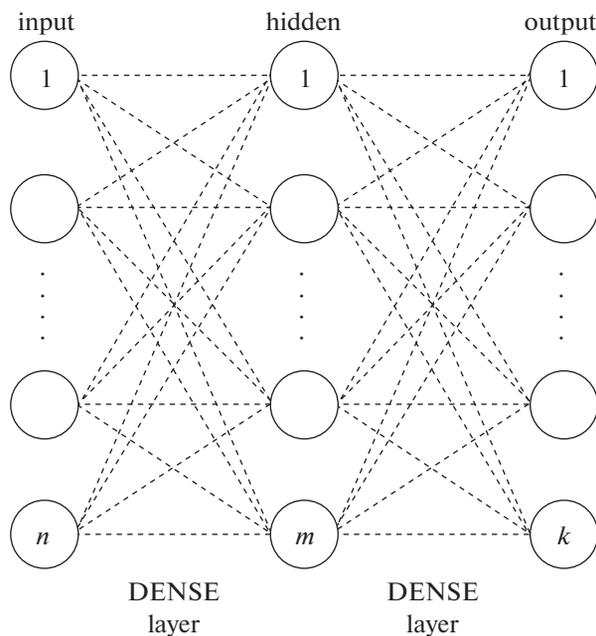


Рис. 2. Архитектура предполагаемой нейронной сети

В работе была построена тестовая модель движения, включающая в себя граф, который в свою очередь состоит из ребер и узлов, а также агентов различных типов, следующих по построенным маршрутам. Весь граф участка, на котором происходит движение, разбит на структурные элементы.

Вводится понятие “инфраструктурный сегмент”. Это такой фрагмент сети, на котором в конкретный момент времени может находиться только один агент. Узел может включать в себя несколько ребер. Каждое такое ребро из совокупности ребер узла является инфраструктурным сегментом. Каждый перегон состоит из некоторого количества блок-участков. Каждый блок-участок является инфраструктурным сегментом. Такие разбиения обеспечивают комфортное формирование входного вектора признаков для нейронной сети. Для каждого инфраструктурного сегмента известна его длина, направленность и смежные с ним инфраструктурные сегменты. Для каждого агента известны все необходимые параметры, такие, как масса, длина и мощность, а также маршрут, по которому он следует. В маршруте содержится информация заданного плана движения: пункт назначения, список остановок, времена прибытия и длительность стоянок. Помимо этого, для каждого инфраструктурного сегмента и каждого агента задается “состояние”. Данная характеристика отображает как текущее состояние инфраструктурного сегмента или агента, так и произошедшие с ними непредвиденные ситуации. Действия главного агента заключаются в том, что он на каждом шаге отдает команды другим агентам таким образом, чтобы они следовали по соответствующим им маршрутам и не нарушали заданные технологические ограничения, связанные с движением по этому маршруту. Непредвиденные ситуации генерируются случайным образом с частотой, задаваемой пользователем в качестве входного параметра. Существует фиксированный набор классов непредвиденных ситуаций, например повреждение путей инфраструктурного сегмента или недееспособность одного из агентов. Для каждого типа непредвиденной ситуации задается его продолжительность. Поиск решения дискретизирован по времени и состоит из “шагов”. Шаг описывает отрезок времени, не содержащий “событий”. Событием называется изменение состояния инфраструктурного сегмента или агента.

В свою очередь событие может быть нештатным или сопутствующим движению агента. Примером такой ситуации является изменение характера движения, основная суть которого заключается в следующем. Пока агент движется по инфраструктурному сегменту или стоит на нем по расписанию, если этот инфраструктурный сегмент служит ребром узла, он находится в одном из следующих состояний – “движется” или “стоит” соответственно.

Однако когда агент доезжает до конца инфраструктурного сегмента или провел нужное количество времени на запланированной остановке, его состояние меняется на “закончил движение”

и “закончил стоянку” соответственно, что и является “запланированным” событием для агента. Это означает, что данный агент готов получить следующую команду и начать движение на указанном инфраструктурном сегменте или продолжать стоять до следующего события. Равномерная дискретизация временной шкалы по минутам была отвергнута по причине слишком мелкой гранулярности шагов, приводящей к их однообразию.

В случае задачи поиска разрешения конфликтных ситуаций с учетом ограничений на соблюдение маршрутов движения агентов главный агент должен решать следующую оптимизационную задачу для всех участников движения:

$$R = \sum_{i=1}^N \sum_{j \in W_i} \omega_i (v_1 \Delta T_{ij} + v_2 \Delta A_{ij} + \dots) \rightarrow \min, \quad (2.1)$$

где  $i$  – индекс агента;  $j$  – индекс участка из множества  $W_i$  перегонов агента  $i$ ;  $\omega_i$  – весовой коэффициент агента  $i$  на основе его типа, задается вручную для каждого агента;  $\Delta T_{ij}$  – изменение времени движения агента  $i$  на участке  $j$  по сравнению с заданным планом движения;  $\Delta A_{ij}$  – изменение затрат энергии агента  $i$  на участке  $j$  по сравнению с заданным планом движения;  $v_k$  – преобразующие коэффициенты линейной комбинации – в текущей версии  $v_k = 1, \forall k > 1$ . Таким образом учитывается только временное отклонение от заданного плана движения, однако допускается также учет энергоэффективности и других параметров.

Система поддержки принятия решений главного агента по управлению движением других агентов основывается на подходе интеллектуального управления с помощью полносвязной искусственной нейронной сети с несколькими скрытыми слоями [1, 6]. Вектор входных параметров  $V_{in}$  формируется на основе информации по каждому инфраструктурному сегменту сети и выглядит следующим образом:

$$V_{in} = [S_{T_1}, S_{W_1}, R_x, 0, S_{W_2}, 0, \dots, S_{T_k}, S_{W_n}, R_x, \dots]. \quad (2.2)$$

Таким образом вектор  $V_{in}$  состоит из  $N$  троек чисел  $S_{T_k}, S_{W_n}, R_x$ , где  $N$  – количество инфраструктурных сегментов в сети; второе число  $S_{W_n}$  в каждой тройке – состояние соответствующего инфраструктурного сегмента; первое  $S_{T_k}$  и третье  $R_x$  числа относятся к агенту, если таковой присутствует на данном инфраструктурном сегменте, они имеют ненулевые значения и задают состояние агента и индекс его маршрута соответственно. Дополнительную информацию об агенте можно получить, зная его маршрут. Размерность вектора  $V_{in}$  равна  $3N$ . Данный подход к формированию вектора  $V_{in}$  обеспечивает ему фиксированную размерность при любом количестве агентов в системе. Вектор выходных параметров  $V_{out}$  содержит информацию о командах, отдаваемых каждому агенту. В системе используются три типа команд:

- “continue”, указывающая агенту, что тот должен сохранять свое состояние до следующего события, т.е. продолжать двигаться или стоять, если он движется или стоит соответственно;
- “move I”, указывающая агенту, который завершил движение или стоянку на текущем инфраструктурном сегменте, следующий инфраструктурный сегмент, на который он должен начать движение;
- “dummy” – служебная команда-заглушка, используемая, если на инфраструктурном сегменте отсутствует агент.

В векторе  $V_{out}$  применяется One-Hot-Encoding, за счет чего его размерность без дополнительных преобразований пропорциональна  $N^2$ , однако может быть амортизирована до  $mN$  за счет учета ограничений команды “move”, где  $m$  – максимальное по сети количество смежных инфраструктурных сегментов. В данном подходе процесс обучения нейронной сети – это обучение оптимальности, т.е. обучение давать агентам команды, обеспечивающие наилучшее решение оптимизационной задачи. Для этого был реализован генетический алгоритм [7, 8]. Структура процесса обучения выглядит следующим образом.

**Шаг 1.** Создается начальная популяция из  $S = 100$  особей. Каждая особь представляет из себя отдельный экземпляр нейронной сети со случайными равномерно распределенными весами связей.

**Шаг 2.** Выполняется подготовка для естественного отбора особей в следующее поколение. Для этого осуществляется ранжирование всех особей по результатам оценки (формула (2.1)) за

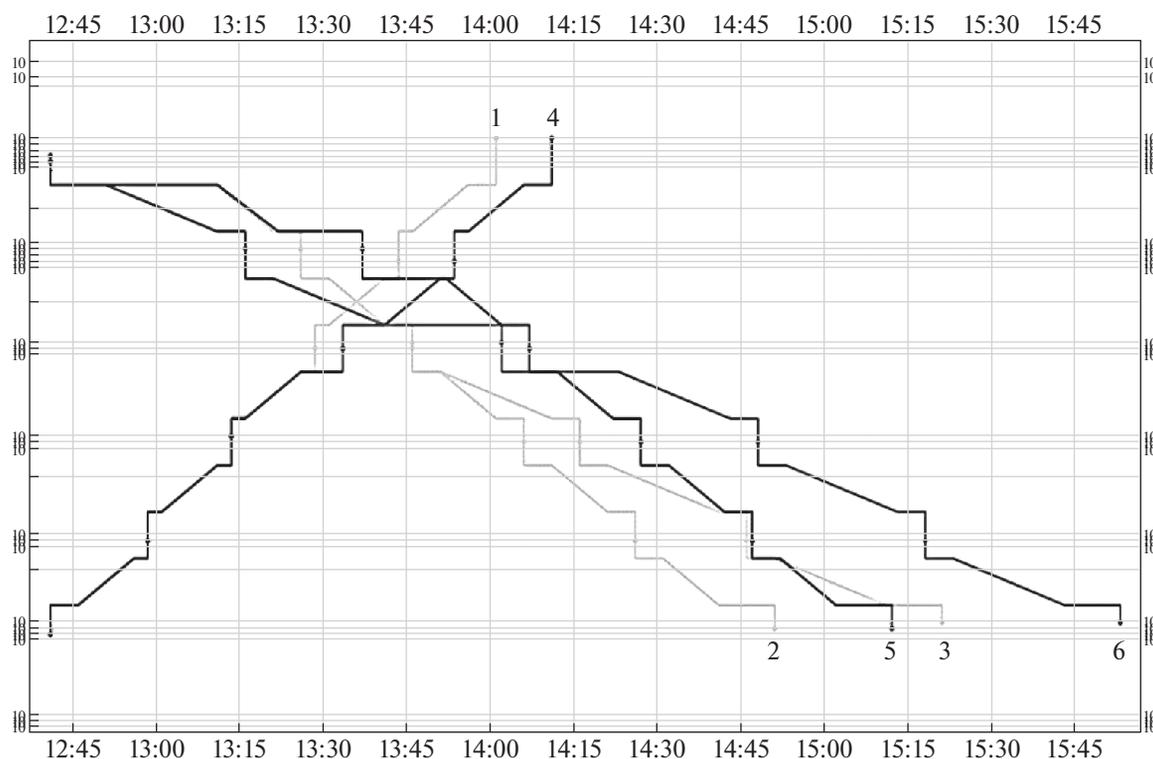


Рис. 3. Пример работы системы

$k = N \text{ avg}_i \text{Dim}(W_i)$  шагов, где  $N$  – текущее количество агентов в сети,  $\text{avg}_i \text{Dim}(W_i)$  – среднее количество инфраструктурных сегментов на маршруте.

Шаг 3. Генерируется следующее поколение: top 40% – прямое копирование, top 10% – кроссовер, random 30% – кроссовер, random 20% – прямое копирование, а также выполняются случайные мутации не более 1% весовых коэффициентов особей.

Шаг 4. Процесс повторяется с шага 2 до достижения заданного условия останова.

**3. Применение к реальному движению.** Модель поведения диспетчера реализована с помощью полносвязной нейронной сети, обученной с использованием генетического алгоритма. В процессе обучения создается несколько экземпляров нейронной сети, из которых выбирается тот, который обеспечивает наилучшее управление движением поездов с соответствию с приведенным функционалом. В процессе работы генетического алгоритма при обучении нейронной сети выполняются параллельные вычисления в многопроцессном режиме.

Для тестирования интеллектуальной системы был выбран линейный участок на графе железнодорожной сети Монголии. Он включает в себя шесть станций и пять перегонов между ними. Каждая станция состоит из трех или более станционных путей. Инфраструктурным сегментом считается станционный путь или перегон между станциями. Для обучения системы используются данные нормативного графика с конфликтами движения, которые возникают в случае, когда два агента претендуют на один и тот же путь. Результатом обучения являются весовые коэффициенты нейронной сети. При обучении минимизируется функционал отклонения от нормативного графика движения и окончательно выбираются упомянутые веса.

На рис. 3 представлен результат работы системы. По вертикальной оси отмечены станции выбранного участка железнодорожной сети Монголии и их станционные пути. На горизонтальной оси задана абсолютная временная шкала. Наклонные отрезки графиков соответствуют движению поездов по перегонам. Горизонтальные и вертикальные отрезки обозначают пребывание поездов на станционных путях.

Номерами 1–3 обозначены траектории планового графика движения с конфликтами. Под номерами 4–6 обозначены полученные системой траектории, где траектория 4 – график движения поезда с максимальным приоритетом, траектория 6 – с наименьшим. Из графика видно, что

отклонение поезда с максимальным приоритетом является минимальным по сравнению с другими поездами.

**Заключение.** В работе описано применение методов искусственного интеллекта для решения конфликтов движения. Дальнейшее развитие предложенного подхода заключается в следующем. Необходимо использовать альтернативные архитектуры нейронных сетей, такие, как, например, CNN и LSTM, с целью повышения быстродействия решения и более близкому приближению к оптимуму. Помимо генетического алгоритма возможно применение других методов оптимизации. Интересно также расширить количество агентов и добавить им дополнительные характеристики.

#### СПИСОК ЛИТЕРАТУРЫ

1. *Bettinelli A., Santini A., Vigo D.* A Real-time Conflict Solution Algorithm for the Train Rescheduling Problem // *Transportation Research. Pt B: Methodological.* 2017. V. 106. P. 237–265.
2. *Zakharova E.M., Paschenko F.F., Takmazyan A.K., Minashina I.K., Kuznetsov N.A.* Intelligent Control Systems for Maintenance of Railway Rolling Stock // *Proc. 11th IEEE Intern. Conf. on the Application of Information and Communication Technologies (AICT2017).* Moscow. 2017. V. 1. P. 423–425.
3. *Ning L., Li Y., Zhou M., Song H., Dong H.* A Deep Reinforcement Learning Approach to High-speed Train Timetable Rescheduling under Disturbances // *IEEE Intelligent Transportation Systems Conf. (ITSC).* Auckland, New Zealand, 2019.
4. *Jih-Wen Sheu, Wei-Song Lin.* Designing Automatic Train Regulation for MRT System by Adaptive Critic Method // *IEEE Intern. Joint Conf. on Neural Networks (IEEE World Congress on Computational Intelligence).* Hong Kong, 2008.
5. *Barman R., Baishya Ch.J., Kharmalki B., Syiemlieh A., Pegu K.B., Das T., Saha G.* Automated Train Scheduling System Using Genetic Algorithm // *Intern. Symp. on Advanced Computing and Communication (ISACC).* Silchar, 2015.
6. *Kattan A.R.M., Abdullah R., Salam R.A.* Training Feed-Forward Neural Networks Using a Parallel Genetic Algorithm with the Best Must Survive Strategy // *Intern. Conf. on Intelligent Systems, Modelling and Simulation.* Liverpool, 2010.
7. *Haupt R.L., Haupt S.E.* *Practical Genetic Algorithms.* 2nd ed., New Jersey: John Wiley and Sons, 2004. P. 253.
8. *Рутковская Д., Пилиньский М., Рутковский Л.* Нейронные сети, генетические алгоритмы и нечеткие системы: учеб. пособие 2-е изд. М.: Горячая линия-Телеком, 2013.