

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ

УДК 004.8

ПОСТРОЕНИЕ ДВИЖЕНИЯ В ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЕ С АВТОМАТИЗИРОВАННЫМ ДИСПЕТЧЕРОМ

© 2022 г. Р. А. Горбачев^{a,*}, Е. М. Захарова^{a,**},
И. С. Макаров^{b,***}, В. И. Цурков^{b,****}

^a Федеральное государственное автономное образовательное учреждение высшего образования
“Московский физико-технический институт (национальный исследовательский университет)” МФТИ,
Долгопрудный, МО, Россия

^b 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 16.08.2021 г.

После доработки 27.08.2021 г.

Принята к публикации 27.09.2021 г.

В предыдущей работе авторов [1] представлена интеллектуальная система для автоматизации диспетчерской деятельности при организации движения железнодорожного транспорта. Речь идет о корректировке расписаний в многоагентных системах при неожиданных изменениях ситуаций. В данной статье конкретизируется основной подход и детально описаны модель, метод решения и возможная модернизация системы.

DOI: 10.31857/S000233882201005X

Введение. В [2, 3] рассмотрены первые исследования по разработке систем искусственного интеллекта в задаче автоматизированного диспетчера. Для принятия решения используется метод обучения многослойных нейронных сетей. В [1] учитываются недостатки в вышеупомянутых публикациях, формируется более широкий подход в данном направлении многоагентных систем. Ниже принципы и подходы [1] конкретизируются и распространяются на более общий случай. Работа диспетчера – человека, контролирующего движение транспорта, и без того достаточно трудоемка. В связи с повышением интенсивности движения его деятельность становится сложнее, поэтому исследование систем помощи диспетчерскому аппарату на сегодняшний день до сих пор является одним из актуальных направлений [4]. При этом вследствие высоких требований к безопасности движения окончательное решение остается за человеком, а предлагаемые системой решения носят рекомендательный характер.

В работе приведено описание алгоритма построения планового расписания в соответствии с регламентом движения и алгоритма прогнозирования движения с учетом графиков исполненного и планового движения.

1. Постановка задачи и основная модель. Для построения расписания необходимо в первую очередь реализовать модель движения агентов. В статье для этой цели используется дополненная модель, описанная в [1, 5].

Для тестирования интеллектуальной системы был выбран линейный участок на графе железнодорожной сети Монголии, включающий 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 каждого сегмента известны его название, длина и смежные с ним инфраструктурные сегменты. Каждый инфраструктурный сегмент может находиться в одном из трех возможных состояний: “normal”, “locked”, “reduced speed”. Состояние сегмента “normal” означает, что он доступен для движения и на нем нет никаких ограничений. Состояние сегмента “locked” означает, что на нем произошел аншлаг. Аншлагом называется любой запрет движения агентов по данному сегменту. Для каждого аншлага

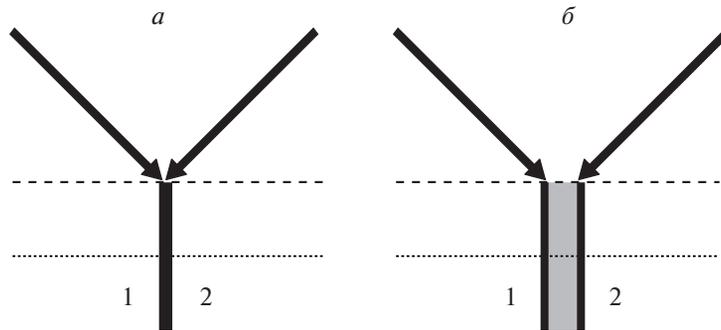


Рис. 1. Схема управления агентами: *a* – некорректная и *б* – корректная

задается номер сегмента и его продолжительность, т.е. время начала и время окончания запрета. Состояние сегмента “reduced speed” означает, что движение по данному сегменту разрешено, однако максимально допустимая скорость агентов ограничена. Максимальное значение скорости в данном случае задается отдельно.

Для каждого сегмента известно время движения каждого агента в обоих направлениях. Стоит отметить, что все сегменты образуют планарный граф, при этом южный полюс любого сегмента соединяется только с северными полюсами других сегментов и наоборот. Таким образом, рассматриваемое движение агентов имеет одно из двух условных направлений – южное или северное (на железной дороге – четное или нечетное). Предыдущие версии алгоритма [1] управляли агентами таким образом, что сегменты занимались без учета интервалов движения между агентами при последовательном движении, т.е. при уходе одного агента с некоторого сегмента, последний мог быть занят другим агентом в тот же момент времени. Однако подобная стратегия управления в реальности не соответствует правилам безопасности движения, поэтому в алгоритм управления был добавлен интервал движения между двумя последовательными агентами, который запрещает агенту проследовать на только что освободившийся сегмент.

На рис. 1 представлены две схемы управления агентами. Вариант *a* соответствует предыдущей версии алгоритма, основанной на использовании нейронной сети, вариант *б* – алгоритму, представленному в данной работе. Точечной пунктирной линией обозначен станционный путь, на котором останавливаются агенты. Штриховой пунктирной линией отмечена граница станции, т.е. условно стрелки, соединяющие станционный путь и перегон. Серым цветом показана занятость, связанная с межпоездным интервалом. Данный интервал означает что время между отбытием одного агента и прибытием другого на один и тот же участок должно быть не менее заданного значения, где стрелки – направление движения агентов.

Каждый агент может находиться в одном из следующих состояний. Состояние “stay” соответствует движению агента по перегону либо остановке на станции, согласно плановому графику движения. Состояние “move” является командой и означает, что агент, который простоял на станционном пути необходимое по плану количество времени, может начать движение к следующей станции по смежному перегону. Агент находится в этом состоянии один момент времени, после чего состояние агента становится “stay”. Состояние “wait” служит командой и означает, что агент, который простоял на станционном пути необходимое по плану количество времени, может продолжать стоянку на этом станционном пути. Данное состояние реализуется только в течение пребывания поезда на станции. Эта команда отдается агенту в том случае, если начинать дальнейшее движение неоптимально в данный момент времени. Состояние “skip” устанавливается для агента, который достиг конечную точку своего маршрута. Такие агенты игнорируются при дальнейшей работе алгоритма.

Важно отметить, что состояния, являющиеся командами, могут быть установлены только в результате возникновения в модели следующих событий: появление нового агента на участке, достижение одним из агентов какой-либо станции своего маршрута, окончание плановой стоянки любого агента.

Для каждого агента известны его тип, направление движения, начальная и конечная точки маршрута и приоритет. Тип движения необходим для определения нормативного времени перемещения агента по каждому сегменту. В каждый отдельный момент времени для каждого агента

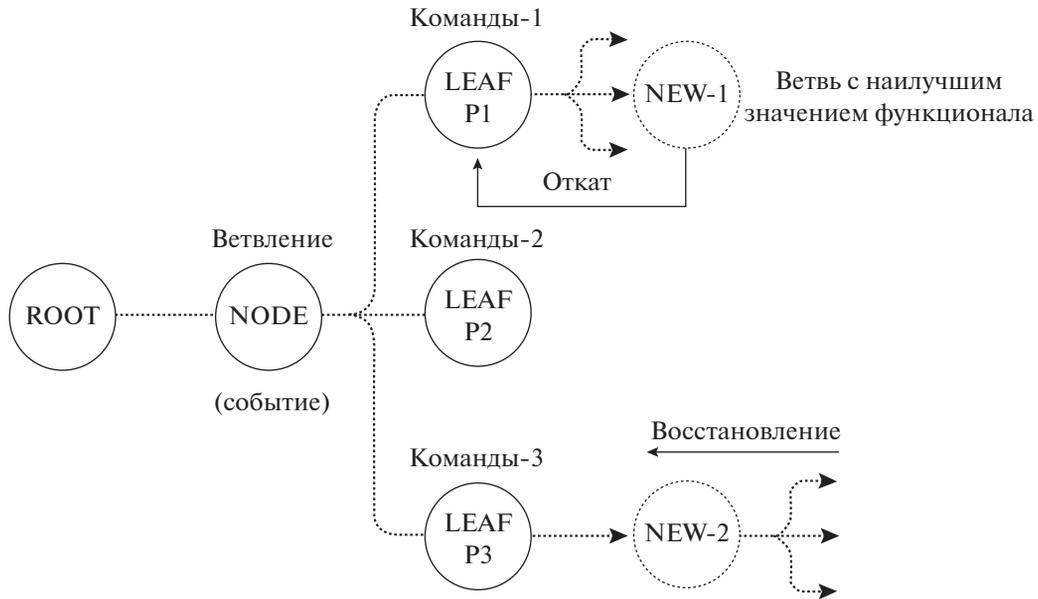


Рис. 2. Построение дерева возможных вариантов развития графика движения: P1, P2, P3 – вероятности выбора листьев

известен текущий сегмент его пребывания, время пребывания на нем и отклонение от планового графика движения. Отклонение пересчитывается для всех агентов на каждом временном шаге.

Отклонение от планового графика движения является целевой функцией, который минимизируется в рассматриваемой оптимизационной задаче и рассчитывается по формуле из [1]:

$$R = \sum_{i=1}^N \sum_{j \in W_i} \omega_i |\Delta T_{ij}| \rightarrow \min_{\theta} \quad (1.1)$$

где i – индекс агента, j – индекс сегмента из множества W_i , обязательных для посещения, согласно маршруту агента i , ω_i – весовой коэффициент агента i на основе его типа – задается вручную для каждого агента, T_{ij} – изменение времени прибытия агента i на станцию j по сравнению с нормативным графиком, θ – тензор весовых коэффициентов нейронной сети.

2. Построение решения. Для построения оптимального графика движения входными данными являются: граф участка дороги, состоящий из инфраструктурных сегментов; список аншлагов (запретов движения); исходный плановый график движения агентов. Для построения прогнозного графика движения в качестве дополнительных данных подается график исполненного до текущего времени движения.

Для маршрутов введем термин нитки. В рамках данной терминологии будут рассматриваться исходные, плановые, исполненные и прогнозные нитки. Каждая нитка задается следующим набором параметров: время начала нитки, приоритет и тип агента движения, массив составляющих ее фрагментов. Фрагментом обозначается совокупность инфраструктурного сегмента, через который проходит нитка, и времени пребывания агента в состоянии “stay” на этом сегменте.

Подход заключается в моделировании движения агентов в рамках многоагентной системы. Моделирование осуществляется дискретно и циклически, с шагом в одну виртуальную единицу времени, которая соответствует одной минуте движения поездов в реальном времени.

Процесс моделирования заключается в построении дерева возможных вариантов развития графика движения, в котором выбираются оптимальные в соответствии с формулой (1.1) стратегии управления.

На рис. 2 представлен алгоритм построения дерева возможных вариантов развития графика движения.

Узлом дерева является вариация состояния всех сегментов и агентов движения в данный момент времени, т.е. массивы агентов и сегментов в некоторых состояниях. В каждом узле содер-

жится информация о смежных узлах дерева – родителе и потомках, текущем отклонении от исходного планового графика, а также флаг события, показывающий, произошло ли какое-либо событие в момент времени, соответствующий данному узлу. Листьями дерева выступают последние рассчитанные варианты. Корень дерева – состояние сегментов и агентов в начальный момент времени. Он устанавливается при инициализации системы. Узлы следуют друг за другом, количество потомков для одного узла может быть больше одного. Несколько потомков создаются за счет вариативности предпринимаемых мер управления, т.е. наборов команд, отдаваемых агентам при возникновении событий. Работа алгоритма заключается в построении данного дерева оптимальным образом и поиску ветви дерева, соответствующей наилучшей стратегии управления движением. При построении дерева используются вариация алгоритма метода Монте-Карло поиска по дереву [6], алгоритм Дейкстры [7], а также отдельные алгоритмы теории графов, такие, как BFS [8], DFS [7]. Результатом работы алгоритма является набор ниток по самой оптимальной ветви дерева стратегий, которые в свою очередь и представляют оптимальное описание движения, передаваемое пользователю.

Алгоритм построения нового узла дерева стратегий заключается в следующем.

Шаг 1. Новые узлы строятся как продолжение одного из имеющихся листов дерева. Изначально лист только один – корень дерева. В структуру узла в первую очередь вносятся агенты, которые уже сформированы и готовы начать свое движение. Для этих агентов устанавливается состояние “stay”. В том случае, если стартовый сегмент маршрута агента занят, его появление откладывается до освобождения данного сегмента.

Шаг 2. На втором шаге осуществляется обновление состояния инфраструктурных сегментов участка. Меняется их статус или, например, появляются новые аншлаги.

Шаг 3. В каждом узле есть массив активных на данный момент времени агентов. Выполняется двойная сортировка данного массива в соответствии с их приоритетами и величиной отклонения по времени от планового графика движения (формула (1.1)).

Шаг 4. Выполнение команд управления движением агентов, унаследованных от предыдущего узла. Так, например, исполнение команды “move” для некоторого агента приводит к его перемещению со стационарного пути на смежный перегон и установке состояния “stay”. В процессе выполнения команд осуществляется проверка на то, что исполнение команды не приведет к возникновению deadlock (deadlock или взаимоблокировка – ситуация, при которой несколько агентов находятся в состоянии ожидания сегментов, занятых друг другом, и ни один из них не может продолжать свое движение). Например, не допускается загруженность агентами локальной области участка графа так, чтобы не оставалось ни одного свободного стационарного пути.

Шаг 5. В соответствии с формулой (1.1) обновляется текущее суммарное временное отклонение от планового графика движения всех агентов.

Шаг 6. Создаются варианты наборов допустимых команд для агентов в данном узле, которые затем передаются в узлы-потомки для исполнения их на следующей итерации цикла в шаге 4. Углубление веток происходит с вероятностью, линейно пропорциональной оценке отклонения от планового графика движения. Данный подход является разновидностью метода Монте-Карло поиска по дереву.

Шаги 1–6 выполняются в цикле до тех пор, пока все агенты не дойдут до конечных точек своих маршрутов либо не будет определено, что движение невозможно из-за ситуации deadlock. При построении дерева стратегий выполняется сжатие и обрезка пройденных веток, не меняющихся на протяжении заданного количества времени. Это позволяет избежать линейного роста потребления памяти при построении дерева.

Особенностью данного алгоритма является стратегия откатов. При построении дерева предпочтение отдается наиболее перспективным ветвям в соответствии с формулой (1.1). В случае ухудшения значений оптимизируемой целевой функции выполняется откат к предыдущим, родительским, узлам, для которых имеется лучшее значение целевой функции. Далее происходит построение другой ветви дерева на основе нового набора команд. Откаты могут выполняться последовательно. Наиболее оптимальной стратегией является откат на один уровень назад. Данный подход также используется для решения deadlock.

На рис. 3 представлен пример deadlock на случайно сгенерированных нитках исполненного движения.

По вертикальной оси отмечены станции выбранного участка железнодорожной сети Монголии и их стационарные пути [9, 10]. На горизонтальной оси задана абсолютная временная шкала.

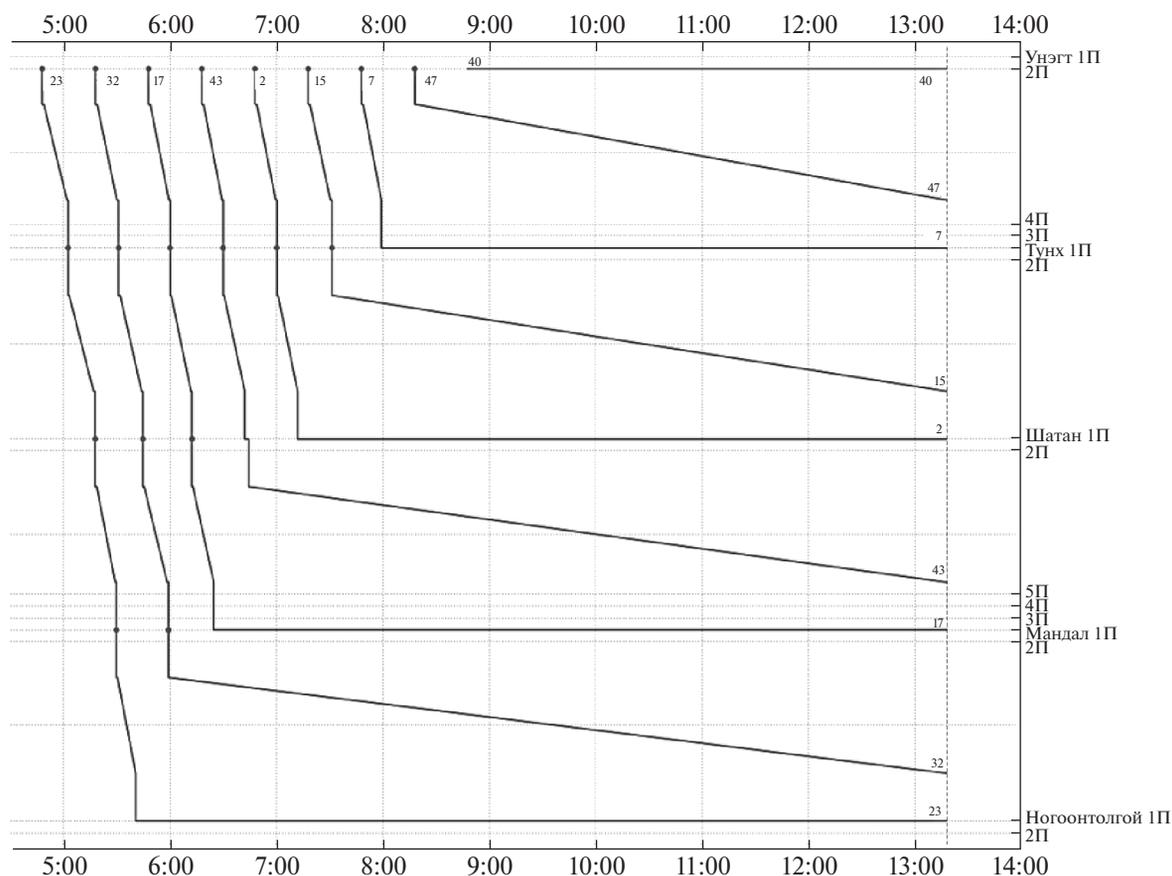


Рис. 3. Пример отображения ситуации deadlock на графике движения

Агенты стоят на сегментах бесконечно долго, поэтому отображаемые нитки тянутся вправо, не достигая конечной точки.

3. Применение. Алгоритм применим как для корректировки планового расписания в соответствии с задаваемыми пользователем ограничениями, так и для построения прогнозного графика движения на основе имеющегося планового и исполненного движения. Во втором случае используется тот же самый алгоритм, однако построение дерева стратегий управления агентами начинается не с нулевого момента времени, а с момента окончания исполненного движения. При этом дерево на протяжении всего исполненного движения состоит из одной единственной ветви. Скорость работы алгоритма на графике любого объема не превышает 1 с.

На рис. 4 приведен результат работы алгоритма по корректировке планового графика движения по реальному участку железной дороги Монголии на основе пожеланий пользователя. По вертикальной оси отмечены станции выбранного участка железнодорожной сети Монголии и их станционные пути. На горизонтальной оси задана абсолютная временная шкала. Исходные плановые нитки обозначены сплошными линиями, а исправленные нитки без конфликтов, предлагаемые алгоритмом, — пунктирными линиями. Стоит обратить внимание на станцию Ногоонтолгой. На ней два станционных пути, однако ни в один момент времени оба не являются занятыми. Это обусловлено алгоритмом предотвращения deadlock.

На рис. 5 приведено предлагаемое алгоритмом прогнозируемое расписание на основе имеющихся графиков планового и исполненного движения. Вертикальная красная черта обозначает текущий момент времени. Слева от черты показано исполненное движение. В данном случае исполненное движение генерировалось случайно с небольшими отклонениями от заданного планового расписания по принципу “первый — главный”. Справа от вертикальной черты также приведено плановое расписание, нитки которого обозначены сплошными линиями. Также представлены нитки прогнозируемого допустимого движения без конфликтов, которые могут быть реализованы в реальности. Прогнозные нитки являются продолжением исполненных,

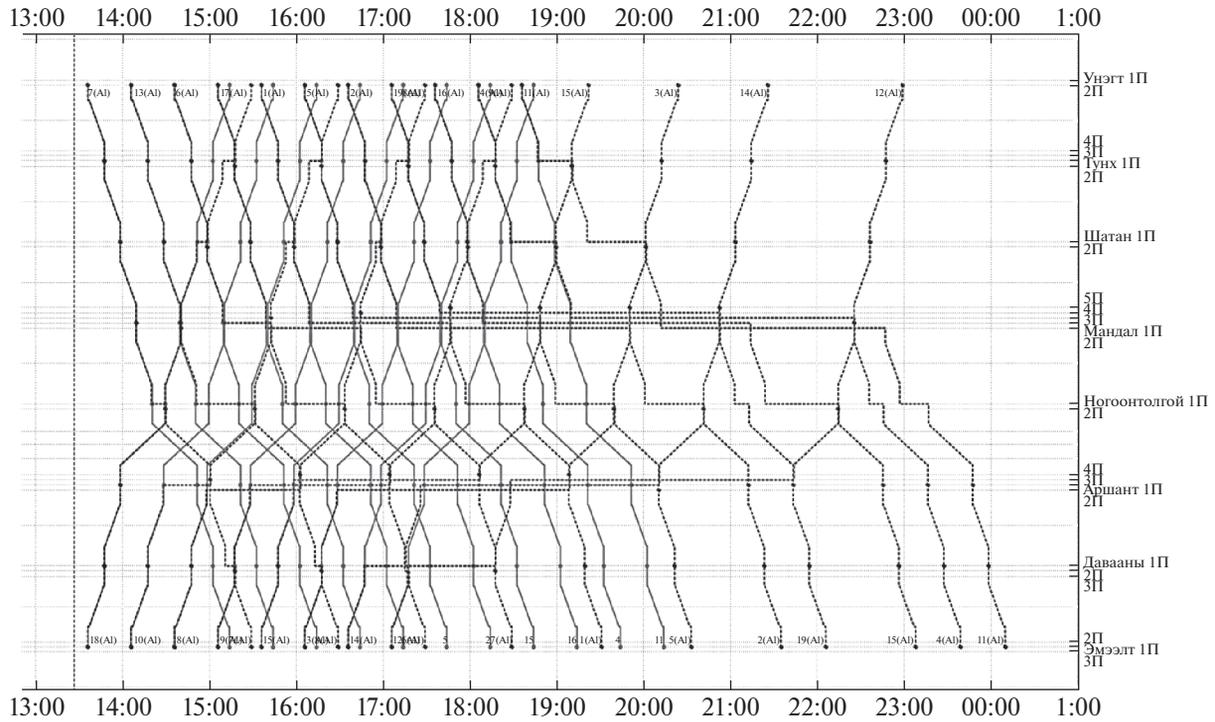


Рис. 4. Результат работы алгоритма при построении оптимального графика движения

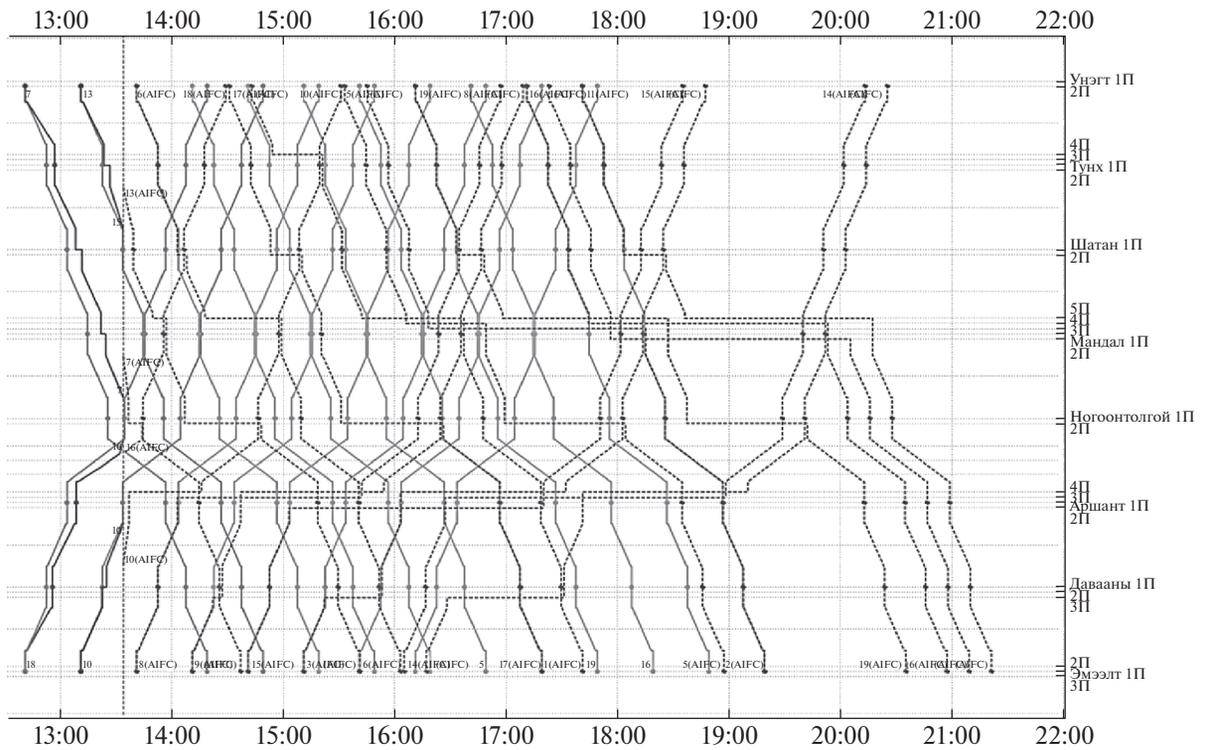


Рис. 5. Пример построенного прогнозного графика движения

при этом они строятся при минимизации отклонения от планового графика движения в соответствии с заданной целевой функцией (формула (1.1)).

Заключение. Предлагаемый подход включает в себя модель поведения диспетчера в случае возникновения конфликтных ситуаций и модель движения. Основная концепция реализуемого решения заключается в приближении задачи управления движением агентов к одной из задач теории игр, в которой участвует один игрок, взаимодействующий с группой агентов в динамической среде. Игрок на каждом шаге игры отдает команды каждому агенту таким образом, чтобы агенты следовали по соответствующим им маршрутам и не нарушали заданные технологические ограничения, связанные с движением по этому маршруту.

Дальнейшее развитие предложенного метода заключается в следующем: отладка существующего решения в соответствии с замечаниями, выявленными в процессах блочного и интеграционного тестирования; реализация возможности пакетного пропуска агентов (возможность пропускать несколько агентов на перегон в одном направлении); оптимизация основного алгоритма игровых стратегий с использованием алгоритмов на графах (BFS, DFS, Dijkstra) для поиска допустимых маршрутов движения и ускорения построения дерева стратегий, а также отсеивание лишних ветвей на основе анализа плотности потока агентов через участок; внедрение оптимизации для прогнозирования циклического движения (пригородное движение, Московское центральное кольцо, Московские центральные диаметры, метро), а именно замена текущей целевой функции, показывающей отклонение от нормативного графика движения, на целевую функцию, работающую с интервальным движением.

СПИСОК ЛИТЕРАТУРЫ

1. Горбачев Р.А., Захарова Е.М., Макаров И.С., Цурков В.И. Интеллектуальная система для пошагового управления при оперативном перестроении расписания // Изв. РАН. ТиСУ. 2021. № 5. С. 111–116.
2. 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. P. 3469–3474.
3. 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. P. 406–412.
4. 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.
5. Zakharova E.M., Gorbachev R.A., Novikov A.N., Makarov I.S. Research and Development of an Intelligent System For Rapid Train Schedule Adjustment Based on Step-by-step Neural Control // 2020 Intern. Conf. Engineering and Telecommunication (En&T). Dolgoprudny, 2020. P. 1–5.
6. Browne C., Powley E., Whitehouse D., Lucas S., Cowling P.I., Rohlfshagen Ph., Tavener S., Perez D., Samothrakis S., Colton S. A Survey of Monte Carlo Tree Search Methods // IEEE Transactions on Computational Intelligence and AI in Games. 2012. V. 4. № 1. P. 1–49.
7. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы. Построение и анализ. М.: Вильямс, 2019. С. 1328.
8. Левитин А.В. Алгоритмы. Введение в разработку и анализ. М.: Вильямс, 2006. С. 576.
9. Балжир М. Перспективное развитие участков Монгольской железной дороги // Транспортное дело России. 2014. № 6. С. 29–33.
10. Batnasan N. Mongolia's Mining-Based Development and Trade Policy // ERINA Report, Japan. 2013. № 109. P. 43–49.