УДК 004.8+65.011.56
doi:10.31799/1684-8853-2022-6-10-19
EDN: LKVJNA

Научные статьи
Articles

# Алгоритмы непрерывного управления для маршрутизации конвейера на основе мультиагентного глубокого обучения с подкреплением

***Я. С. Журба***[а], *студент, orcid.org/0000-0003-3281-9216*
***А. А. Фильченков***[а], *канд. физ.-мат. наук, доцент, orcid.org/0000-0002-1133-8432, aaafil@mail.ru*
***А. А. Азаров***[а,б], *канд. техн. наук, научный сотрудник, orcid.org/0000-0003-3240-597X*
***А. А. Шалыто***[а], *доктор техн. наук, профессор, orcid.org/0000-0002-2723-2077*
[а]*Университет ИТМО, Кронверкский пр., 49, Санкт-Петербург, 197101, РФ*
[б]*Северо-Западный институт управления — филиал РАНХиГС, Средний пр. В. О., 57/43, Санкт-Петербург, 199178, РФ*

***Введение:*** *при перемещении грузов в конвейерной системе необходимо минимизировать не только время транспортировки, но и энергию, затрачиваемую на это перемещение. Энергия перемещения является не декомпозируемой по ребрам функцией, что не позволяет применить к означенной задаче классические алгоритмы маршрутизации.* ***Цель:*** *разработать алгоритм маршрутизации, адаптивный к изменениям в топологии графа маршрутизации и способный оптимизировать время доставки и затрачиваемую электроэнергию.* ***Результаты:*** *предложен алгоритм на основе мультиагентного глубокого обучения с подкреплением, помещающий агентов в вершины графа конвейерной сети, использующий новую функцию ценности состояний. Алгоритм имеет два настраиваемых параметра: длину пути, по которой считается функция ценности состояния, и коэффициент обучения. Благодаря подбору параметров выявлено, что оптимальными значениями являются 2 и 1 соответственно. На основе экспериментального исследования алгоритма с использованием симуляционной модели установлено, что его применение позволяет свести к нулю число столкновений перемещаемых объектов, обеспечивая достижение устойчивых результатов работы по обоим оптимизируемым функционалам, а также приводит к меньшему потреблению электроэнергии в сравнении с референтным алгоритмом.* ***Практическая значимость:*** *предложенный алгоритм может быть использован для уменьшения времени и энергии доставки при управлении конвейерными системами.*

***Ключевые слова*** *— маршрутизация, мультиагентное обучение, обучение с подкреплением, конвейерная лента.*

**Для цитирования:** Журба Я. С., Фильченков А. А., Азаров А. А., Шалыто А. А. Алгоритмы непрерывного управления для маршрутизации конвейера на основе мультиагентного глубокого обучения с подкреплением. *Информационно-управляющие системы*, 2022, № 6, с. 10–19. doi:10.31799/1684-8853-2022-6-10-19, EDN: LKVJNA
**For citation:** Zhurba Y. S., Filchenkov A. A., Azarov A. A., Shalyto A. A. Continuous control algorithms for conveyer belt routing based on multi-agent deep reinforcement learning. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2022, no. 6, pp. 10–19 (In Russian). doi:10.31799/1684-8853-2022-6-10-19, EDN: LKVJNA

## Введение

Одним из пластов развития современной постиндустриальной цивилизации является удешевление доставки грузов за счет построения сложных логистических цепочек между центрами производства и до конечного потребителя. Минимизация затрат на логистику и управление логистикой позволяет формировать новые бизнесы в парадигме логистика-как-сервис, к которым относятся компании по доставке еды и продуктов и компании, связывающие таксистов и потребителей [1].

Одновременно с этим растет и число математических постановок такого рода задач, пытающихся вобрать в себя все больше факторов, чтобы обеспечить применимость найденного аналитическим или вычислительным способом решения к исходной проблеме реального мира. Это порождает богатое и неоднородное семейство математических постановок задач маршрутизации, тяготеющее к росту числа учитываемых ограничений [2].

Одно из принципиальных и наиболее универсальных таких ограничений — требование адаптируемости решения к изменению исходных условий задачи, например топологии графа [2–4]. Это требование продиктовано регулярными непредсказуемыми воздействиями внешней для проблемы среды. Например, на дорогах возникают пробки, каналы связи обрываются, пути подвоза подвергаются естественным разрушениям.

Настоящая работа посвящена задаче маршрутизации штучных грузов в конвейерных системах. Она возникает в промышленном производстве [6, 7], в аэропортах в системах распределения багажа [8], в теплицах [9]. Такие конвейерные системы характеризуются высокой значимостью оптимизации потребляемой электроэнергии [5].

Целью данной статьи является разработка алгоритма маршрутизации, адаптивного к изменениям в топологии графа и способного оптимизировать время доставки и затрачиваемую электроэнергию. В качестве основы для разработки алгоритма был выбран подход, использующий мультиагентное глубокое обучение с подкреплением, поскольку мультиагентные подходы известны хорошей адаптивностью под условия среды [10], а глубокое обучение с подкреплением позволяет обучать сложные модели управления [11].

## Формальная постановка задачи

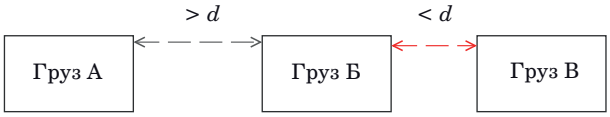
Задача маршрутизации ставится для некоторого графа $G = \langle E, V \rangle$, где $E$ — множество ребер, а $V$ — множество вершин. Будем полагать, что граф ориентированный. Назовем грузом объект, который будет транспортироваться по данному графу. Необходимо оценивать затраты на доставку груза из вершины $b$ в вершину $d$. Для этого будем считать, что известна стоимость $c(i, j)$ каждого ребра $(v_i, v_j) \in E$. В данном графе при заданных $b$ и $d$ необходимо проложить маршрут наименьшей стоимости из $b$ в $d$:

$$\sum_{\substack{i=1 \\ v_1=b \\ (v_i,v_{i+1})\in E \\ v_n=d}}^{n} c(i, i+1) \to \min. \qquad (1)$$

Рассмотрим более частный случай, а именно постановку задачи маршрутизации для конвейерных систем. Модель должна учитывать: 1) возможность столкновения транспортируемых грузов, что является крайне нежелательным событием при транспортировке; 2) затрачиваемую на перемещение энергию. Эти ограничения формируют множество функций оценки качества решения.

Отсутствие столкновений грузов является критически важным условием для работы алгоритма маршрутизации конвейерных систем. Формализуем это условие. Введем параметр $d > 0$, который будет отвечать за минимальное допустимое расстояние между грузами; если в какой-то момент времени расстояние будет меньше $d$, то система будет фиксировать факт столкновения (рис. 1).

Затрачиваемая на перемещение энергия хотя и положительно скоррелирована со временем доставки, но не эквивалентна ему. В конвейерных лентах энергия тратится на перемещение ленты по участку пути, а не на перемещение конкретного груза, поэтому совокупная энергия, потраченная на перемещение нескольких грузов по одному длинному участку пути, может оказаться меньше, чем энергия, потраченная на перемещение отдельно каждого груза на нескольких коротких участках пути.




Столкновение грузов Б и В

■ ***Рис. 1.*** Пример столкновения грузов на конвейере
■ ***Fig. 1.*** An example of collision of cargo pieces on a conveyor

Для описания затрачиваемой энергии была использована формула [12]

$$P(V) = \frac{M_c(m_b) \cdot V}{\rho}, \qquad (2)$$

где $V$ — мгновенная скорость ленты конвейера; $M_c$ — величина полного сопротивления движению, которая зависит от массы всех объектов $m_b$, перемещающихся на ленте; $\rho$ — КПД конвейерной системы.

С точки зрения постановки задачи оптимизации отличие функционала энергии от функционала времени доставки заключается в более сложной математической природе первого, являющегося *не декомпозируемым по ребрам и объектам*, что затрудняет поиск решения классическими алгоритмами. Функция называется декомпозируемой, если ее можно разложить на отдельные части. Например, время доставки груза можно представить как сумму значений времени по ребрам. В случае энергии данное разложение невозможно, поскольку значение затрачиваемой энергии является динамической величиной, описываемой формулой (2).

## Маршрутизация на основе обучения с подкреплением

Классические алгоритмы маршрутизации, такие как, например, алгоритм Дейкстры [13], не позволяют решать задачу поиска оптимального по не декомпозируемому по ребрам функционалу пути. Применимым для такого рода задач подходом является обучение с подкреплением, позволяющее работать с более широким семейством функционалов, чем классические алгоритмы [14].

Обучение с подкреплением является областью машинного обучения и отличается в постановке задачи наличием взаимодействия агента со средой, которое заключается в итеративном совершении действий агентом и получении в от-

вет награды. В процессе обучения агент ищет оптимальную стратегию для наиболее эффективного с точки зрения введенного функционала взаимодействия со средой. Развитие нейронных сетей сформировало подраздел глубокого обучения с подкреплением. Данный подход использует непрерывную обратную связь, чтобы подстраиваться под текущее состояние среды. Это позволяет создавать алгоритмы с высоким уровнем адаптивности под изменчивые условия, что является крайне важным свойством при разработке алгоритмов маршрутизации для конвейерной сети.

**Q-routing**

Алгоритм Q-routing использует в основе метод обучения с подкреплением, но без нейронных сетей [15]. Удобно считать, что в каждую вершину графа помещен агент, а выбор соседа для передачи груза является списком доступных действий. С каждой вершиной ассоциирована таблица оценок ожидаемой награды при выборе того или иного соседа для пересылки груза $Q_x(d, y)$, где $x$ — текущая вершина, в которой находится груз; $d$ — конечный пункт назначения; $y$ — один из соседей вершины $x$, куда переместится груз. Находясь в вершине $x$ и стремясь направить объект в конечный пункт $d$, при пересылке груза в вершину $y$ агент получит награду $Q_x(y, d)$. Поскольку агент старается максимизировать значение функции $Q$, то его невозможно использовать для минимизации среднего времени доставки. Припишем весам всех ребер отрицательный знак и теперь рассмотрим в таком графе задачу максимизации среднего отрицательного времени доставки, решение которой будет совпадать с решением исходной задачи.

Ниже представлена формула, по которой происходят обновления значений функции $Q$:

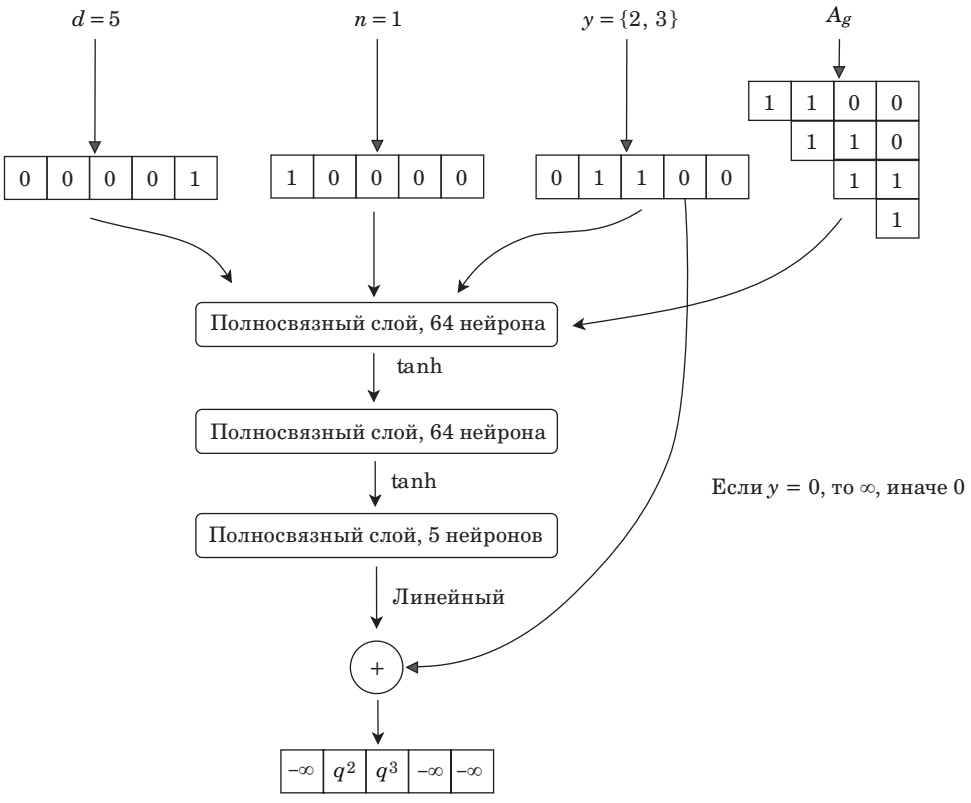
$$Q^*(s) = Q(s) + \alpha(r + \gamma Q(s') - Q(s)). \qquad (3)$$

Данный подход обладает хорошей гибкостью, это связано с постоянным получением актуальной оценки действий агента из среды [16]. Существенным недостатком алгоритма является высокое потребление памяти, поскольку под таблицы оценок необходимо выделить место.

**DQN-routing**

Алгоритм DQN-routing был описан в работе [17]. Базовая идея вдохновлена алгоритмом DQN [18]: обновления значений $Q$-функции аппроксимируются нейронной сетью (рис. 2).

Она представляет сеть прямого распространения с двумя скрытыми слоями, размер каждо-




■ ***Рис. 2.*** Архитектура нейронной сети DQN

■ ***Fig. 2.*** Architecture of the DQN neural network

го слоя составляет 64 нейрона. Слои соединяются функцией активации гиперболического тангенса. Нейронная сеть принимает на вход 1) $d$ — вершину назначения; 2) $x$ — текущую вершину; 3) $y$ — одного из соседей $x$; 4) граф $G$, при этом для представления графа используется матрица смежности, а для представления вершин — прямое кодирование. Вектором прямого кодирования вершины выступает вектор длины, равной числу вершин в графе, в котором на позиции номера вершины стоит единица, на остальных позициях стоят нули. Такой метод кодирования помогает избежать корреляции между номерами вершин, что повышает качество работы сети. На выходе сеть возвращает оценочную стоимость пути при выборе в качестве следующей вершины на пути транспортировки груза соседа $y$.

Данный подход обладает большей адаптивностью по сравнению с алгоритмом Q-routing, однако имеет недостатки. Во-первых, он потребляет много памяти, так как хранит всю матрицу смежности в каждом узле, что делает данный подход применимым только на графах размерности порядка 100– 200 вершин. Во-вторых, требуется предобучение сети.

**DQN-LE**

Развитием алгоритма DQN-routing является алгоритм DQN-LE [19]. Основное улучшение состоит в том, что удалось избавиться от необходимости передавать матрицу смежности, поскольку теперь по номерам вершин строятся векторные представления [20], которые отражают меру близости вершин. Функция активации была заменена на ReLU, что ослабило проблему затухания градиента и, как следствие, увеличило скорость обучения агентов.

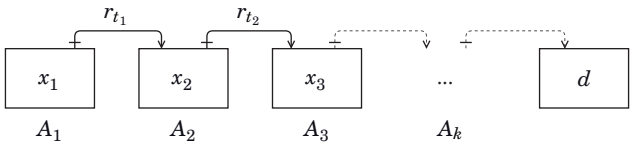
Несмотря на то, что данный подход демонстрирует лучшие результаты работы в конвейерной системе, алгоритм имеет ряд недостатков: 1) сохраняется необходимость предварительного обучения; 2) приближение оценок путей считается недостаточно точно, что приводит к высокой дисперсии при работе алгоритма; 3) незначительное число параметров не позволяет проводить более тонкую настройку работы алгоритма.

## Алгоритм DQN-Path

**Общее описание алгоритма**

Предлагаемый в настоящей работе алгоритм DQN-Path основан на алгоритме DQN-LE.

Средой для агента выступает конвейерная система. В качестве награды используются затрачиваемая энергия и время на перемещение груза, взятое с противоположным знаком. Именно в таком случае максимизация полученной наград будет соответствовать минимизации затраченных времени и энергии.




■ ***Рис. 3.*** Пример перемещения одного груза

■ ***Fig. 3.*** An example of a single cargo piece transportation

В вершине $x_i$ расположен агент, с которым ассоциирована нейронная сеть $A_i$ (архитектура которой показана на рис. 2). Нейронная сеть принимает на вход векторные представления соседней вершины $y$ и направления транспортировки $d$, а на выход возвращает ожидаемое значение награды $Q$. Агент принимает решение, используя стохастический подход, опираясь на softmax-стратегию, при которой вероятность выбрать соседа $y$ для пересылки ему груза определяется следующим образом:

$$p(y) = \frac{A_i(y, d)}{\sum_{y_j:(x_i,y_j)\in E} A_i(y_j, d)}. \quad (4)$$

Основное отличие алгоритма DQN-Path от DQN-LE заключается в обучении агентов. Для объяснения воспользуемся рис. 3, на котором изображен маршрут движения груза. Через $x_i$ обозначены вершины, через $r_{t_i}$ — полученные награды, через $A_i$ — нейронные сети агентов в соответствующих вершинах, $d$ — вершина назначения перемещаемого груза. Обучение агента в вершине $x_1$ происходит следующим образом. После принятого агентом решения груз отправляется в выбранном направлении, и по достижении им следующей вершины можно посчитать реально затраченные время и энергию и, как следствие, вычислить награду $r_{t_i}$, полученную за прохождение выбранного агентом участка конвейерной ленты.

**Уточнение оценок**

В алгоритме DQN-LE обучение агента в вершине $x_1$ происходило за счет обратного распространения ошибки, которая вычислялась как квадрат невязки наблюдаемого значения ценности $Q$-функции и ее предсказания:

$$Loss(A_1) = \left\| Q_1'(x_2, d) - A_1(x_2, d) \right\|^2, \quad (5)$$

где значение $Q_1'(x_2, d)$ определялось по формуле

$$Q_1'(x_2, d) = r_{t_1} + A_2(x_3, d). \quad (6)$$

Следует обратить внимание, что оценка ожидаемой награды, используемая в уравнении (6), является в значительной степени неточной, что может приводить к большой дисперсии результатов, как показывают описанные в следующем разделе эксперименты. Уточнение оценки приводит к уменьшению ее смещенности. Для достижения этого обратимся к стандартной формуле ценности состояния для обучения с подкреплением без использования нейронных сетей с одним агентом, записанной в классическом виде [16], в которой пересчитывается оценка целевой функции после совершенного действия в состоянии $s$ и при переходе в состояние $s'$:

$$Q(s) = Q(s) + \alpha\left(r_{t_1} + \gamma Q(s') - Q(s)\right). \quad (7)$$

Перепишем ее в следующем виде:

$$Q(s) = Q(s)(1-\alpha) + \alpha\left(r_{t_1} + \gamma Q(s')\right). \quad (8)$$

Далее совершим замену:

$$Q(s) = Q(s)(1-\alpha) + \gamma Q'(s), \quad (9)$$

где

$$Q'(s) = r_{t_1} + \gamma Q(s'). \quad (10)$$

Уравнения (6) и (10) похожи, но имеют следующие различия:

1) в уравнении (6) имеются разные индексы у $Q$, это связано с тем, что разным индексам соответствуют разные агенты;

2) в уравнении (10) происходит обучение методом экспоненциального скользящего среднего, в отличие от метода обратного распространения ошибок, использующего ошибку из уравнения (5);

3) в уравнении (10) используется *коэффициент обучения* $\gamma$. Добавив его к уравнению (6), получим

$$Q_1'(x_2, d) = r_{t_1} + \gamma A_2(x_3, d). \quad (11)$$

Следующая идея для улучшения этой оценки состоит в том, чтобы заменить значение $A_2$ на $Q_2'$, т. е. вместо предсказания оценки следующего агента воспользоваться полученной ценностью состояния. Распишем, в свою очередь, оценку $Q_2'$:

$$Q_2'(x_3, d) = r_{t_2} + \gamma A_3(x_4, d). \quad (12)$$

Подставив это значение в уравнение (11), получим

$$Q_1'(x_2, d) = r_{t_1} + \gamma\left(r_{t_2} + \gamma A_3(x_4, d)\right), \quad (13)$$

$$Q_1'(x_2, d) = r_{t_1} + \gamma r_{t_2} + \gamma^2 A_3(x_4, d). \quad (14)$$

Продолжив замену на значения функций в следующих вершинах, мы придем к обобщенной формуле

$$Q_1'(x_2, d) = r_{t_1} + \sum_{i=1}^{k} \gamma^{i-1} r_{t_1} \gamma r_{t_2} + \gamma^k A_k(x_k, d). \quad (15)$$

Подставив ее в уравнение (5), получим значение ошибки для обучения нейронной сети.

Использование уравнения (15) при обучении позволяет уточнять оценку ценности пути за счет использования обучения не по одному шагу, а по частичному маршруту. Получилась относительно несложная формула, что является несомненным плюсом для реализации алгоритма. Кроме того, алгоритм стал параметрическим, поскольку теперь его работа определяется коэффициентом обучения $\gamma$ и длиной частичного маршрута.

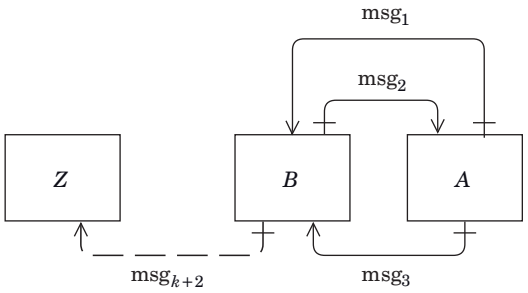
Стоит отметить, что обучение является *асинхронным*, поскольку происходит именно по частичному маршруту груза, а не по подпоследовательности шагов агента. Это хорошо видно из индексов награды, получение которой происходит не в последовательные моменты времени. Строго говоря, обучение происходит на основе одного шага каждого из агентов, соответствующих вершинам отрезка маршрута передачи груза.

**Протоколы общения агентов**

Из уравнения (15) видно, что для работы алгоритма необходимо хранить историю посещения последних $k$ агентов грузом, так как значения наград этого пути используются в формуле для обучения.

Основная идея программной реализации заключается в создании у каждого агента локального хранилища, в котором будут храниться истории маршрутов только что отправленных грузов. В таком случае сохраняется децентрализованность всей системы, при этом получение информации об истории будет осуществляться за счет реализации протокола общения агентов между собой. Опишем работу протокола в общем виде (рис. 4).

Пусть агент $A$ получил управление от среды с запросом принять решение о пересылке только что прибывшего груза. Предполагается, что груз приехал от агента $B$, соответственно, в его локальном хранилище есть информация о маршруте. Поэтому от $A$ к $B$ отправляется сообщение с запросом о прибывшем грузе. Агент $B$, получив это сообщение, удаляет из своего хранилища информацию о грузе и направляет ее агенту $A$. Агент $A$ обновляет информацию, используя полученные данные, и сохраняет в локальную копию.




■ ***Рис. 4.*** Иллюстрация работы протокола: msg — пересылаемые между агентами сообщения

■ ***Fig. 4.*** An illustration of the protocol: msg stands for messages sent between agents

Если накоплен достаточный маршрут для обучения, то история передается по обратному пути, пока не пройдет $k$ вершин, после этого происходит обучение последнего в этом эпизоде агента $Z$, получившего всю необходимую историю.

## Экспериментальное исследование алгоритма

### Симуляционная модель и сценарии тестирования

Для проведения экспериментов использовалась симуляционная модель конвейерной системы, представленной взвешенным ориентированным графом. Ребрам соответствуют части ленты, при этом ориентация отображает направление движения, а вес является мерой длины. Вершинами являются контрольные точки: стартовые вершины и конечные вершины, а также места стыковки конвейерных лент (все остальные вершины). В каждую вершину помещен агент. Децентрализованность достигается путем ограничения возможности конвейерной системы обмениваться сообщениями между узлами, т. е. каждый узел может посылать сообщения только соседним узлам.

Симуляционная модель позволяет рассчитывать время доставки, энергию доставки согласно уравнению (2), а также фиксировать столкновения грузов.

Для тестирования использовались два типа сценария. Первый тип сценария, *без поломок*, работает с фиксированной топологией графа, при его выполнении случайным образом добавляются грузы со случайно выбранными начальной вершиной из $B$ и конечной вершиной из $D$.

Второй тип сценария, *с поломками*, отображает изменение топологии конвейерной сети. В нем возникают события поломки и восстановления конвейерных лент. Данные события позволят оценить важнейшее свойство реализованного алгоритма, а именно способность алгоритма к адаптации в условиях изменяющейся нагрузки и топологии сети, а также свойства отказоустойчивости. В остальном он повторяет первый тип.

### Критерии сравнения алгоритмов и оценка статистической значимости

Ключевыми метриками сравнения являются среднее время доставки груза, средняя энергия, затраченная на доставку груза, и число столкновений. Наличие столкновений недопустимо.

Для проверки статистической значимости в работе использовался критерий Уилкоксона для связанных выборок с уровнем значимости 0,05, поскольку сравнивались пары результатов, в которых каждая пара была протестирована на одинаковом сценарии работы конвейера.

### Эксперименты по подбору параметров алгоритма DQN-Path

Алгоритм DQN-Path имеет два параметра: длину пути $c$ и коэффициент обучения $\gamma$. В исследовании рассматривались значения параметра $c$, равные 1 и 2. Для каждого варианта длины пути выполнены запуски с различными значениями параметра $\gamma$ из диапазона [0,5; 1,5], среди них оставлены варианты с наилучшими результатами. Такими оказались пары: $c = 1, \gamma = 1{,}01$; $c = 2$, $\gamma = 1$; $c = 2, \gamma = 0{,}7$. Также был добавлен вариант $c = 1$, $\gamma = 1$, что полностью соответствует алгоритму DQN-LE. Для каждого из данных алгоритмов подсчитаны средние метрики по десяти запускам. Результаты сравнений представлены в табл. 1.

Как видно из таблицы, значение параметра $\gamma$ влияет на результат работы алгоритма. Лучшим по обоим критериям является алгоритм со значениями параметров $c = 2$, $\gamma = 1$. Именно данные параметры алгоритма были взяты для тестирования с DQN-LE. Число столкновений во всех алгоритмах равно 0.

■ ***Таблица 1.*** Сравнение значений параметров по средним показателям

■ ***Table 1.*** Comparison of parameter values with respect to mean time and mean energy

| Алгоритм | Среднее время доставки, с | Средняя затраченная энергия, ед. |
|---|---|---|
| DQN-Path, $c = 2$; $\gamma = 1$ | 56,025 | 53 829 |
| DQN-Path, $c = 2$; $\gamma = 0{,}7$ | 56,095 | 54 120 |
| DQN-Path, $c = 1$; $\gamma = 1{,}01$ | 56,112 | 54 761 |
| DQN-Path, $c = 1$; $\gamma = 1$ (DQN-LE) | 56,367 | 55 370 |

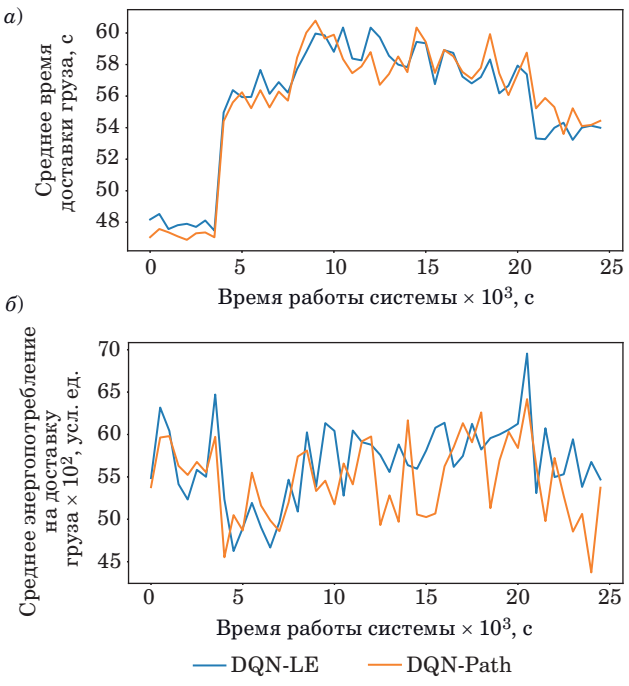
### Эксперименты по сравнению DQN-Path с DQN-LE

Поскольку алгоритм DQN-LE превосходит классические и нейросетевые алгоритмы с точки зрения адаптивности и возможности использования недекомпозируемых метрик, как было показано в [17, 19], сравнение будет проводиться с ним.

Для тестирования алгоритмов было зафиксировано 20 случайных сценариев, на основе которых строились графики и происходило усреднение метрик.

*Анализ сценария с поломками.* Число столкновений у обоих алгоритмов было равно 0 во всех запусках. Из графика среднего времени доставки (рис. 5, *а*) непонятно, какой из алгоритмов справился с оптимизацией данной метрики лучше, анализ значений из табл. 2 также не свидетельствует о сколько-то значимых улучшениях. Однако алгоритм DQN-Path отмечается более несмещенной оценкой, а следовательно, более стабильной и прогнозируемой работой.

Из графика средней энергии доставки (рис. 5, *б*) и табл. 2 заметно улучшение данной метрики. Был проведен статистический тест Уилкоксона, который показал уровень статистической значимости равным 0,00059. Итого имеем уменьшение средней энергии доставки для данного сценария.




■ ***Рис. 5.*** Зависимость среднего времени (*а*) и средней энергии (*б*) доставки груза от времени работы конвейера для сценария с поломками

■ ***Fig. 5.*** Dependence of the average delivery time (*а*) and mean energy consumption (*б*) on conveyor belt system operating time for a breakdown scenario

■ ***Таблица 2.*** Сравнение средних данных для сценария с поломками

■ ***Table 2.*** Comparison of the mean time and mean energy for the collision scenarios

| Алгоритм | Среднее время доставки, с | Среднее отклонение, с | Средняя энергия, ед. | Среднее отклонение, ед. |
|---|---|---|---|---|
| DQN-LE | 56,047 | 1,153 | 51 960 | 625 |
| DQN-Path 2; 1 | 56,029 | 0,309 | 51 756 | 507 |

■ ***Таблица 3.*** Сравнение средних данных для сценария без поломок

■ ***Table 3.*** Comparison of the mean time and the mean energy for the collisionless scenarios

| Алгоритм | Среднее время доставки, с | Среднее отклонение, с | Средняя энергия, ед. | Среднее отклонение, ед. |
|---|---|---|---|---|
| DQN-LE | 55,141 | 0,701 | 51 960 | 625 |
| DQN-Path 2; 1 | 54,316 (нет стат. значимости) | 0,205 | 51 756 | 507 |

*Анализ сценария без поломок.* Как и в прошлом случае, число столкновений у обоих алгоритмов было равно 0 во всех запусках.

Из табл. 3 следует, что не удалось статистически значимо улучшить среднее время доставки, однако удалось сократить среднее отклонение этой величины. При этом статистически значимо уменьшилась потребляемая энергия и сократилось среднее отклонение этой величины. Статистическая значимость по статистическому тесту Уилкоксона равна 0,000006.

Таким образом, можно заключить, что предложенный алгоритм показывает более устойчивую работу с точки зрения обеих рассмотренных метрик, сокращая среднее отклонение, а также уменьшает затраты на использованную энергию.

### Заключение

В результате настоящего исследования был разработан алгоритм DQN-Path, который решает задачу маршрутизации штучных грузов на конвейерной ленте на основе мультиагентного глубокого обучения. Новизна алгоритма заключается в подборе функции ценности состояния для агента, которая вычисляется асинхронно для перемещения грузов и позволяет учитывать функцию ценности состояний вершин пути, которые проходит груз. Проведенные эксперименты показали, что использование предложенной

функции ценности увеличивает устойчивость работы алгоритма, а также уменьшает потребляемую конвейером энергию.

Алгоритм может быть применен для управления конвейерными системами и позволит уменьшать энергозатраты при доставке грузов.

Следует также отметить, что алгоритм может быть обобщен на более широкий класс задач маршрутизации и применяться к произвольным оптимизируемым функциям, таким как амортизация грузов / каналов передачи или стоимость доставки.

**Финансовая поддержка**

Исследование выполнено за счет гранта Российского научного фонда (проект № 20-19-00700).

## Литература

1. **Winkelhaus S., Grosse E. H.** Logistics 4.0: A systematic review towards a new logistics system. *International Journal of Production Research*, 2020, vol. 58, iss. 1, pp. 18–43. doi:10.1080/00207543.2019.1612964
2. **Toth P., Vigo D.** An overview of vehicle routing problems. *The Vehicle Routing Problem* / Eds. Toth P., Vigo D. SIAM, 2002. Pp. 1–26. doi:10.1137/1.9780898718515.ch1
3. **Sweda T. M., Dolinskaya I. S., Klabjan D.** Adaptive routing and recharging policies for electric vehicles. *Transportation Science*, 2017, vol. 51(4), pp. 1326–1348. doi:10.1287/trsc.2016.0724
4. **Puthal M. K., Singh V., Gaur M. S., Laxmi V.** C-Routing: An adaptive hierarchical NoC routing methodology. *2011 IEEE/IFIP 19th Intern. Conf. on VLSI and System-on-chip*, IEEE, 2011, pp. 392–397. doi:10.1109/VLSISoC.2011.6081616
5. **Marasova D., Andrejiova M., Grincova A.** Dynamic model of impact energy absorption by a conveyor belt in interaction with the support system. *Energies*, 2021, vol. 15(1), p. 64. doi:10.3390/en15010064
6. **Köken E., Lawal A. I., Onifade M., Özarslan A.** A comparative study on power calculation methods for conveyor belts in mining industry. *International Journal of Mining, Reclamation and Environment*, 2022, vol. 36, iss. 1, pp. 26–45. doi:10.1080/17480930.2021.1949859
7. **Karami F., Fathi M., Pardalos P. M.** Conveyor operations in distribution centers: Modeling and optimization. *Optimization Letters*, 2022. doi:10.1007/s11590-022-01912-7. https://link.springer.com/article/10.1007/s11590-022-01912-7 (дата обращения: 11.09.2022).
8. **Black G., Vyatkin V.** Intelligent component-based automation of baggage handling systems with IEC 61499. *IEEE Transactions on Automation Science and Engineering*, 2009, vol. 7, iss. 2, pp. 337–351. doi:10.1109/TASE.2008.2007216
9. **Dorri A., Kanhere S. S., Jurdak R.** Multi-agent systems: A survey. *IEEE Access*, 2018, no. 6, pp. 28573–28593. doi:10.1109/ACCESS.2018.2831228
10. **Arulkumaran K., Deisenroth M. P., Brundage M., Bharath A. A.** Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 2017, vol. 34, iss. 6, pp. 26–38. doi:10.1109/MSP.2017.2743240
11. **Ma D., Carpenter N., Amatya S., Maki H., Wang L., Zhang L., Neeno S., Tuinstra M. R., Jin J.** Removal of greenhouse microclimate heterogeneity with conveyor system for indoor phenotyping. *Computers and Electronics in Agriculture*, 2019, no. 166. doi:10.1016/j.compag.2019.104979. https://www.sciencedirect.com/science/article/abs/pii/S016816991930691X (дата обращения: 11.09.2022).
12. **Halepoto I. A., Shaikh M. Z., Chowdhry B. S., Uqaili M. A**. Design and implementation of intelligent energy efficient conveyor system model based on variable speed drive control and physical modeling. *International Journal of Control and Automation*, 2016, vol. 9, iss. 6, pp. 379–388. doi:10.14257/ijca.2016.9.6.36
13. **Noto M., Sato H.** A method for the shortest path search by extended Dijkstra algorithm. *2000 IEEE Intern. Conf. on Systems, Man and Cybernetics "Cyberbetics Evolving to Systems, Humans, Organizations, and their Complex Interactions" (SMC 2000 Conf. Proc.)*, IEEE, 2000, vol. 3, pp. 2316–2320. doi:10.1109/ICSMC.2000.886462
14. **Mammeri Z.** Reinforcement learning based routing in networks: Review and classification of approaches. *IEEE Access*, 2019, no. 7, pp. 55916–55950. doi:10.1109/ACCESS.2019.2913776
15. **Boyan J., Littman M.** Packet routing in dynamically changing networks: A reinforcement learning approach. *NIPS'93: Proc. of the 6th Intern. Conf. on Neural Information Processing Systems*, 1993, pp. 671–678.
16. **Sutton R. S., Barto A. G.** *Reinforcement Learning: An Introduction*. 2nd ed. MIT Press, 2018. 552 p.
17. **Mukhutdinov D., Filchenkov A., Shalyto A., Vyatkin V.** Multi-agent deep learning for simultaneous optimization for time and energy in distributed routing system. *Future Generation Computer Systems*, 2019, no. 94, pp. 587–600. doi:10.1016/j.future.2018.12.037
18. **Mnih V., Kavukcuoglu K., Silver D., Graves A., Antonoglou I., Wierstra D., Riedmiller M.** *Playing Atari with Deep Reinforcement Learning*. arXiv preprint, 2013. https://arxiv.org/abs/1312.5602 (дата обращения: 11.09.2022).
19. **Мухудинов Д.** *Децентрализованный алгоритм управления конвейерной системой с использованием методов мультиагентного обучения с подкреплением*: магистерская дис. СПб., Университет

ИТМО, 2020. 92 с. http://is.ifmo.ru/diploma-theses/2019/2_5458464771026191430.pdf._ (дата обращения: 11.09.2022).
20. **Belkin M., Niyogi P.** Laplacian eigenmaps and spectral techniques for embedding and clustering. *NIPS'01: Proc. of the 14th Intern. Conf. on Neural Information Processing Systems: Natural and Synthetic*, 2001, pp. 585–591. doi:10.5555/2980539.2980616

UDC 004.8+65.011.56
doi:10.31799/1684-8853-2022-6-10-19
EDN: LKVJNA

**Continuous control algorithms for conveyer belt routing based on multi-agent deep reinforcement learning**

Y. S. Zhurba[a], Student, orcid.org/0000-0003-3281-9216
A. A. Filchenkov[a], PhD, Phys.-Math., Associate Professor, orcid.org/0000-0002-1133-8432, aaafil@mail.ru
A. A. Azarov[a,b], PhD, Tech., Research Fellow, orcid.org/0000-0003-3240-597X
A. A. Shalyto[a], Dr. Sc., Tech., Professor, orcid.org/0000-0002-2723-2077
[a]ITMO University, 49, Kronverksky Pr., 197101, Saint-Petersburg, Russian Federation
[b]North-West Institute of Management — Branch of the RANEPA, 57/43, Sredny Pr., V. O., 199178, Saint-Petersburg, Russian Federation

**Introduction:** We consider the problem of routing of piece cargo by a conveyor system. When moving cargo pieces, it is necessary not only to minimize the time of transportation, but also to minimize the energy spent on it. **Purpose:** Development of a routing algorithm that is adaptive to changes in the topology of the routing graph and is able to optimize the delivery time and the consumed energy. **Results:** We propose an algorithm based on multi-agent deep reinforcement learning that places agents at the vertices of a conveyor network graph and uses a new state value function. The algorithm has two tunable parameters: the length of the path along which the state value function is calculated, and the learning coefficient. Through the selection of parameters, we have revealed that the optimal values are 2 and 1, respectively. An experimental study of the algorithm using a simulation model has shown that it allows to reduce the number of collisions of moving objects to zero, demonstrates stable results for both optimized scores, and also leads to a lower energy consumption compared with the method used as a baseline. **Practical relevance:** The proposed algorithm can be used to reduce delivery time and energy when managing conveyor systems.
**Keywords** — routing, multi-agent learning, reinforcement learning, conveyor belt.

**For citation:** Zhurba Y. S., Filchenkov A. A., Azarov A. A., Shalyto A. A. Continuous control algorithms for conveyer belt routing based on multi-agent deep reinforcement learning. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2022, no. 6, pp. 10–19 (In Russian). doi:10.31799/1684-8853-2022-6-10-19, EDN: LKVJNA

**Financial support**
The study was supported by a grant from the Russian Science Foundation (project No. 20-19-00700).

**References**

1. Winkelhaus S., Grosse E. H. Logistics 4.0: A systematic review towards a new logistics system. *International Journal of Production Research*, 2020, vol. 58, iss. 1, pp. 18–43. doi:10.1080/00207543.2019.1612964
2. Toth P., Vigo D. *An overview of vehicle routing problems*. In: *The Vehicle Routing Problem*. Eds. P. Toth, D. Vigo. SIAM, 2002. Pp. 1–26. doi:10.1137/1.9780898718515.ch1
3. Sweda T. M., Dolinskaya I. S., Klabjan D. Adaptive routing and recharging policies for electric vehicles. *Transportation Science*, 2017, vol. 51(4), pp. 1326–1348. doi:10.1287/trsc.2016.0724
4. Puthal M. K., Singh V., Gaur M. S., Laxmi V. C-Routing: An adaptive hierarchical NoC routing methodology. *2011 IEEE/IFIP 19th Intern. Conf. on VLSI and System-on-chip*, IEEE, 2011, pp. 392–397. doi:10.1109/VLSISoC.2011.6081616
5. Marasova D., Andrejiova M., Grincova A. Dynamic model of impact energy absorption by a conveyor belt in interaction with the support system. *Energies*, 2021, vol. 15(1), p. 64. doi:10.3390/en15010064
6. Köken E., Lawal A. I., Onifade M., Özarslan A. A comparative study on power calculation methods for conveyor belts in mining industry. *International Journal of Mining, Reclamation and Environment*, 2022, vol. 36, iss. 1, pp. 26–45. doi:10.1080/17480930.2021.1949859
7. Karami F., Fathi M., Pardalos P. M. Conveyor operations in distribution centers: Modeling and optimization. *Optimization Letters*, 2022. doi:10.1007/s11590-022-01912-7. Available at: https://link.springer.com/article/10.1007/s11590-022-01912-7 (accessed 11 September 2022).
8. Black G., Vyatkin V. Intelligent component-based automation of baggage handling systems with IEC 61499. *IEEE Transactions on Automation Science and Engineering*, 2009, vol. 7, iss. 2, pp. 337–351. doi:10.1109/TASE.2008.2007216
9. Dorri A., Kanhere S. S., Jurdak R. Multi-agent systems: A survey. *IEEE Access*, 2018, no. 6, pp. 28573–28593. doi:10.1109/ACCESS.2018.2831228
10. Arulkumaran K., Deisenroth M. P., Brundage M., Bharath A. A. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 2017, vol. 34, iss. 6, pp. 26–38. doi:10.1109/MSP.2017.2743240
11. Ma D., Carpenter N., Amatya S., Maki H., Wang L., Zhang L., Neeno S., Tuinstra M. R., Jin J. Removal of greenhouse microclimate heterogeneity with conveyor system for indoor phenotyping. *Computers and Electronics in Agriculture*, 2019, no. 166. doi:10.1016/j.compag.2019.104979. Available at: https://www.sciencedirect.com/science/article/abs/pii/S016816991930691X (accessed 11 September 2022).
12. Halepoto I. A., Shaikh M. Z., Chowdhry B. S., Uqaili M. A. Design and implementation of intelligent energy efficient conveyor system model based on variable speed drive control and physical modeling. *International Journal of Control and Automation*, 2016, vol. 9, iss. 6, pp. 379–388. doi:10.14257/ijca.2016.9.6.36
13. Noto M., Sato H. A method for the shortest path search by extended Dijkstra algorithm. *2000 IEEE Intern. Conf. on Systems, Man and Cybernetics "Cyberbetics Evolving to Systems, Humans, Organizations, and their Complex Interactions" (SMC 2000 Conf. Proc.)*, IEEE, 2000, vol. 3, pp. 2316–2320. doi:10.1109/ICSMC.2000.886462
14. Mammeri Z. Reinforcement learning based routing in networks: Review and classification of approaches. *IEEE Ac-*

*cess*, 2019, no. 7, pp. 55916–55950. doi:10.1109/ACCESS.2019.2913776
15. Boyan J., Littman M. Packet routing in dynamically changing networks: A reinforcement learning approach. *NIPS'93: Proc. of the 6th Intern. Conf. on Neural Information Processing Systems*, 1993, pp. 671–678.
16. Sutton R. S., Barto A. G. *Reinforcement Learning: An Introduction*. 2nd ed. MIT Press, 2018. 552 p.
17. Mukhutdinov D., Filchenkov A., Shalyto A., Vyatkin V. Multi-agent deep learning for simultaneous optimization for time and energy in distributed routing system. *Future Generation Computer Systems*, 2019, no. 94, pp. 587–600. doi:10.1016/j.future.2018.12.037
18. Mnih V., Kavukcuoglu K., Silver D., Graves A., Antonoglou I., Wierstra D., Riedmiller M. *Playing Atari with Deep Reinforcement Learning*. arXiv preprint, 2013. Available at: https://arxiv.org/abs/1312.5602 (accessed 11 September 2022).
19. Mukhutdinov D. *Detsentralizovannyy algoritm upravleniya konveyernoy sistemoy s isplozovaniyem metodov multiagentnogo obucheniya s podkrepleniem* [Decentralized conveyor system control algorithm using methods of multi-agent reinforcement learning. Master diss.] Saint-Petersburg, Universitet ITMO Publ., 2020. 92 p. (In Russian). Available at: http://is.ifmo.ru/diploma-theses/2019/2_54584647710261 91430.pdf. (accessed 11 September 2022).
20. Belkin M., Niyogi P. Laplacian eigenmaps and spectral techniques for embedding and clustering. *NIPS'01: Proc. of the 14th Intern. Conf. on Neural Information Processing Systems: Natural and Synthetic*, 2001, pp. 585–591. doi:10.5555/2980539.2980616

## ПАМЯТКА ДЛЯ АВТОРОВ

*Поступающие в редакцию статьи проходят обязательное рецензирование.*

При наличии положительной рецензии статья рассматривается редакционной коллегией. Принятая в печать статья направляется автору для согласования редакторских правок. После согласования автор представляет в редакцию окончательный вариант текста статьи.

Процедуры согласования текста статьи могут осуществляться как непосредственно в редакции, так и по e-mail (ius.spb@gmail.com).

При отклонении статьи редакция представляет автору мотивированное заключение и рецензию, при необходимости доработать статью — рецензию.

*Редакция журнала напоминает, что ответственность за достоверность и точность рекламных материалов несут рекламодатели.*