УДК 004.85

Обзорная статья

https://doi.org/10.32603/2071-8985-2024-17-10-28-39

Анализ перспектив обучения умных автономных логистических систем на основе оптимизации функции ценности

Н. А. Верзун✉, М. О. Колбанев, А. Р. Салиева

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина), Санкт-Петербург, Россия

✉verzun.n@unecon.ru

Аннотация. Цель статьи состоит во всестороннем анализе и классификации методов обучения с подкреплением по различным критериям для выявления их преимуществ, недостатков и областей эффективного применения. Особое внимание уделяется анализу методов с оптимизацией ценности: *Q*-Learning, SARSA и Deep *Q*-Network. Описаны преимущества и недостатки каждого метода в контексте их использования в умных автономных логистических системах. Рассмотрены примеры успешного использования методов обучения с подкреплением с оптимизацией ценности в сфере логистики; выявляются наиболее перспективные направления их применения; формулируются рекомендации по выбору того или иного метода для решения задач в автономных логистических системах.

Ключевые слова: автономные логистические системы, обучение с подкреплением, оптимизация функции ценности, *Q*-Learning, SARSA, Deep *Q*-Network

Для цитирования: Верзун Н. А., Колбанев М. О., Салиева А. Р. Анализ перспектив обучения умных автономных логистических систем на основе оптимизации функции ценности // Изв. СПбГЭТУ «ЛЭТИ». 2024. Т. 17, № 10. С. 28–39. doi: 10.32603/2071-8985-2024-17-10-28-39.

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Review article

Analysis Learning Prospects of Smart Autonomous Logistics Systems Based on Value Function Optimization

N. A. Verzun✉, M. O. Kolbanev, A. R. Salieva

Saint Petersburg Electrotechnical University, Saint Petersburg, Russia

✉verzun.n@unecon.ru

Abstract. Autonomous logistic systems require effective decision-making techniques under the conditions of uncertainty and dynamically changing environments. Reinforcement learning is a promising approach that allows systems to search for optimal action strategies autonomously. Such systems do not need to accumulate additional historical data, compared to systems based on other methods. When reinforcement learning is applied, the system learns to make decisions based on analyzing its own errors, which can be useful for logistics. There exists a diversity of methods of reinforcement learning, making the selection of the most appropriate method for a particular task an important problem. In this work, we set out to analyze and classify reinforcement learning methods according to various criteria in order to identify their advantages, disadvantages, and areas of effective application. Special attention is paid to the analysis of methods with value optimization: *Q*-Learning, SARSA, and Deep *Q*-Network. The advantages and disadvantages of each method are described in

the context of logistic problems; examples of their successful application in the field of logistics are considered. The most promising directions of their application are identified; recommendations on the selection of a particular method for solving problems in autonomous logistic systems are formulated.

Keywords: autonomous logistics systems, reinforcement learning, value function optimization, *Q*-Learning, SARSA, Deep *Q*-Network

For citation: Verzun N. A., Kolbanev M. O., Salieva A. R. Analysis Learning Prospects of Smart Autonomous Logistics Systems Based on Value Function Optimization // LETI Transactions on Electrical Engineering & Computer Science. 2024. Vol. 17, no. 10. P. 28–39. doi: 10.32603/2071-8985-2024-17-10-28-39.

Conflict of interest. The authors declare no conflicts of interest.

Введение. В современном мире применение автономных логистических систем приобретает все большую актуальность, позволяя оптимизировать процессы транспортировки, складирования и распределения товаров. Одной из ключевых задач в развитии таких систем становится создание эффективных алгоритмов принятия решений, способных адаптироваться к динамически изменяющимся условиям и находить оптимальные стратегии действий.

Обучение с подкреплением (*Reinforcement Learning*, *RL*) представляет собой перспективный подход к решению данной проблемы. Методы RL позволяют агентам обучаться на основе взаимодействия со средой, получая награды за успешные действия и штрафы за ошибки. Такой подход дает возможность автономным системам самостоятельно находить оптимальные политики поведения без явного программирования.

Однако, несмотря на значительный потенциал RL в области автономной логистики, выбор наиболее подходящего метода для конкретной задачи остается нетривиальной проблемой. Существует множество различных алгоритмов RL, каждый из которых обладает своими особенностями, преимуществами и недостатками.

Данная статья ставит своей целью проведение классификации методов обучения с подкреплением, применяемых в автономных логистических системах и сравнительного анализа алгоритмов value optimization. Результаты исследования позволят выделить наиболее перспективные подходы и определить направления их эффективного применения в умных автономных логистических системах.

Автономные системы и сферы их применения. Автономные системы (АС) – это сложные технические устройства, способные функционировать без постоянного контроля со стороны человека. Они могут самостоятельно выполнять задачи в рамках своих возможностей и даже обучаться в процессе работы [1].

Сферы применения автономных систем:

– промышленность. АС широко применяются для автоматизации процессов. Это может включать в себя автономные производственные линии, роботизированные системы и другие технологии, которые позволяют сократить участие человека в процессе производства;

– медицина. В медицинской сфере АС используются для диагностики и лечения заболеваний [2]. Примеры включают автономные медицинские устройства, например искусственные сердца или имплантаты, которые работают без постоянного контроля со стороны врачей;

– транспорт. Автономные транспортные средства – беспилотные автомобили, представляют один из наиболее известных примеров АС [3]. Они способны передвигаться без участия водителя, что повышает безопасность дорожного движения и уменьшает аварийность на дорогах;

– сельское хозяйство. АС применяют для оптимизации использования земельных ресурсов [4]. Это могут быть автономные сельскохозяйственные машины, которые автоматически обрабатывают поля, поливают растения и собирают урожай.

Логистические автономные системы – это комплекс технических средств, обеспечивающих автоматизацию управления материальными потоками в процессе снабжения, производства и распределения продукции. Такие системы включают в себя различные виды транспорта, погрузочно-разгрузочное оборудование, склады и информационные технологии [5]. Логистические АС используются для оптимизации процессов доставки товаров от производителя к потребителю, минимизации затрат на транспортировку и хранение, а также для повышения эффективности деятельности всей цепочки поставок.

Примеры логистических автономных систем:

– роботизированные склады [6] – автоматизированные системы хранения и поиска товаров, которые работают без участия человека. Они мо-

гут быстро находить нужный товар и доставлять его на конвейерную линию или в зону отгрузки;

– автономные грузовики [7] – транспортные средства, оснащенные системами автоматического управления, позволяющими транспорту двигаться по заранее заданному маршруту без участия водителя, что значительно снижает риск аварий и повышает оперативность перевозок;

– дроны-доставщики [8] – беспилотные летательные аппараты, которые могут доставлять небольшие грузы на короткие расстояния. Они особенно полезны в труднодоступных районах, в условиях отсутствия дорог;

– системы автоматического управления складом [5] – это аппаратно-программные комплексы, управляющие всеми процессами на складе: от приема товара до его отправки. Подобные системы позволяют оптимизировать использование пространства, сократить время выполнения заказов и уменьшить вероятность ошибок.

С развитием технологий АС возникает необходимость в поиске эффективных методов и стратегий, которые позволят системам подстраиваться под изменения окружающей среды и принимать решения на основе полученного опыта. В этом контексте методы обучения с подкреплением становятся ключевым инструментом для обучения и улучшения работы АС. Методы RL позволяют системам собирать данные из окружающей среды, анализировать их и принимать оптимальные решения на основе полученной информации. Такой подход открывает новые перспективы для повышения эффективности и надежности АС в самых различных областях применения, начиная от промышленности и медицины и заканчивая транспортом и сельским хозяйством.

Обучение с подкреплением. RL – это область машинного обучения, в которой *агент* (*agent*) обучается взаимодействовать с *окружающей средой (environment)*, выполняя *действия* (*action*) и получая за них *вознаграждения* (*reward*). Основная цель агента в RL – максимизировать суммарное вознаграждение за определенный период времени. На рис. 1 представлена обобщенная схема обучения с подкреплением [9].

Такое обучение представляет собой процесс, в котором агент, выбирая действия на основе текущего *состояния* (*state*), взаимодействует с окружающей средой с целью максимизировать получаемое вознаграждение и избегать наказания. Стратегия, которую агент применяет для выбора действия на основе текущего состояния, называется *политикой* (*policy*). В результате обучения агент стремится оптимизировать свое поведение, принимая во внимание предыдущий опыт и обратную связь от окружающей среды.

Обучение с подкреплением обладает рядом преимуществ по сравнению с другими методами машинного обучения, которые делают его особенно полезным для решения задач, требующих динамической адаптации к изменяющимся условиям [10]. Выделим наиболее существенные преимущества:

– *адаптивность*. Обучение с подкреплением позволяет системе искусственного интеллекта (ИИ) адаптироваться к динамическим изменениям в окружающей среде. Это весьма важно в логистике, где условия могут изменяться быстро и непредсказуемо (*unpredictably*);

– *возможность учета сложных факторов*. RL может учитывать множество факторов. Так, в логистике это могут быть: загрузка производственных мощностей, транспортные расходы, спрос на товары и пр., что позволяет системе ИИ динамически распределять заказы для обеспечения оптимальной загрузки производственных мощностей, складских помещений, уменьшения транспортных расходов;

– *отсутствие необходимости предварительной маркировки данных* или четкого определения целей. Вместо этого ИИ-агент учится через взаимодействие с окружающей средой и получает обратную связь в виде наград или штрафов за свои действия;

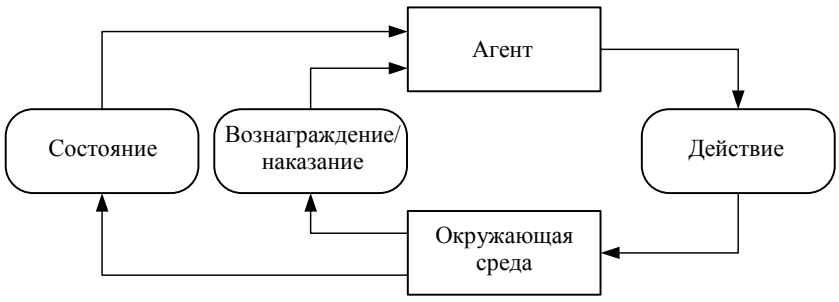

Рис. 1. Обобщенная схема обучения с подкреплением
Fig. 1. Generalized scheme of reinforcement learning

– *возможность обучать агентов оптимальному поведению*. Обучение с подкреплением позволяет обучать оптимальному поведению агентов в мультиагентных средах с большим числом агентов;

– *менее жесткие требования к вычислительным ресурсам*. Используемая в обучении модель мира позволяет обучать оптимальное поведение агентов в десятки раз быстрее текущих аналогов, что значительно уменьшает требования к вычислительным мощностям;

– *обучение агентов может осуществляться на основе их опыта*;

– *возможность адаптации агентов к новым ситуациям*. RL позволяет агентам адаптироваться к новым условиям, что особенно полезно при использовании в непредсказуемых ситуациях.

Классификация методов обучения с подкреплением. Разнообразие типов решаемых задач, условий окружающей среды, особенностей

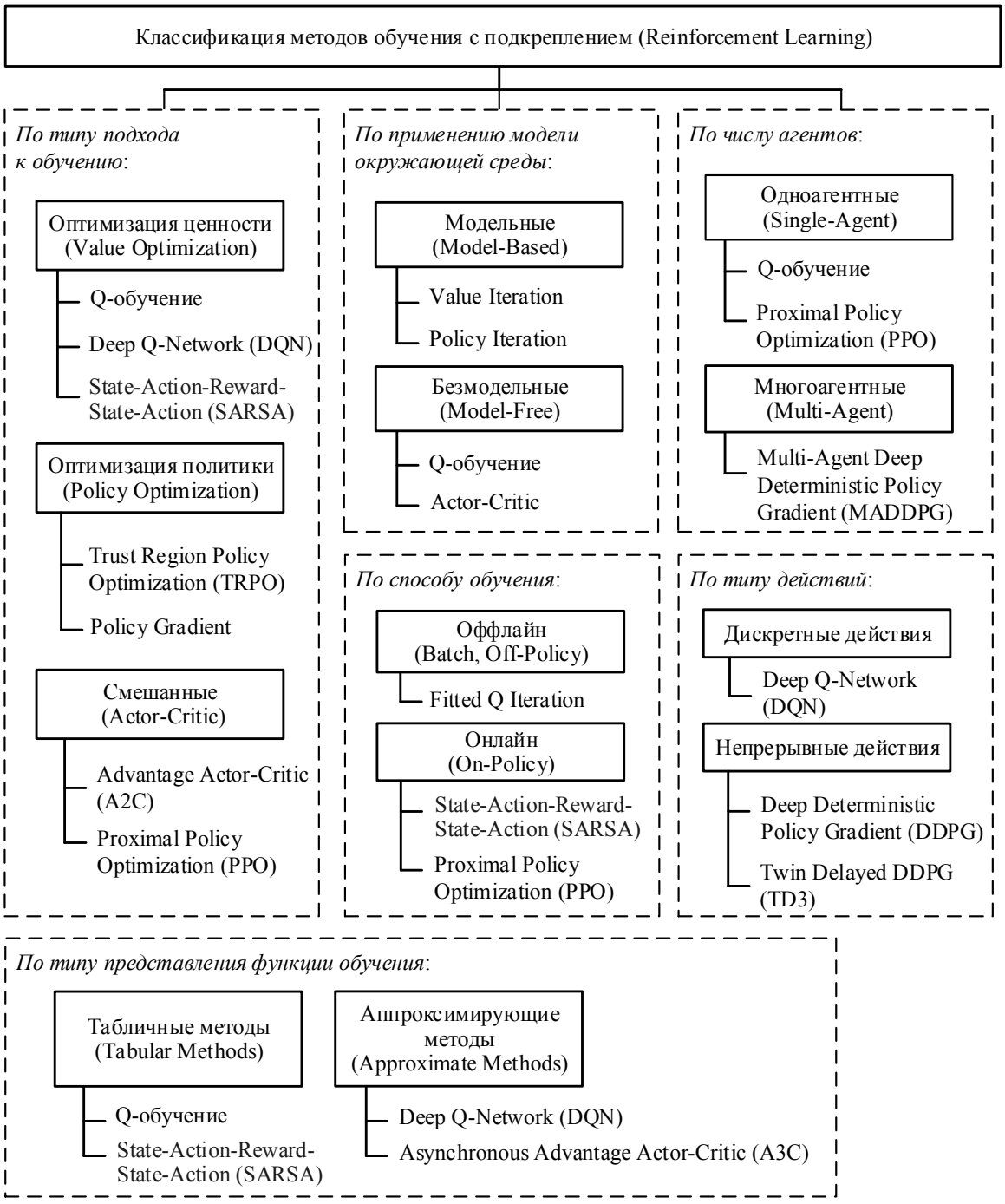

Рис. 2. Классификация методов обучения с подкреплением
Fig. 2. Classification of reinforcement learning methods

стратегии взаимодействия с окружающей средой обусловило создание множества различных алгоритмов обучения с подкреплением. Классификация методов обучения с подкреплением по различным критериям приведена на рис. 2.

Классифицировать методы RL можно по следующим признакам:

1. По типу *подхода к обучению*:

– оптимизация ценности (Value Optimization). Методы RL данного типа (Value-Based) основаны на оптимизации функции ценности (value function), которая заключается в том, чтобы найти политику поведения (policy), дающую максимум ожидаемой награды (reward) [11]. Например, оценивают *Q*-функцию для принятия решений. Пример: *Q*-обучение, State-Action-Reward-State-Action (SARSA), Deep *Q*-Network (DQN);

– оптимизация политики (Policy Optimization). В основе этих методов RL оптимизация функции политики (policy function). Цель: найти такую политику поведения, которая минимизирует ожидаемое количество ошибок [12]. Алгоритмы на основе политики (Policy-Based) обучают непосредственно политику, а не ценность действий. Пример: метод градиента политики (Policy Gradient), Trust Region Policy Optimization (TRPO);

– смешанные (Actor-Critic). Комбинация обоих подходов, где актор (actor) обучает политику, а критик (critic) оценивает ценность. Пример: Advantage Actor-Critic (A2C), Proximal Policy Optimization (PPO).

2. По *применению модели окружающей среды*:

– *моделированное обучение с подкреплением* (*model-based reinforcement learning*). Модельные (Model-Based) алгоритмы строят модель среды и используют ее для планирования и принятия решений. Пример: алгоритмы динамического программирования – методы итерации ценности (Value Iteration) и итерации политики (Policy Iteration);

– *безмодельное обучение с подкреплением* (*model-free reinforcement learning*). Безмодельные (Model-Free) алгоритмы не требуют явной модели среды и обучаются через непосредственное взаимодействие со средой. Пример: *Q-обучение* (*Q*-Learning), метод актор-критик (Actor-Critic).

3. По *числу агентов*:

– *одноагентные* (*Single-Agent*). Алгоритмы, разработанные для одного агента, взаимодействующего со средой. Пример: *Q*-Learning, PPO;

– *многоагентные* (*Multi-Agent*). Алгоритмы, разработанные для взаимодействия нескольких агентов в одной среде. Пример: Multi-Agent Deep Deterministic Policy Gradient (MADDPG).

4. По *способу обучения*:

– *оффлайн* (*Batch*, *Off-Policy*). Алгоритмы, которые обучаются на фиксированном наборе данных, собранном заранее, без необходимости непрерывного взаимодействия с реальной средой. Пример: Fitted *Q* Iteration;

– *онлайн* (*On-Policy*). Алгоритмы, которые обучаются в реальном времени, взаимодействуя с окружающей средой и адаптируются к изменениям. Пример: SARSA, PPO.

5. По типу *представления функции обучения*:

– *табличные методы* (*Tabular Methods*). Методы, использующие таблицы для хранения ценностей или политик для каждого состояния или состояния–действия. Пример: *табличное Q*-обучение, SARSA;

– *аппроксимирующие методы* (*Approximate Methods*). Методы, использующие функции аппроксимации – такие, как нейронные сети, для оценки ценностей или политик. Пример: DQN, Asynchronous Advantage Actor-Critic (A3C). A3C – представляет собой развитие Actor-Critic алгоритма и используется для тренировки агента в средах с непрерывным или дискретным пространством действий.

6. По *типу действий*:

– *дискретные*: алгоритмы, разработанные для сред, где набор возможных действий ограничен и дискретен. Пример: DQN;

– *непрерывные*: алгоритмы, предназначенные для сред с непрерывным пространством действий. Пример: Deep Deterministic Policy Gradient (DDPG), Twin Delayed DDPG (TD3).

Сравнительный анализ методов обучения с подкреплением типа Value Optimization. Методы обучения с подкреплением с оптимизацией ценности можно разделить на два типа:

– *табличные* методы. К ним можно отнести *Q*-Learning и SARSA. В табличных методах используется *Q*-таблица, где каждая ячейка хранит оценку *Q* для определенной пары состояние–действие;

– методы, применяющие нейронные сети для аппроксимации *Q*-функции, – например, DQN.

Рассмотрим подробнее суть каждого метода.

Q-Learning – это безмодельный алгоритм обучения с подкреплением, реализующий обучение вне политики. *Q*-Learning использует функцию ценности (*Q*-функцию) для оценки ожидаемого вознаграждения для каждой пары состояние–действие [13]. *Q*-функция, или функция ценности действия, оценивает качество действий

агента в конкретных состояниях, предоставляя меру ожидаемого вознаграждения за выполнение определенного действия в данном состоянии и следование определенной стратегии (policy). *Q*-функция используется для оценки или предсказания ожидаемой награды или возвращения (return) при выборе определенного действия в конкретном состоянии среды.

Формула обновления алгоритма обучения в данном случае [14]

$$Q_{(s,\,a)} \leftarrow Q(s,\,a)Q_{(s,\,a)} + \\ + \alpha\left[r + \gamma \max_{a'} Q(s',\,a') - Q(s,\,a)\right],$$

где $Q_{(s,\,a)}$ – текущее значение *Q*-функции для состояния s и действия a; α – скорость обучения (learning rate), $0 < \alpha \le 1$, определяет, насколько сильно новые значения обновляют старые; r – вознаграждение, полученное после выполнения действия a в состоянии s; γ – коэффициент дисконтирования (discount factor), $0 \le \gamma < 1$, определяет важность будущих вознаграждений; s' – новое состояние, в которое агент попадает после выполнения действия a в состоянии s; $Q(s,\,a)$ – обновляемое значение *Q*-функции; $\max_{a'} Q(s'a')$ – максимальное значение *Q*-функции для всех возможных действий a' в новом состоянии s'.

Q-Learning – простой и эффективный алгоритм, который хорошо работает для задач с дискретными и малыми пространствами состояний и действий благодаря своей математической стабильности, независимости от модели среды, низким вычислительным затратам и способности к быстрой адаптации и обучению.

State-Action-Reward-State-Action (***SARSA***) – это алгоритм обучения с подкреплением типа on policy, который обновляет *Q*-значения на основе действий, выполняемых текущей политикой, т. е. реально выполняемых агентом. Формула обновления алгоритма [15]:

$$Q_{(s_t,\,a_t)} \leftarrow Q(s_t,\,a_t) + \\ + \alpha[r_{t+1} + \gamma Q(s_{t+1}a_{t+1}) - Q(s_t a_t)],$$

где $Q_{(s_t,\ a_t)}$ – текущее значение *Q*-функции для состояния s_t и действия a_t; r_{t+1} – вознаграждение, полученное после выполнения действия a_t в состоянии s_t; s_{t+1} – новое состояние, в которое агент попадает после выполнения действия a_t в состоянии s_t; a_{t+1} – следующее действие, выбранное агентом в новом состоянии s_{t+1}.

SARSA учитывает политику агента на каждом шаге и обновляет *Q*-значения в зависимости от действий, действительно выполняемых агентом [15], что делает его полезным в средах, где важно безопасное обучение, так как алгоритм не основывается на гипотетически лучших действиях, а учитывает реальные действия и их последствия. Также SARSA прост в реализации, обеспечивает стабильность и согласованность обучения, поддерживает непрерывное улучшение и эффективно работает в различных типах сред.

Deep Q-Network (***DQN***) – это усовершенствованный алгоритм обучения с подкреплением, который использует нейронные сети для аппроксимации *Q*-функции, что позволяет справляться с большими и непрерывными пространствами состояний [11]. DQN вводит несколько ключевых улучшений для стабилизации обучения и повышения эффективности.

Основные элементы DQN включают в себя:

– *повторение опыта* (*Experience Replay*) – хранение и случайная выборка опыта для уменьшения корреляции между последовательными обучающими примерами;

– *фиксированную целевую сеть* (*Fixed Target Network*) – использование отдельной целевой сети для стабилизации обновлений *Q*-значений;

– *формулу обновления алгоритма* [16]:

$$\theta_i \leftarrow \theta_i + \alpha[r + \gamma \max_{a'} Q(s',\,a';\,\theta^-) - \\ - Q(s,a;\theta)]\,\nabla\theta_i Q(s,\,a;\,\theta),$$

где θ – параметры основной сети; θ^- – параметры целевой сети; a' – следующее действие; $\nabla\theta_i Q(s,a;\theta)$ – градиент функции потерь относительно параметров сети; $Q(s,a;\theta)$ – оценка *Q*-значения для текущего состояния s и действия a, вычисленная основной сетью с параметрами θ; $Q(s',a';\theta^-)$ – оценка *Q*-значения для следующего состояния s' и действия a', вычисленная целевой сетью с параметрами θ';

– *основная сеть* (*Q-Network*) – нейронная сеть, аппроксимирующая *Q*-значения для каждого состояния–действия;

– *целевая сеть* (*Target network*) – копия основной сети, обновляемая периодически для стабилизации обучения.

DQN позволил значительно улучшить производительность в задачах обучения с подкреплением и стал основой для многих последующих алгоритмов, обеспечивая хорошую производи-

Ключевые различия между алгоритмами типа Value Optimization
Key differences between Value Optimization algorithms

Характеристика	*Q*-Learning / SARSA	Deep *Q*-Network (DQN)
Аппроксимация функции ценности	Таблица *Q*	Нейронная сеть
Пространство состояний	Дискретное и малое	Большое и непрерывное
Обучение	Табличное, легко реализуемое	Нейронная сеть, сложное обучение
Память	Требует много памяти для больших пространств	Требует памяти для хранения опыта, но меньше памяти для хранения *Q*-значений
Сходимость	Гарантированное схождение для дискретных задач	Может быть проблемой из-за нестабильности
Преимущества	Простота, гарантированное схождение	Масштабируемость, автоматическое обучение признаков
Недостатки	Не масштабируются на большие пространства	Требует много вычислительных ресурсов, сложность настройки

тельность в сложных и высокоразмерных средах благодаря использованию нейронных сетей для аппроксимации *Q*-функции, стабилизации обучения с помощью фиксированного целевого *Q*-сетапа и метода опыта повторного воспроизведения, что позволяет более эффективно использовать прошлые опыты и уменьшать корреляцию между последовательными обучающими примерами – *Double DQN*, *Dueling DQN*, *Rainbow* [17] и т. п.

Сравнительный анализ трех рассмотренных алгоритмов обучения представлен в таблице.

Применение алгоритмов типа Value Optimization в автономных логистических системах.

Табличные методы. *Q*-Learning и SARSA могут использоваться для задач с дискретными состояниями и действиями. Часто применяются для определения оптимальных маршрутов в сетке (например, внутри склада) или управления движением роботов по заранее определенным маршрутам.

Применение *Q*-Learning в сфере логистики:

– при решении задачи оптимизации маршрутов автоматизированных транспортных средств внутри складов [13]. Исследование показало, что использование *Q*-Learning позволяет значительно уменьшить время выполнения логистических операций и повысить общую эффективность системы;

– в [18] рассмотрена оптимизация маршрутов автономных грузовиков. *Q*-Learning можно использовать для обучения грузовиков выбирать оптимальные маршруты в зависимости от текущего состояния дорог, трафика и времени доставки;

– управление инвентарем на складе [19]: алгоритм поможет оптимизировать размещение товаров в складских помещениях, учитывая частоту заказа товара и сезонные тенденции;

– оптимизация загрузки транспортных средств, например, контейнеров [20]. В данном случае *Q*-learning может помочь в размещении грузов в контейнерах, минимизируя пустое пространство и максимизируя устойчивость.

Применение SARSA в сфере логистики:

– динамическое планирование доставки [21], основанное на SARSA позволяет адаптироваться в реальном времени к изменениям, например к появлению новых заказов или отмена существующих, корректируя маршруты доставки;

– управление автономными роботами на складе [22] с использованием SARSA основано на обучение роботов-погрузчиков выбирать оптимальные пути и избегать столкновений;

– оптимизация цепочки поставок SARSA [23] может помочь в принятии решений о том, когда и сколько заказывать товаров, учитывая спрос, задержки поставок и стоимость хранения.

Методы, применяющие нейронные сети. Deep *Q*-Network представляет собой более сложный алгоритм, чем табличные методы. DQN использует нейронные сети для работы с непрерывными состояниями и действиями, и подходит для более масштабных задач, требующих обработки большого объема данных и сложных вычислений.

Применение DQN в сфере логистики:

– управление беспилотными летательными аппаратами (дронами) в условиях плотной застройки. Исследование [24] продемонстрировало, что использование DQN позволяет дронам успешно избегать препятствий и находить оптимальные маршруты для доставки товаров;

– автономная сортировка посылок [25]. Применение DQN для обучения роботов-манипуляторов сортировать посылки по размеру, хрупкости и приоритету доставки;

– оптимизация энергопотребления в холодильных складах [26]. DQN позволяет управлять системами охлаждения, балансируя между сохра-

нением качества продуктов и минимизацией энергозатрат;

– предсказание спроса и управление запасами [27]. Используя DQN, система анализирует множество факторов (включая исторические данные, сезонность, маркетинговые акции) для точного прогнозирования спроса и автоматической корректировки уровней запасов;

– динамическое распределение ресурсов в портах [28]. DQN помогает оптимизировать назначение причалов, кранов и других ресурсов для минимизации времени обработки судов;

– автономная навигация дронов для доставки. DQN может обучать дроны выбирать оптимальные маршруты с учетом погодных условий, батареи и зон ограниченного полета.

Рекомендации по применению методов обучения типа Value Optimization для автономных логистических систем. Опираясь на проведенный сравнительный анализ и обзор известных успешных примеров использования в автономных логистических системах [19]–[28] различных методов обучения с подкреплением типа Value Optimization можно следующим образом охарактеризовать типичные задачи для них:

***Q-Learning*:**

1. Подходит для задач, связанных с исследованием неизвестной среды. Хорошо работает, когда структура и функционирование логистической среды не полностью известны. Алгоритм может исследовать различные действия и обновлять свои оценки *Q*-values на основе получаемых вознаграждений.

2. Оффлайн-обучение. В случае, когда доступен большой объем данных или симуляций, *Q*-Learning может быть эффективен для обучения без активного взаимодействия с реальной средой. Это позволяет минимизировать риски и оптимизировать стратегии до внедрения в реальные условия.

3. Эксплуатация и исследование. *Q*-Learning имеет механизм для балансировки между эксплуатацией известных знаний (максимизация вознаграждений) и исследованием новых действий (обновление *Q*-values для улучшения стратегии).

Примеры задач для Q-Learning:

– маршрутизация и управление ресурсами – оптимизация маршрутов доставки грузов или управление использованием складских ресурсов.

Данный метод хорошо подходит для случаев, когда возможны различные варианты действий (маршруты, распределение ресурсов) и важно обучение на основе полученных вознаграждений без активного взаимодействия с реальной средой;

– управление транспортными потоками – управление движением автономных транспортных средств (например, беспилотных грузовиков) в динамических условиях дорожного движения.

Этот алгоритм позволяет агентам исследовать различные стратегии перемещения и адаптироваться к изменениям в дорожной среде, оптимизируя время доставки или минимизируя затраты.

***SARSA*:**

1. On-policy-обучение. SARSA подходит, если важна стабильность и сходимость обучения. Алгоритм использует текущую стратегию для выбора действий и обновления *Q*-values, что способствует более плавному обучению по сравнению с *Q*-Learning.

2. Ограниченная разведка. Если необходимо ограничить исследование новых действий в пользу эксплуатации текущих знаний, SARSA может стать предпочтительным выбором.

Пример задачи для SARSA:

Контроль и управление складскими операциями – управление планированием и выполнением операций на складе. Например, перемещение товаров между различными секциями склада.

SARSA подходит для ситуаций, где важна стабильность и сходимость в процессе обучения. Он использует текущую стратегию для выбора действий и обновления *Q*-values, что может быть полезно при управлении складскими операциями с четко определенными правилами и ограничениями.

***DQN*:**

1. Подходит для высокоразмерных или сложных сред, для задач, где пространство состояний и действий очень большое или сложное для традиционных методов RL.

2. Обучение с использованием нейронных сетей. DQN использует глубокие нейронные сети для аппроксимации функции *Q*-values, что позволяет адаптироваться к сложным структурам среды и извлекать более сложные зависимости между состояниями и действиями.

Пример задачи для DQN:

– управление многокритериальной оптимизацией – оптимизация совокупных целей (время доставки, стоимость и экологические параметры в логистических операциях).

DQN может эффективно работать с большими объемами данных и сложными взаимодействиями между различными аспектами логистики. Глубокие нейронные сети, используемые в DQN, спо-

собны выявлять сложные зависимости и адаптироваться к изменениям в среде;

– прогнозирование спроса и управление запасами – прогнозирование спроса на продукцию и управление уровнем запасов на складе.

DQN может быть полезен для управления множеством товаров с различными временными шкалами спроса и сезонными изменениями. Это позволяет оптимизировать уровни запасов и минимизировать издержки хранения при условии изменяющегося спроса.

Проведенный анализ позволяет выделить следующие преимущества применения методов обучения умных автономных логистических систем на основе оптимизации функции ценности:

1. *Эффективность и точность*. Алгоритмы, оптимизирующие функцию ценности, позволяют значительно повысить эффективность и точность в управлении логистическими процессами. Это ведет к оптимальному распределению ресурсов и снижению затрат.

2. *Адаптивность и гибкость*. Алгоритмы способны адаптироваться к изменениям в реальном времени, что обеспечивает гибкость системы и позволяет быстро реагировать на непредвиденные обстоятельства и изменяющиеся условия.

3. *Улучшенное принятие решений*. Оптимизация функции ценности способствует улучшению процесса принятия решений, так как алгоритмы учитывают множество факторов и условий, что позволяет находить наилучшие решения для каждой конкретной ситуации.

4. *Самообучение и развитие*. Благодаря использованию методов машинного обучения, логистические системы могут непрерывно самообучаться и совершенствоваться, что способствует постоянному повышению их производительности и эффективности.

Итак, резюмируя вышенаписанное можно сформулировать следующие рекомендации по выбору подходящего метода обучения в логистических системах.

Для начала использования, в случае, когда точная модель среды неизвестна или она меняется со временем, целесообразно начать с *Q*-Learning, что может быть разумным выбором для первоначального исследования и определения базовых стратегий.

Для стабильного обучения, если необходимо обеспечить стабильность и сходимость в процессе обучения и учитывать текущую стратегию, предпочтителен SARSA.

Для сложных сред с большим пространством действий, для задач с высокой размерностью данных или сложными взаимодействиями, где необходима адаптация и обучение на основе больших объемов информации, подходящим выбором будет DQN, особенно если есть доступ к мощным вычислительным ресурсам для обучения глубоких нейронных сетей.

Заключение. Основываясь на проведенном сравнительном анализе методов обучения с подкреплением с оптимизацией функции ценности, можно сделать вывод, что выбор конкретного метода обучения в автономных логистических системах зависит от ряда факторов – структуры системы, доступных данных, требований к точности и скорости обучения и др.

В статье охарактеризованы типичные задачи для рассмотренных трех методов (табличных: *Q*-Learning и SARSA, и применяющих нейронные сети для аппроксимации *Q*-функции) и сформулированы рекомендации по выбору подходящего метода обучения в логистических системах. В случае, если точная модель среды неизвестна или она меняется со временем, целесообразно применять *Q*-Learning. Когда важно обеспечить стабильность и сходимость в процессе обучения и учитывать текущую стратегию, предпочтителен является SARSA. Для задач с высокой размерностью данных или сложными взаимодействиями, где необходима адаптация и обучение на основе больших объемов информации, лучше применять DQN.

Независимо от выбранного метода, важно учитывать особенности решаемой логистической задачи и стремиться к достижению оптимального баланса между качеством управления и вычислительной сложностью алгоритма.

Дальнейшие исследования в данной области могут быть направлены на разработку более эффективных алгоритмов обучения с подкреплением, учитывающих специфику автономных логистических систем, а также на адаптацию существующих методов к конкретным практическим задачам.

Список литературы

1. Intelligent robotic systems in industry 4.0: A review / M. Soori, R. Dastres, B. Arezoo, F. K. G. Jough // J. of Advanced Manufacturing Sci. and Technol. 2024. Vol. 4, iss. 3. 28 p. doi: 10.51393/j.jamst2024007.

2. Беляев А. О., Кириенко В. В., Убирайло Д. С. Применение технологии Bluetooth для разработки медицинского оборудования с автономным питанием // Ползуновский вестн. 2014. № 2. С. 203–207.

3. Жанказиев С. В., Воробьев А. И., Морозов Д. Ю. Технические и технологические особенности автономных транспортных средств // Транспорт РФ. Журн. о науке, практике, экономике. 2019. № 3(82). С. 39–43.

4. Егоров И. А., Шипулин Н. С., Косников С. Н. Современные технологии автоматизации и роботизации процессов аграрного производства // Региональная и отраслевая экономика. 2023. № 2. С. 8–17.

5. Ермаков А. А. Автономные интралогистические системы. Уровни задач и этапы автоматизации // Актуальные исследования. 2021. № 47(74). С. 19–22.

6. Халын А. В., Халын В. Г. Автоматизация систем складирования на основе внедрения логистических роботов // Вестн. Алтайской академии экономики и права. 2023. № 4–1. С. 133–136.

7. Меретджаева Г., Абаева Г., Абдуллаева С. Беспилотные конвои: будущее транспортной логистики // Вестн. науки. 2024. Т. 3, № 2 (71). С. 101–104.

8. Гранкин Е. Дроны в логистике // Логистика. 2019. № 1(146). С. 14–16.

9. Kaelbling L. P., Littman M. L., Moore, A. W. Reinforcement learning: A survey // J. of Artificial Intelligence Research. 1996. Vol. 4. P. 237–285.

10. Mastering the game of Go with deep neural networks and tree search / D. Silver, A. Huang, Ch. J. Maddison, A. Guez, L. Sifre, G. Van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, D. Hassabis // Nature. 2016. Vol. 529(7587). P. 484–489. doi: 10.1038/nature16961.

11. Human-level control through deep reinforcement learning / V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, D. Hassabis // Nature. 2015. Vol. 518, no. 7540. P. 529–533. doi: 10.1038/nature14236.

12. Continuous control with deep reinforcement learning / T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra // 4th Intern. Conf. on Learning Representations (ICLR 2016). San Juan, Puerto Rico. 2016. P. 1–14. URL: https://arxiv.org/pdf/1509.02971v5 (дата обращения: 16.10.2024).

13. Грессер Л., Кенг В. Л. Глубокое обучение с подкреплением: теория и практика на языке Python. СПб.: Питер, 2022. 416 с.

14. Sutton R. S., Barto A. G. Reinforcement learning: An introduction. 2nd ed. Cambridge, MA: the MIT Press, 2018. 552 p.

15. Рассел С., Норвиг П. Искусственный интеллект: современный подход / пер. с англ. и ред. К. А. Птицына. 2-е изд. М.: ИД «Вильямс», 2006. 1408 с.

16. Lapan M. Deep reinforcement learning hands-on. Birmingham: Packt Publishing, 2018. 548 p.

17. Van Hasselt H., Guez A., Silver D. Deep reinforcement learning with double Q-learning // Nature. 2016. Vol. 529, no. 7587. P. 476–480. URL: https://arxiv.org/pdf/1509.06461 (дата обращения: 16.10.2024).

18. Гудфеллоу Я., Бенджио И., Курвилль А. Глубокое обучение / пер. с англ. А. А. Слинкина. 2-е изд., испр. М.: ДМК Пресс, 2018. 652 с.

19. Trust region policy optimization / J. Schulman, S. Levine, P. Abbeel, M. Jordan, P. Moritz // Proc. of the 32nd Intern. Conf. on Machine Learning. Lille, France: PMLR, 2015. P. 1889–1897. URL: https://proceedings.mlr.press/v37/schulman15.html (дата обращения: 16.10.2024).

20. Proximal policy optimization algorithms / J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov. 2017. P. 1–12. URL: https://arxiv.org/abs/1707.06347 (дата обращения: 16.10.2024).

21. Williams R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning // Machine Learning. 1992. Vol. 8, no. 3–4. P. 229–256. doi: 10.1007/BF00992696.

22. Learning modular neural network policies for multi-task and multi-robot transfer / C. Devin, A. Gupta, T. Darrell, P. Abbeel, S. Levine // Intern. Conf. on Machine Learning (ICML 2016). New York City, NY, USA: JMLR, 2016. P. 1–10. URL: https://arxiv.org/abs/1609.07088 (дата обращения: 16.10.2024).

23. A Survey of imitation learning: Algorithms, recent developments, and challenges / M. Zare, P. M. Kebria, A. Khosravi, S. Nahavandi // IEEE Transactions on Cybernetics. 2024. P. 1–14. doi: 10.1109/TCYB.2024.3395626.

24. Ho J., Ermon S. Generative adversarial imitation learning // Advances in Neural Information Proc. Systems (NIPS 2016). Barcelona, Spain: Curran Associates, Inc., 2016. Vol. 29 P. 1–14. URL: https://doi.org/10.48550/arXiv.1606.03476 (дата обращения: 16.10.2024).

25. Asynchronous methods for deep reinforcement learning / V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, K. Kavukcuoglu // Proc. of the 33rd Intern. Conf. on Machine Learning (ICML 2016). New York City, NY, USA: JMLR, W&CP, 2016. Vol. 48. P. 1–19. URL: https://arxiv.org/pdf/1602.01783 (дата обращения: 16.10.2024).

26. The uncertainty Bellman equation and exploration / B. O'Donoghue, I. Osband, R. Munos, V. Mnih // Proc. of the 35th Intern. Conf. on Machine Learning (ICML 2018). Stockholm, Sweden: PMLR, 2018. P. 3836–3845.

27. Rainbow: Combining Improvements in deep reinforcement learning / M. Hessel, J. Modayil, H. Van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. Azar, D. Silver // Proc. of the AAAI Conf. on Artificial Intelligence. 2018. Vol. 32, no. 1. P. 3215–3222. doi: 10.1609/aaai.v32i1.11796.

28. Distributional Reinforcement Learning with Quantile Regression / W. Dabney, M. Rowland, M. G. Bellemare, R. Munos // Proc. of the AAAI Conf. on Artificial Intelligence. 2018. Vol. 32, no. 1. P. 1–10 doi: 10.1609/aaai.v32i1.11791. URL: https://ojs.aaai.org/index.php/AAAI/article/view/11791 (дата обращения: 16.10.2024).

Информация об авторах

Верзун Наталья Аркадьевна – канд. техн. наук, доцент кафедры «Информационные системы» СПбГЭТУ «ЛЭТИ».
E-mail: verzun.n@unecon.ru
https://orcid.org/0000-0002-0126-2358

Колбанев Михаил Олегович – д-р техн. наук, профессор кафедры «Информационные системы» СПбГЭТУ «ЛЭТИ».
E-mail: mokolbanev@mail.ru
https://orcid.org/0000-0003-4825-6972

Салиева Аделина Рустамовна – аспирант гр. 3933, кафедра «Информационные системы» СПбГЭТУ «ЛЭТИ».
E-mail: rustamovna.a3@gmail.com

References

1. Intelligent robotic systems in industry 4.0: A Review / M. Soori, R. Dastres, B. Arezoo, F. K. G. Jough // J. of Advanced Manufacturing Sci. and Technol. 2024. Vol. 4, iss. 3. 28 p. doi: 10.51393/j.jamst2024007.

2. Beljaev A. O., Kirienko V. V., Ubirajlo D. S. Primenenie tehnologii Bluetooth dlja razrabotki medicinskogo oborudovanija s avtonomnym pitaniem // Polzunovskij vestn. 2014. № 2. S. 203–207. (In Russ.).

3. Zhankaziev S. V., Vorob'ev A. I., Morozov D. Ju. Tehnicheskie i tehnologicheskie osobennosti avtonomnyh transportnyh sredstv // Transport RF. Zhurn. o nauke, praktike, jekonomike. 2019. № 3(82). S. 39–43. (In Russ.).

4. Egorov I. A., Shipulin N. S., Kosnikov S. N. Sovremennye tehnologii avtomatizacii i robotizacii processov agrarnogo proizvodstva // Regional'naja i otraslevaja jekonomika. 2023. № 2. S. 8–17. (In Russ.).

5. Ermakov A. A. Avtonomnye intralogisticheskie sistemy. Urovni zadach i jetapy avtomatizacii // Aktual'nye issledovanija. 2021. № 47(74). S. 19–22. (In Russ.).

6. Halyn A. V., Halyn V. G. Avtomatizacija sistem skladirovanija na osnove vnedrenija logisticheskih robotov // Vestn. Altajskoj akademii jekonomiki i prava. 2023. № 4–1. S. 133–136. (In Russ.).

7. Meretdzhaeva G., Abaeva G., Abdullaeva S. Bespilotnye konvoi: budushhee transportnoj logistiki // Vestn. nauki. 2024. T. 3, № 2 (71). S. 101–104. (In Russ.).

8. Grankin E. Drony v logistike // Logistika. 2019. № 1(146). S. 14–16. (In Russ.).

9. Kaelbling L. P., Littman M. L., Moore, A. W. Reinforcement learning: A survey // J. of Artificial Intelligence Research. 1996. Vol. 4. P. 237–285.

10. Mastering the game of Go with deep neural networks and tree search / D. Silver, A. Huang, Ch. J. Maddison, A. Guez, L. Sifre, G. Van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, D. Hassabis // Nature. 2016. Vol. 529(7587). P. 484–489. doi: 10.1038/nature16961.

11. Human-level control through deep reinforcement learning / V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, D. Hassabis // Nature. 2015. Vol. 518, no. 7540. P. 529–533. doi: 10.1038/nature14236.

12. Continuous control with deep reinforcement learning / T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra // 4th Intern. Conf. on Learning Representations (ICLR 2016). San Juan, Puerto Rico. 2016. P. 1–14. URL: https://arxiv.org/pdf/1509.02971v5 (data obrashhenija: 16.10.2024).

13. Gresser L., Keng V. L. Glubokoe obuchenie s podkrepleniem: teorija i praktika na jazyke Python. SPb.: Piter, 2022. 416 s. (In Russ.).

14. Sutton R. S., Barto A. G. Reinforcement Learning: An Introduction. 2nd ed. Cambridge, MA: the MIT Press, 2018. 552 p.

15. Rassel S., Norvig P. Iskusstvennyj intellekt: sovremennyj podhod / per. s angl. i red. K. A. Pticyna. 2-e izd. M.: ID «Vil'jams», 2006. 1408 s. (In Russ.).

16. Lapan M. Deep Reinforcement learning hands-on. Birmingham: Packt Publishing, 2018. 548 p.

17. Van Hasselt H., Guez A., Silver D. Deep reinforcement learning with double Q-learning // Nature. 2016. Vol. 529, no. 7587. P. 476–480. URL: https://arxiv.org/pdf/1509.06461 (data obrashhenija: 16.10.2024).

18. Gudfellou Ja., Bendzhio I., Kurvill' A. Glubokoe obuchenie / per. s angl. A. A. Slinkina. 2-e izd., ispr. M.: DMK Press, 2018. 652 c. (In Russ.).

19. Trust region policy optimization / J. Schulman, S. Levine, P. Abbeel, M. Jordan, P. Moritz // Proc. of the 32nd Intern. Conf. on Machine Learning. Lille, France: PMLR, 2015. P. 1889–1897. URL: https://proceedings.mlr.press/v37/schulman15.html (data obrashhenija: 16.10.2024).

20. Proximal policy optimization algorithms / J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov. 2017. P. 1–12. URL: https://arxiv.org/abs/1707.06347 (data obrashhenija: 16.10.2024).

21. Williams R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning // Machine Learning. 1992. Vol. 8, no. 3–4. P. 229–256. doi: 10.1007/BF00992696.

22. Learning modular neural network policies for multi-task and multi-robot transfer / C. Devin, A. Gupta, T. Darrell, P. Abbeel, S. Levine // Intern. Conf. on Machine Learning (ICML 2016). New York City, NY, USA: JMLR, 2016. P. 1–10. URL: https://arxiv.org/abs/1609.07088 (data obrashhenija: 16.10.2024).

23. A Survey of imitation learning: Algorithms, recent developments, and challenges / M. Zare, P. M. Kebria, A. Khosravi, S. Nahavandi // IEEE Transactions on Cybernetics. 2024. P. 1–14. doi: 10.1109/TCYB.2024.3395626.

24. Ho J., Ermon S. Generative adversarial imitation learning // Advances in Neural Information Proc. Systems (NIPS 2016). Barcelona, Spain: Curran Associates, Inc., 2016. Vol. 29 P. 1–14. URL: https://doi.org/10.48550/arXiv.1606.03476 (data obrashhenija: 16.10.2024).

25. Asynchronous Methods for deep reinforcement learning / V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, K. Kavukcuoglu // Proc. of the 33rd Intern. Conf. on Machine Learning (ICML 2016). New York City, NY, USA: JMLR, W&CP, 2016. Vol. 48. P. 1–19. URL: https://arxiv.org/pdf/1602.01783 (data obrashhenija: 16.10.2024).

26. The uncertainty bellman equation and exploration / B. O'Donoghue, I. Osband, R. Munos, V. Mnih // Proc. of the 35th Intern. Conf. on Machine Learning (ICML 2018). Stockholm, Sweden: PMLR, 2018. P. 3836–3845.

27. Rainbow: Combining improvements in deep reinforcement learning / M. Hessel, J. Modayil, H. Van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. Azar, D. Silver // Proc. of the AAAI Conf. on Artificial Intelligence. 2018. Vol. 32, no. 1. P. 3215–3222. doi: 10.1609/aaai.v32i1.11796.

28. Distributional reinforcement learning with quantile regression / W. Dabney, M. Rowland, M. G. Bellemare, R. Munos // Proc. of the AAAI Conf. on Artificial Intelligence. 2018. Vol. 32, no. 1. P. 1–10 doi: 10.1609/aaai.v32i1.11791. URL: https://ojs.aaai.org/index.php/AAAI/article/view/11791 (data obrashhenija: 16.10.2024).

Information about the authors

Natalia A. Verzun – Cand. Sci. (Eng.), Associate Professor of the Department of Information systems, Saint Petersburg Electrotechnical University.
E-mail: verzun.n@unecon.ru
https://orcid.org/0000-0002-0126-2358

Mikhail O. Kolbanev – Dr Sci. (Eng.), Professor of the Department of Information systems, Saint Petersburg Electrotechnical University.
E-mail: mokolbanev@mail.ru
https://orcid.org/0000-0003-4825-6972

Adelina R. Salieva – postgraduate student gr. 3933, Department of Information systems, Saint Petersburg Electrotechnical University.
E-mail: rustamovna.a3@gmail.com

Статья поступила в редакцию 01.07.2024; принята к публикации после рецензирования 22.10.2024; опубликована онлайн 25.12.2024.

Submitted 01.07.2024; accepted 22.10.2024; published online 25.12.2024.