Санкт–Петербургский государственный университет

Маркелова Анастасия Юрьевна

Выпускная квалификационная работа

Прикладные задачи оптимизации и алгоритмы управления системами электроснабжения с использованием возобновляемых источников энергии

Уровень образования: бакалавриат
Направление 01.03.02 «Прикладная математика и информатика»
Основная образовательная программа СВ.5005.2017 «Прикладная математика, фундаментальная информатика и программирование»
Кафедра «математического моделирования энергетических систем»

Научный руководитель:
кандидат физ.-мат. наук, доцент
кафедры математического моделирования
энергетических систем:
Петросян Ованес Леонович

Рецензент:
главный специалист-аналитик компании
АО «Совкомбанк страхование»
Бойко Алина Владимировна

Санкт-Петербург
2021 г.

Содержание

Введение

В настоящее время в связи с экологическими проблемами, возросшим спросом на энергию, нестабильной ценовой политикой на топливные ресурсы и нехваткой энергетических мощностей особое внимание было сосредоточено на технологиях распределенной энергетики (РЭ). Распределенные энергетические ресурсы в общем случае – это маломасштабные источники выработки и хранения энергии, расположенные в непосредственной близости к месту использования электроэнергии, они могут обеспечить альтернативу или улучшение традиционной электрической сети. Технологии распределенной энергетики включают в себя газопоршневые, газотурбинные и микротурбинные электростанции, тепловые насосы, паровые котлы, возобновляемую энергетику (солнечные батареи, ветровые генераторы), хранилища энергии, топливные элементы, когенерационные установки и т.д. Вместе они предлагают потребителям потенциал для снижения затрат, электроэнергию высокого качества, энергетическую независимость и повышение энергоэффективности.

Для внедрения и эффективного использования технологий РЭ необходим механизм управления, контролирующий процессы протекающие в энергосистеме пользователя. Система управления энергией или энергоменеджмент – это система, которая управляет компонентами энергосистемы для достижения оптимальной работы в целях снижения затрат на энергетические ресурсы и максимизации эффективности потребления энергии. Энергоменеджмент включает в себя планирование и эксплуатацию энергопроизводящих и энергопотребляющих установок, а также распределение и хранение энергии. Основными целями энергоменеджмента являются: сохранение ресурсов, защита климата и экономия затрат, при условии, что пользователи имеют постоянный доступ к необходимой им энергии.

Внедрение в электроэнергетическую систему (ЭЭС) пользователя технологий РЭ с системой управления формирует концепцию Smart Grid (умных сетей), где потенциальный электропотребитель любого уровня, получает возможность взаимодействовать с ЭЭС: прогнозировать и планировать потребление, выбирать поставщика и влиять на тарифы. Основные

атрибуты концепции Smart Grid определяются следующим образом: доступность, надежность, гибкость, эффективность, обеспечение безопасности, способность к аккумулированию энергии, стимулирование активности электропотребителя, экономичность, снижение экологического давления на окружающую среду.

Возобновляемые источники энергии и системы накопления энергии играют решающую роль в оптимальном планировании работы микросетей. Накопление энергии может повысить гибкость интеллектуальной энергосистемы, а возобновляемые источники энергии обеспечить частичную или полную независимость от коммунального предприятия. Поэтому энергетическая инжиниринговая компания Schneider Electric [1], специализирующаяся на энергетическом менеджменте и автоматизации, опубликовала в открытом доступе данные, которые были предложены для решения задач энергетического менеджмента. Перед исследователями стояла задача разработать оптимизационную модель, минимизирующую затраты на приобретение электроэнергии путем планирования зарядки и разрядки аккумуляторных батарей, а также обмена с энергетическим рынком при условии соблюдения ограничений системы и достижения энергетического баланса. Однако предложенные решения [2] базировались на детерминистических подходах, которые имеют ряд недостатков и не могут быть применены для решения реальных индустриальных задач.

Решение задачи управления распределенными ресурсами невозможно без теоретико-методологического и практического развития подходов, адаптированных под современные системы и запросы. Однако данное научное направление только в последние годы начинает активно развиваться в отечественной науке, что повышает актуальность темы. Поэтому целью моего исследования является разработка быстрого и эффективного инструмента, отвечающего требованиям современного энергоменеджмента.

Постановка задачи

Объект исследования – системы электроснабжения с распределенными энергетическими ресурсами, включающие генерацию возобновляемой энергии от фотоэлектрической станции и систему хранения энергии в

виде аккумуляторных батарей.

Предмет исследования – планирование графика хранения энергии и анализ оптимальных режимов электропотребления в интеллектуальных электрических сетях.

Цель работы – разработка оптимизационной модели, метода и программного модуля позволяющего в режиме реального времени управлять энергосистемой для достижения минимизации финансовых затрат пользователей.

Для достижения цели поставлены и решены следующие задачи:

1. обзор основных классов энергетических задач;
2. обзор существующих подходов и практик для оптимизации, стабилизации и управления электропотреблением;
3. обзор индустриальных решений для задач энергоменеджмента;
4. описание математической постановки задачи в терминах смешанного целочисленного линейного программирования (MILP);
5. предобработка и анализ исходных данных;
6. описание подходов на основе обучения с подкреплением: глубокое Q-обучение (DQN), глубокий детерминированный градиент политики (DDPG);
7. разработка среды и моделей DQN и DDPG;
8. проведение экспериментов и сравнение результатов работы оптимизационных моделей на базе обучения с подкреплением с детерминистическими подходами;
9. разработка программного модуля для комфортного взаимодействия пользователя со средой.

Обзор литературы

Целью данной главы является анализ фундаментальной научной литературы в сфере энергоменеджмента. На Рис. 1 показаны основные направления и методы решения, применяемые для улучшения энергосистемы.

Глобально энергетические задачи оптимизации и управления системами в энергетической сфере можно разделить на два основных направления: теплоэнергетику и электроэнергетику.

Рис. 1: Основные направления оптимизации энергопотребления

Тепловая энергетическая отрасль включает в себя задачи связанные с удовлетворением потребительского спроса в отоплении и охлаждении помещений. Эти проблемы обычно рассматриваются с точки зрения проектирования энергосистемы и управления энергетическими потоками.

Первая проблема имеет стратегический подход для оптимизации инвестиционных затрат при разработке новых сетей, а также для оптимизации инкрементного проектирования и модификации существующих сетей; в то время как вторая проблема направлена на решение задач диспетчеризации или планирования работы систем в существующих сетях с конкретной фиксированной конструкцией.

Электроэнергетическая область может быть разделена на два основных подмножества: системы, связанные с сетью и изолированные системы. Как и в случае с тепловой энергетической отраслью, эти проблемы в основном изучаются для задач проектирования и оперативного управления сетями.

Среди задач оперативного управления выделяются области исследо-

вания, связанные с оптимальным распределением/диспетчеризацией энергии, оптимальным планированием ресурсов энергосистем и оперативного управления спросом. В частности, в литературе широко изучаются проблемы реагирования на спрос, связанные со смещением нагрузок [3] и арбитражными методами [4]. Эти проблемы преимущественно изучаются в сфере электроэнергетики для систем взаимосвязанных с центральными сетями, поскольку возможность обмена электрической энергией с национальной сетью представляет собой важный экономический источник для потребителей. Арбитражные методы необходимы для стимулирования потребителей переносить свои электрические нагрузки в течение дня, используя динамические цены на энергию.

Еще одной областью исследований является сфера когенерации – процесс совместной выработки нескольких видов энергии. Это комбинированные теплоэнергетические системы (CHP), которые являются системами для одновременного производства тепловой и электрической энергии, или комбинированные тепло-охлаждающие и энергетические системы (CHCP), характеризующиеся дополнительным производством холодной энергии за счет использования абсорбционных чиллеров. Оптимальному проектированию [5], размещению [6], управлению и планированию [7] такого рода систем посвящено множество научно-исследовательских работ.

Как для тепловой, так и для электрической энергии, системы накопления играют решающую роль в оптимальном планировании работы микросетей. Как правило, задачи управления аккумуляторными системами рассматриваются в рамках проблем оперативного управления энергетическими сетями, но их следует выделить в отдельную группу, поскольку в настоящее время накопление энергии является центральной темой в научных исследованиях.

Исследования в сфере оптимального управления батареями охватывают широкий спектр научных направлений и задач. Оптимальное управление батареями может фокусироваться на изучении процессов протекающих внутри батареи, тем самым стараясь оптимизировать работу батареи: уменьшить эффект деградации [8], стабилизировать энергию заряда/разряда [9]. Другая область деятельности направлена на изучение

возможностей аккумулятора для оптимизации микросетей. Исследования для накопительных систем в масштабах энергосистемы в основном фокусируется на двух задачах. Первая – это определение оптимального размера батареи для минимизации затрат при проектировании и максимизации эффективности дальнейшего использования батареи [10], [11]. Вторая ориентирована на оптимальное управление батареей для минимизации финансовых затрат или воздействия на окружающую среду. За последние годы было опубликовано несколько исследовательских для управления системами сбережения энергии с применением различных подходов [12] – [20], где для моделирования энергосистемы применяется смешанное целочисленное линейное программирование (MILP). В работах [12], [13] для решения поставленной задачи применяются пакеты программного обеспечения или «решатели». Управление энергией микросетей с использованием нечеткой логики обсуждается в работе [14]. Интеллектуальное управление энергией микросетей с использованием генетического алгоритма обсуждается в работах [15] и [16]. Энергоменеджмент гибридной генерации возобновляемой энергии с использованием ограниченной оптимизации был предложен в работе [17]. Экспертная система и другие классические и эвристические алгоритмы управления энергией микросетей обсуждаются в работах [18], [19] и [20].

Эти исследования не могут учитывать все факторы и проблемы, которые возникают в задачах управления накопительными элементами в гибридных энергосистемах. Проектирование контроллера на основе модели требует точных входных данных и параметров метода для решения задачи. В условиях поставленной задачи это может быть затруднено гетерогенным и динамическим характером потребления электроэнергии и прерывистым характером выработки возобновляемой энергии. К тому же многие алгоритмы требуют больших вычислительных мощностей и не могут быть адаптированы для решения реальных индустриальных задач. Поэтому в качестве решения принимается метод, основанный на обучении с подкреплением (Reinforcement Learning (RL)). Подход RL изучает оптимальные стратегии с помощью механизма проб и ошибок, и не требует описания распределения неопределенностей в наборах данных, так как этот метод

адаптируется и обучается автоматически захватывать и использовать множество неопределенностей, содержащихся в исторических данных, RL широко используется в планировании умных сетей [21] – [24].

Например, в работе [21], адаптивный RL был использован для поиска равновесия Нэша в энергетической торговой игре с неполной информацией. Авторы в работе [22] предложили основанный на обучении с подкреплением алгоритм динамического ценообразования и планирования энергопотребления для энергосистемы. В работе [23] был реализован пакетный подход RL на основе профиля жилой нагрузки, чтобы составить план потребления на день вперед. В работе [24] авторы разработали управление питанием в режиме реального времени на основе RL для решения задачи распределения мощности для гибридной системы хранения энергии с внедрением электромобиля.

В последние годы некоторые алгоритмы RL, такие, как Q-обучение и глубокие Q-сети, показали хорошую производительность для управления ветряной станцией при отслеживании точек максимальной мощности [25], в локальных энергетических торговых стратегиях [26] и многих других проблемах с неопределенностями [27], [28]. Без указания точной модели и ее параметров алгоритм RL может определить стратегию управления, извлекая эффективные характеристики из данных. Однако в контексте задачи управления системой накопления энергией множественные параметры принятия решений, управляющие переменные, а также неизбежные неопределенности в данных приводят к многомерному и непрерывному пространству состояний и действий. Это приводит к медленной скорости сходимости вышеупомянутых алгоритмов RL, которые дискретизируют непрерывное пространство состояний и отбрасывают наблюдения после каждого обновления, что приводит к неэффективному использованию данных и влияет на результаты оптимизации. Чтобы решить эту проблему, в данной работе применяется глубокий детерминированный градиент политики (DDPG) [29] для управления мощностью заряда/разряда. Он не требует специальных статистических моделей и дискретизации непрерывных задач [30], а также не требует описания распределения неопределенностей в наборах данных, так как этот метод адаптируется и обучается автоматически за-

хватывать и использовать множество неопределенностей, содержащихся в исторических данных.

Глава 1. Обзор существующих решений

1.1 Энергоменеджмент в России

Нынешняя социально-экономическая ситуация в России обусловлена замедлением темпов роста в мировой экономике. Российские компании и потребители вынуждены снижать издержки хозяйственной деятельности, совершенствовать сферу энергосбережения для повышения энергоэффективности, а также искать альтернативу топливно-энергетическому обеспечению. На сегодняшний день в России уже накоплен некоторый практический опыт по разработке и внедрению интеллектуальных энергетических систем [31], [32], однако в настоящее время нет программной реализации, способной решить поставленную задачу планирования графика для хранения энергии в полном объеме. В основном текущие решения направлены на повышение качества энергии и стабилизацию энергетических потоков, что является важным для сбережения технических ресурсов оборудования, но недостаточным для увеличения энергоэффективности системы. Медленное развитие технологий и научных исследований и этой сфере объяснялось тем, только с 11 декабря 2019 года Государственная Дума Российской Федерации в третьем чтении приняла Федеральный закон «О внесении изменений в Федеральный закон «Об электроэнергетике»[33] в части развития микрогенерации» (проект № 581324-7), который вносит в Федеральный закон № 35-ФЗ «Об электроэнергетике» такое понятие, как «объект микрогенерации», тем самым упрощая возможность установки, подключения к общей сети и продажу электроэнергии частными лицами в сеть.

1.2 Индустриальные предложения

На рынке уже сейчас доступно несколько индустриальных предложений для решения задач энергоменеджмента с открытым исходным кодом. Однако одним из недостатков существующих решений является то, что их

может быть трудно внедрить в энергосистему и использовать пользователю без профильных знаний.

WattDepot [34] – это сервис-ориентированный фреймворк с открытым исходным кодом, реализованный на Java для систем энергетического менеджмента. Он обеспечивает сбор, хранение, анализ и визуализацию энергетических данных. Данное решение позволяет клиентам запрашивать данные с сервера и предоставляет мощный инструмент для их анализа.

Neurio Home Energy Monitor [35] – это открытая платформа, которая может использоваться для мониторинга потребления энергии и выходной мощности солнечных панелей. Одной из особенностей решения является, то что оно может информировать пользователя о том, какие устройства включены/выключены и генерировать рекомендации по потреблению энергоресурсов.

OpenHAB [36] – это программное обеспечение для автоматизации энергосистемы с открытым исходным исходным кодом, которое может работать на любом устройстве с JVM (Java virtual machine). Оно помогает интегрировать различные технологии домашней автоматизации и может быть использован для мониторинга и управления различными устройствами, а также предоставляет график для энергетической истории. Используя OpenHAB-designer, разработчики могут создавать свои пользовательские интерфейсы. При этом пользователю необходимо самостоятельно настроить систему под свою конфигурацию микрогрид.

Home Assistant [37] – это платформа домашней автоматизации с открытым исходным кодом, которая может контролировать и управлять устройствами на низком уровне. Он также может поддерживать автоматизацию работы устройств, основанную на предпочтениях пользователя.

Помимо открытых платформ для оптимизации энергосистем пользователей существуют разработки и проекты от крупнейших энергетических компаний, которые заинтересованы в имплементации эффективной управляющей компоненты для производимого ими оборудования. К примеру, компания Siemens уже сейчас предлагает программное обеспечение Spectrum Power [38], которое нацелено на обеспечение аварийной и предупредительной сигнализации, и регулирование частоты и мощности энер-

гетических потоков. Данное программное обеспечение в режиме реального времени предоставляет диагностику текущих процессов, что является важным аспектом для нормального функционирования энергосистемы. Платформа управления энергопотреблением от компании Schneider Electric – EcoStruxure [39] предоставляет решение по улучшению эксплуатации и оптимизации обслуживания сетей питания с распределенными источниками питания. И имеет функционал по оценке состояния энергосистемы с возможностью визуализации текущих процессов, а также применяется для распределения нагрузки оптимального потока мощности, анализа последствий аварийных ситуаций, прогнозирования сбоев и стабилизации напряжения в системе.

Коммерческие предложения, предоставляющие инструменты для управления энергетическими потоками и анализа состояния энергосистемы, имеют ряд недостатков, среди которых самыми весомыми являются отсутствие прозрачности и высокая стоимость программного обеспечения. Отсутствие прозрачности в первую очередь означает, что пользователь полностью изолирован от технологий, которые подключены к его оборудованию. Это может стать решающим фактором для внедрения подобных систем на предприятия государственного масштаба, из-за угрозы взлома и внедрения шпионских программ. А высокая стоимость программного обеспечения объясняется тем, что для использования коммерческих систем требуется лицензия и необходимы постоянные затраты на ее обновление.

Открытые решения также имеют недостатки, которые заключаются в определенных трудностях при внедрении системы и осуществлении технической поддержки. Подобное программное обеспечение трудно использовать для клиентов без специальной подготовки, а некоторые решения требуют установки и поддержки специалистами с обширными экспертными знаниями. К тому же большинство систем управления не являются интеллектуальными системами и осуществляют управление на низком уровне, что значительно сказывается на результатах оптимизации.

Для того чтобы решения энергоменеджмента в значительной степени использовались домовладельцами в эпоху интеллектуальных сетей, существует необходимость в системе управления энергией с открытым исходным

кодом, которую легко развернуть пользователю. Кроме того, для лучшей поддержки энергосистемы и снижения энергозатрат при удовлетворении уровня комфорта пользователя, в систему должны быть включены высокоэффективные интеллектуальные алгоритмы.

Глава 2. Моделирование

2.1 Модель энергосистемы

Рассматриваемая в данной работе энергосистема состоит из фотоэлектрической станции, аккумуляторной батареи в качестве накопителя энергии, жилой нагрузки, инверторов и трансформатора, соединяющего микрорешетку с местной коммунальной сетью. Инверторы преобразуют постоянный ток (DC) от батареи и фотоэлектрической системы в переменный ток (AC) для подачи в сеть пользователя. Жилую нагрузку можно удовлетворить, используя энергию от местной фотоэлектрической системы или покупая энергию у местной коммунальной сети. Избыточная энергия, произведенная при низком спросе на энергию или высоком производстве, может храниться в батарее и повторно использоваться во время пикового спроса, или продана в местную коммунальную сеть. В момент времени t система управления запрашивает необходимую информацию из базы данных: тарифы цена на электроэнергию, прогнозные значения выработки фотоэлектрической станции и нагрузки, а также характеристики оборудования. Встроенный алгоритм должен определять $SoC(t+1)$ — остаточную мощность батареи для следующего момента времени и передавать полученное значение контролеру. Затем управляющий элемент посылает команды различным системам для оптимального управления энергосистемой пользователя. Описанная архитектура микросетей показана на Рис. 2.

Параметры и переменные, используемые при моделировании, приведены в таблице 1. В данной работе планирование графика хранения энергии в интеллектуальных электрических сетях определяется на основе следующих допущений:

1. Не учитывается эффект старения или деградации батареи;

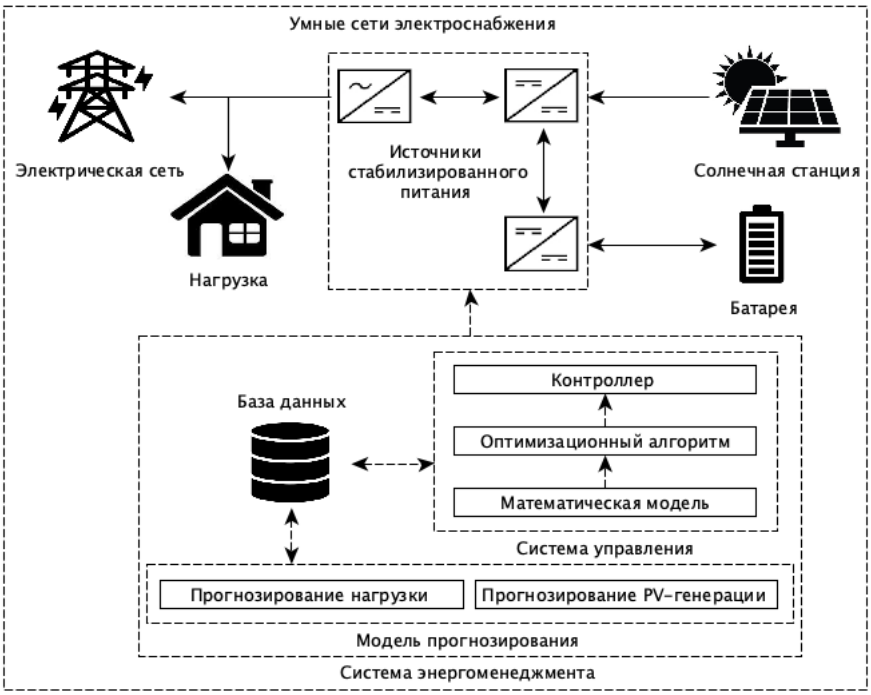


Рис. 2: Модель энергосистемы. Где пунктирная линия – поток данных, прямая – поток энергии

2. Окружающая среда не оказывает влияния на работу батареи;

3. Мощность заряда/разряда и вместимость батареи постоянна в течение заданного периода;

4. В любой момент времени батарея может выполнять только одно из действий: зарядка, разрядка, холостой ход;

5. Прогнозные значения выработки фотоэлектрической станции и потребности пользователей в энергии могут иметь погрешность;

6. Горизонт планирования – 24 часа, временной интервал – 15 минут. Первая точка времени для принятия решения за день устанавливается в 00:00.

Минимальные затраты на использование электроэнергии сети могут быть достигнуты путем покупки энергии в периоды низких цен и продажи ее в периоды высоких цен, а также накопление и последующее использование энергии в часы пика. Целевую функцию оптимизационной модели

Таблица 1: Параметры и переменные математической модели.

Обозначение	Описание
	Параметры
η_{ch}, η_{dch}	эффективность зарядки/разрядки аккумулятора
$P_{ch}^{max}, P_{dch}^{max}$	максимальная зарядная/разрядная мощность батареи
SoC_{min}, SoC_{max}	минимальное/максимальное состояние заряда батареи
$C_{buy}(t), C_{sell}(t)$	тариф на покупку/продажу энергии
$L(t)$	прогнозируемая жилая нагрузка
$P_{pv}(t)$	прогнозируемая мощность фотоэлектрической станции
SoC_{In}	начальное состояние заряда батареи
	Переменные
$P_{buy}(t), P_{sell}(t)$	приобретенная/проданная энергия
$SoC(t)$	остаточная мощность в батареи
$P_{ch}(t), P_{dch}(t)$	мощность зарядки/разрядки батареи
$u(t), \nu(t)$	1, если батарея разряжается, 0 иначе

можно сформулировать следующим образом:

$$F = \sum_{t=0}^{95} P_{buy}(t) \cdot C_{buy}(t) + \sum_{i=0}^{95} P_{sell}(t) \cdot C_{sell}(t) \rightarrow min,$$

Первая часть формулы – это общая стоимость за электроэнергию, импортируемую из электрической сети, вторая часть относится к продаже электроэнергии сетевой компании в течение заданных периодов времени. В оптимизационной модели для расчета допустимого решения функции затрат рассматриваются следующие ограничения. Ограничение энергетического баланса указывает на то, что возобновляемые источники энергии, батареи и электросети должны удовлетворять потребность сети в электроэнергии на каждом шаге, а именно:

$$P_{buy}(t) + P_{pv}(t) - P_{dch}(t) = L(t) + P_{ch}(t) - P_{sell}(t), \quad \forall t.$$

Инициализация начальной мощности батареи и соотношение мощности батареи между двумя последовательными шагами определяется по форму-

лам:

$$SoC(t) = SoC(t-1)+P_{ch}(t)\cdot\eta^{ch}+P_{dch}(t)/\eta^{dch}, \quad \forall t, t \neq 1, \quad SoC(0) = SoC_{In}.$$

В данном исследовании в качестве системы хранения выбраны литий-ионные батареи. Они являются одним из самых популярных типов систем хранения и все чаще используются в энергосистемах из-за более высокого соотношения вместимости энергии, медленной потери заряда в режиме холостого хода и отсутствия эффекта памяти. При моделировании работы аккумуляторной системы должны быть удовлетворены ограничения на скорость зарядки, скорость разрядки и вместимость батареи:

$$0 \leqslant P_{ch}(t) \leqslant u(t) \cdot P_{ch}^{max}, \quad \nu(t) \cdot P_{dch}^{max} \leqslant P_{dch}(t) \leqslant 0, \quad \forall t.$$

$$u(t) + \nu(t) \leqslant 1, \quad \forall t.$$

$$SoC_{min} \leqslant SoC(t) \leqslant SoC_{max}, \quad \forall t.$$

Однако, из-за того, что подаваемые на вход модели прогнозные значения жилой нагрузки $L(t)$ и мощности выработки фотоэлектрической станции $P_{pv}(t)$ имеют ошибку прогнозирования, стандартные методы решения MILP могут излишне точно аппроксимировать неточные данные, что в итоге влияет на результаты оптимизации энергосистемы пользователя. Поэтому необходимо определить методы, которые смогут адаптироваться и обучается автоматически захватывать и использовать множество неопределенностей, содержащихся в исторических данных.

2.2 Симуляция

Механизм моделирования передает данные контроллеру батареи на каждом временном шаге и запрашивает SoC для следующего временного шага. После установления SoC батареи рассчитывается энергия, необходимая для удовлетворения энергетического баланса – Δ. Если полученное

Исходные параметры: $\eta_{ch}, \eta_{dch}, P_{ch}^{max}, P_{dch}^{max}, SoC_{min}, SoC_{max}$
Результат: $\sum$
пока $episode = \overline{1, M}$ **выполнять**
  Входные данные: SoC_{In}
  $SoC(0) = SoC_{In}$ **пока** $t = \overline{1, T}$ **выполнять**
    Входные данные: $SoC(t)$,$C_{buy}^{*}(t), C_{sell}^{*}(t), L^{*}(t), P_{pv}^{*}(t)$
    если $SoC(t) > SoC_{max}$ **тогда**
      $SoC(t) \leftarrow SoC_{max}$
      $P_{ch}(t) \leftarrow min(SoC(t) - SoC(t-1))/\eta_{ch}, P_{ch}^{max})$,
      $\mathrm{P}_{dch}(t) \leftarrow 0$, $SoC(t) \leftarrow SoC(t-1) + P_{ch}(t) * \eta_{ch}$
    иначе если $SoC(t) < SoC_{min}$ **тогда**
      $SoC(t) \leftarrow SoC_{min}$,
      $P_{dch}(t) \leftarrow max((\mathrm{SoC(t)} - \mathrm{SoC(t\text{-}1)})\eta_{dch}, P_{dch}^{max})$,
      $\mathrm{P}_{ch}(t) \leftarrow 0$,$\mathrm{SoC(t)} \leftarrow SoC(t-1) + P_{dch}(t)/\eta_{dch}$
    конец
    если $SoC(t) - SoC(t-1) < 0$ **тогда**
      $\mathrm{P}_{dch}(t) \leftarrow max((\mathrm{SoC(t)} - \mathrm{SoC(t\text{-}1)})\eta_{dch}, P_{dch}^{max})$,
      $\mathrm{P}_{ch}(t) \leftarrow 0$, $SoC(t) \leftarrow SoC(t-1) + P_{dch}(t)/\eta_{dch}$
    иначе
      $SoC(t) \leftarrow SoC_{max}$
      $P_{ch}(t) \leftarrow min(SoC(t) - SoC(t-1))/\eta_{ch}, P_{ch}^{max})$,
      $\mathrm{P}_{dch}(t) \leftarrow 0$, $SoC(t) \leftarrow SoC(t-1) + P_{ch}(t) * \eta_{ch}$
    конец
    если $\Delta = P_{pv}^{*}(t) - L^{*}(t) + P_{dch}(t) + P_{ch}(t) > 0$ **тогда**
      $P_{sell}(t) \leftarrow -\Delta, P_{buy}(t) \leftarrow 0$,
      $\sum \leftarrow \sum + P_{sell}(t)C_{sell}(t), t \leftarrow t + 1$
    иначе
      $P_{buy}(t) \leftarrow \Delta, P_{sell}(t) \leftarrow 0$,
      $\sum \leftarrow \sum + P_{buy}(t)C_{sell}(t), t \leftarrow t + 1$
    конец
  конец
конец

Алгоритм 1: Механизм симмуляции с накомительной батареей

значение отрицательно, энергия будет закуплена из сети в необходимом объеме, если положительно, то это количество энергии продается в сеть. Затем рассчитывается стоимость покупки/продажи энергии и добавляется к текущей сумме – $\sum$. Подробный механизм симуляции представлен в алгоритме 1, где M – количество дней в тестовом наборе. Для получения результатов оптимизации также рассчитывалось значение затрат пользо-

вателя на покупку энергии без использования накопительных батарей. В этом случае нагрузка была сбалансирована с помощью электроэнергии центральной сети и произведенной фотоэлектрической станцией. Для моделирования применялся алгоритм 2.

Результат: $\sum$
пока $episode = \overline{1, M}$ **выполнять**
  пока $t = \overline{1, T}$ **выполнять**
    Входные данные: $C^*_{buy}(t), C^*_{sell}(t), L^*(t), P^*_{pv}(t)$
    если $\Delta = P^*_{pv}(t) - L^*(t) > 0$ **тогда**
      $P_{sell}(t) \leftarrow -\Delta, P_{buy}(t) \leftarrow 0,$
      $\sum \leftarrow \sum + P_{sell}(t) * C_{sell}(t), t \leftarrow t + 1$
    иначе
      $P_{buy}(t) \leftarrow \Delta, P_{sell}(t) \leftarrow 0,$
      $\sum \leftarrow \sum + P_{buy}(t) * C_{sell}(t), t \leftarrow t + 1$
    конец
  конец
конец

Алгоритм 2: Механизм симмуляции без накомительной батареей

2.3 Источник данных

Исследование проводилось на основе данных предоставленных компанией Schneider Electric [1] – энергомашиностроительная компания, производитель оборудования и программного обеспечения для управлении энергией и автоматизации процессов. Компания при поддержке портала Driven data [40] опубликовала в открытом доступе данные, которые были предложены для решения задач энергоменеджмента. Данные включают в себя 3 файла CSV формата: Metadata (4 КБ) – содержит таблицу с характеристиками оборудования, Training data (2.2 ГБ) – набор данных для обучения, Submit data (546.9 МБ) – набор данных для тестирования. Размер обучающей выборки – 460800 наблюдений, размер тестовой выборки – 115200 наблюдений.

2.4 Обзор данных

Наборы данных содержат 11 тестовых случаев, каждый из которых включает в себя несколько периодов моделирования, длительность одного периода составляет не более 10 дней, период характеризуется профилем потребления и выработки энергии. Для каждого тестового случая заданы характеристики используемого оборудования. Данные оборудования содержат следующую информацию: эффективность зарядки и разрядки, максимальная мощность и максимальное состояние заряда батареи. Для того чтобы сравнить как характеристики оборудования влияют на результаты оптимизации, предлагается проводить эксперименты для каждого тестового случая на батареях с различными значениями вместимости и максимальной/минимальной мощности (Battery 1 – таблица 2 и Battery 2 – таблица 3).

Таблица 2: Характеристики оборудования (Battery 1)

Test case	SOC_{max}	SOC_{min}	P_{ch}^{max}	P_{dch}^{max}	η^{dch}	η^{ch}
1	300	0	75	-75	0.950	0.950
2	600	0	150	-150	0.950	0.950
3	100	0	25	-25	0.950	0.950
4	100	0	25	-25	0.950	0.950
5	10	0	2.5	-2.5	0.950	0.950
6	1500	0	375	-375	0.950	0.950
7	400	0	100	-100	0.950	0.950
8	10	0	2.5	-2.5	0.950	0.950
9	100	0	25	-25	0.950	0.950
10	300	0	75	-75	0.950	0.950
11	600	0	150	-150	0.950	0.950

Для каждого момента времени из периода приведены фактические значения энергопотребления и выходной мощности фотоэлектрической станции на предыдущем шаге (Рис. 3) и актуальные тарифы на покупку и продажу энергии (Рис. 4). Также предлагается многомерный временной ряд, состоящий из прогнозных значений энергопотребления, выходной мощности фотоэлектрической станции на следующие сутки с шагом в 15 минут.

Таблица 3: Характеристики оборудования (Battery 2)

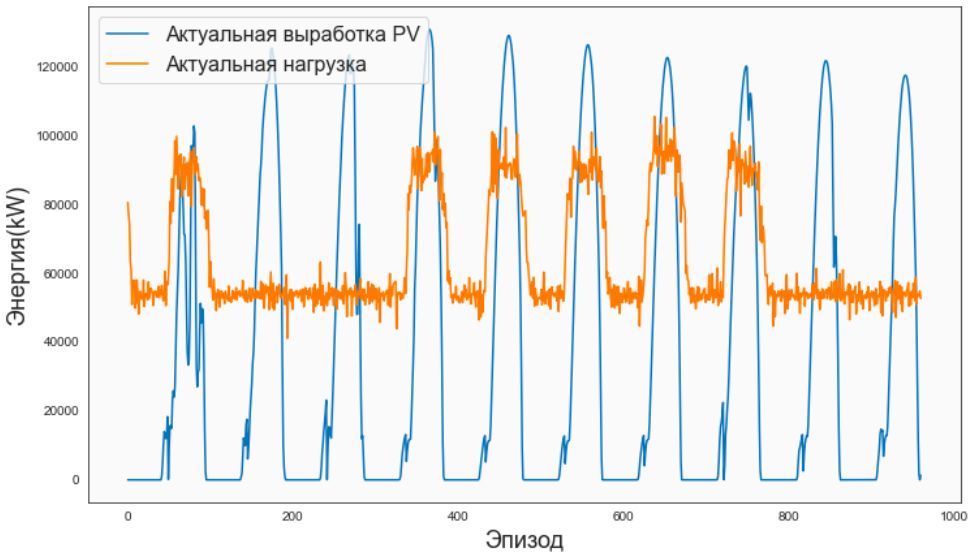
Test case	SOC_{max}	SOC_{min}	P_{ch}^{max}	P_{dch}^{max}	η^{dch}	η^{ch}
1	600	0	150	-150	0.950	0.950
2	1200	0	300	-300	0.950	0.950
3	200	0	50	-50	0.950	0.950
4	200	0	50	-50	0.950	0.950
5	20	0	5	-5	0.950	0.950
6	3000	0	750	-750	0.950	0.950
7	800	0	200	-300	0.950	0.950
8	20	0	5	-5	0.950	0.950
9	200	0	50	-50	0.950	0.950
10	600	0	150	-150	0.950	0.950
11	1200	0	300	-300	0.950	0.950

Визуализация актуальных значений для нагрузки и выработки фотоэлектрической станции по сравнению с прогнозируемыми приведено на Рис.5. Понимание и анализ входных данных имеет решающее значение при разработке хорошей модели и является ориентиром при выборе алгоритма.

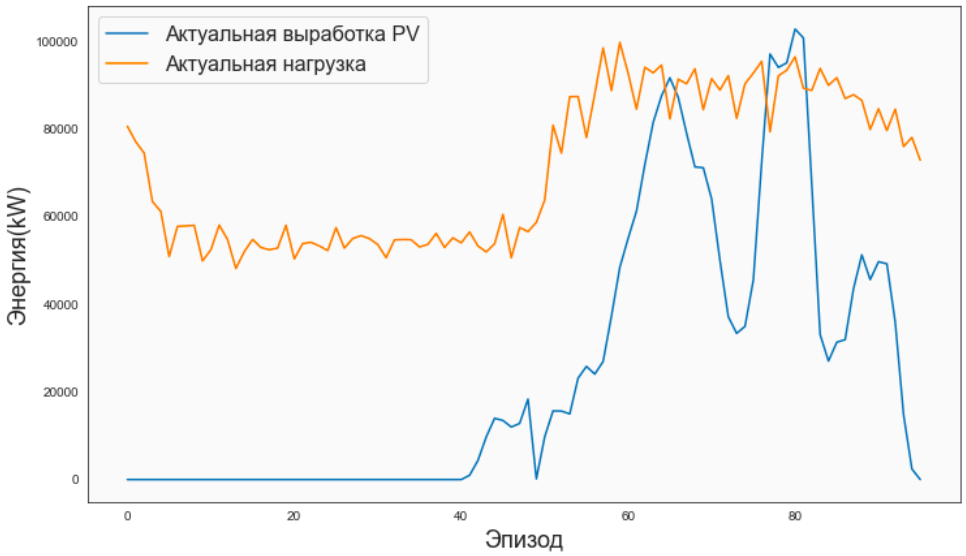
Имеющиеся прогнозные данные энергопотребления и выработки фотоэлектрической станции содержат ошибку прогнозирования. Поэтому необходимо оценить точность прогнозирования, а именно ввести показатель, который характеризует насколько полученные данные соответствуют истинными фактическим значениям. При расчете точности прогнозирования применяются несколько метрик: средняя абсолютная процентная ошибка, средняя процентная ошибка, медиана абсолютной процентной ошибки, средняя абсолютная масштабированная ошибка, взвешенная абсолютная процентная ошибка. Наименее чувствительная к выбросам и искажениям, а так же легко интерпретируемая метрика – взвешенная абсолютная процентная ошибка:

$$WAPE(y, \widehat{y}) = \frac{\sum_{i=1}^{T} |y_i - \widehat{y}_i|}{\sum_{i=1}^{T} y_i},$$

где $y, \widehat{y}$ — фактическое и предсказанное значение энергопотребления



(а) За период в 10 дней



(б) За период в 1 день

Рис. 3: Примеры профиля энергопотребления и выработки фотоэлектрической станции

или выработки фотоэлектрической станции соответственно, T – количество шагов для управления в тестовом случае.

Значения метрики для прогнозных значений энергопотребления и выработки фотоэлектрической станции приведены в таблице 4. Прогнозирование нагрузки имеет большую взвешенную абсолютную процентную ошибку, однако прогнозные значения выработки фотоэлектрической станции больше склонны к завышению относительно актуальных значений. При решении задачи управления энергосистемой стандартными методами

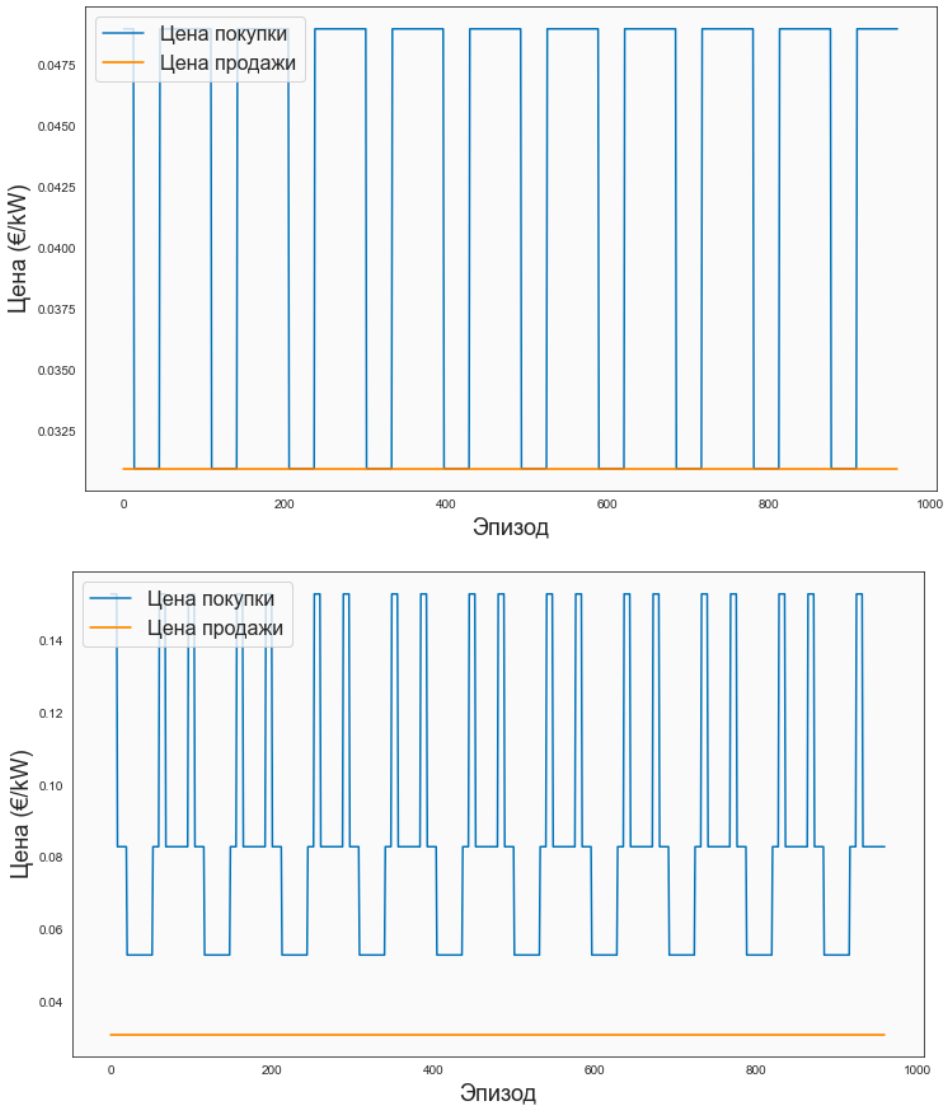
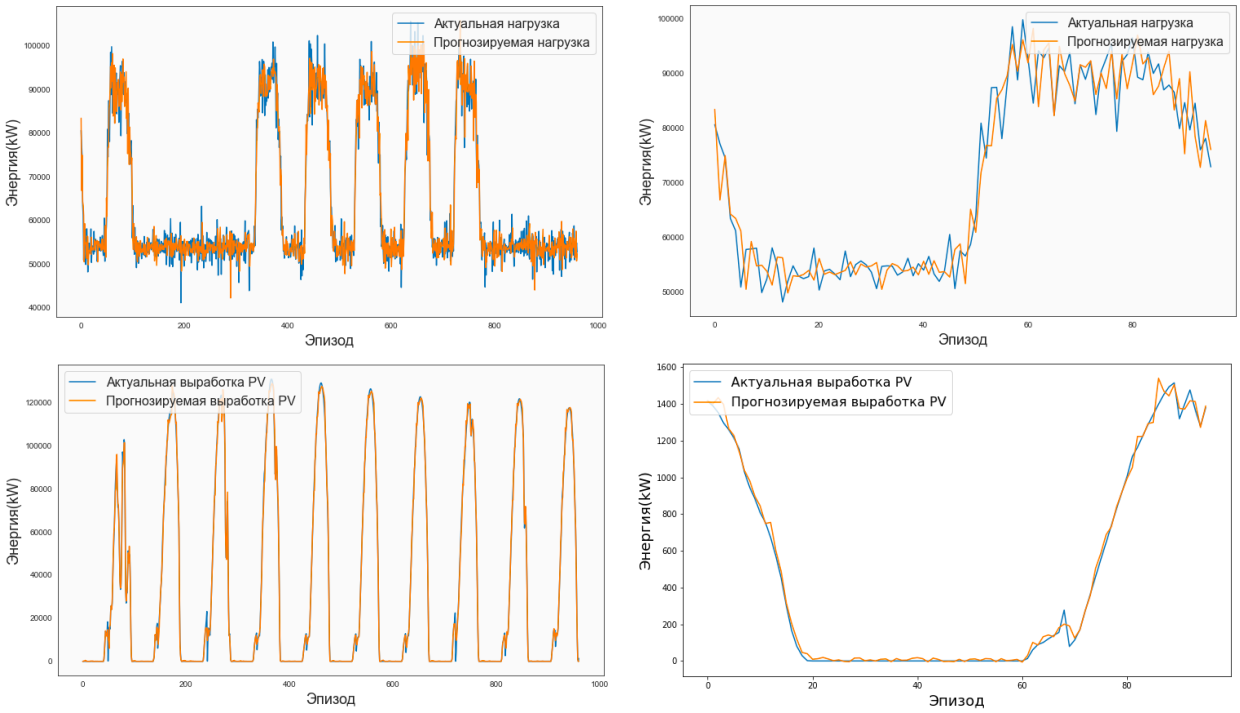


Рис. 4: Примеры тарифных планов на покупку и продажу энергии за 10 дней

MILP даже небольшой процент ошибки на каждом из наблюдений приводит кумулятивному эффекту накопления ошибок, а учитывая, что 192 входных параметра из 391 являются прогнозируемыми, это может привести к значительному отклонению полученных решений от оптимальных. Наличие ошибки в данных затрудняет построение модели и ведет к ухудшению результатов оптимизации, поэтому необходим алгоритм, который будет способен учитывать существующую неопределенность при решении задачи.

Таблица 4: Значение метрики $WAPE$ для прогнозных значений энергопотребления – L и выработки фотоэлектрической станции – PV

$WAPE(\%)$	1	2	3	4	5	6	7	8	9	10	11
L	4.12	5.12	5.5	9.05	15.31	3.34	15.34	15.48	11.78	6.64	5.81
PV	3.64	5.74	7.23	2.84	2.87	5.70	2.78	7.16	3.19	4.61	3.18



(а) За период в 10 дней (б) За период в 1 день

Рис. 5: Примеры актуальных и прогнозируемыех значений нагрузки и выработки фотоэлектрическоий станции

Глава 3. Управление энергосистемой как задача обучения с подкреплением

В данной работе рассматриваются методы обучения с подкреплением, как альтернатива детерминированным подходам, для решения последовательной задачи принятия решений, включающей оперативное планирование работы накопительного элемента. В зависимости от нагрузки, состояния заряда батареи (SoC), эффективности оборудования и электроэнергии, вырабатываемой фотоэлектрическим преобразователем, агент определяет оптимальный режим работы батареи: оставаться в режиме ожидания, заряжать или разряжать батарею. Цель агента – минимизировать затраты

на электроэнергию и, следовательно, максимизировать собственное потребление электроэнергии местного производства. Чтобы решить эту проблему, сначала будет введено определение задачи оптимального управления и сформулирована вышеприведенная проблема как марковский процесс принятия решений, а затем предложен алгоритм управления энергией, основанный на глубоком Q-обучении(DQN) и глубоких детерминированных градиентах политики(DDPG). В этом разделе также будет приведен краткий обзор обучения с подкреплением, формализм марковских решений и глубоких нейронных сетей.

3.1 Понятие оптимального управления

Задача оптимального управления состоит в том, чтобы определить систему управления или закон управления, который осуществлял бы последовательность действий над объектом контроля таким образом, чтобы достичь максимума/минимума некоторой совокупности критериев качества системы. Объект управления, или управляемый объект – это некоторая часть системы, на которую целенаправленно оказывает воздействие субъект управления Рис. 6. Пусть управление происходит в течение периода заданного промежутком $[t_0, t_T]$, тогда состояние управляемого объекта в любой момент времени $t \in [t_0, t_T]$ характеризуется вектором состояния, которые могут изменяться с течением времени: $S(t) = s_1(t), s_2(t), ..., s_d(t)$ и находится под воздействием управляемых переменных – вектора управления $A(t) = a_1(t), a_2(t), ..., a_m(t)$.

Рис. 6: Схема взаимодействия объекта, субъекта и среды

В задачах оптимального управления критерием оптимальности выступает функционал для которого необходимо определить неизвестную функ-

цию управления $A(t)$, доставляющую минимум интегралу J при условии, что управление $A(t)$ выбирается из множества допустимых управлений A. Формулировка задачи оптимального управления в общем виде может быть записана в следующем виде:

$$J = \int_{t_0}^{t_T} F(t, S(t), A(t))dt \to \min_{A(t) \in A}$$

где $F(t, S(t), A(t))$ – функция $d+m+1$ переменных, целевая функция оптимизационной задачи, задающая выгоду в зависимости от параметров управления и состояния окружающей среды (к примеру, тариф на покупку/продажу электроэнергии в заданный момент времени). $A(t)$ определяют какие значения в заданный момент времени должны принимать параметры управления (запасать энергию в батарею, выгружать в сеть или бездействовать).

Однако, модели электроэнергетических систем могут быть сложны, и поэтому в явном аналитическом виде функцию $F(t, S(t), A(t))$ и ее интеграл задать нельзя, но зачастую ее можно определить алгоритмически. Целевая функция в поставленной задаче является кусочно-непрерывной и определяется как разница между доходами от продажи электроэнергии генерирующего потребителя и расходами на ее покупку на сутки вперед с шагом в 15 минут. В этом случае аналитическое выражение для $F(t, S(t), A(t))$ записать затруднительно, так как цена на электроэнергию является кусочно - постоянной функцией, а обмен электроэнергией с сетевым предприятием зависит от состояний среды, которые определяются факторами содержащими неопределенность прогнозных значений. Таким образом, расчет значения $F(t, S(t), A(t))$ необходимо выполнять алгоритмически.

3.2 Обучение с подкреплением. Основные понятия и принципы

В контексте обучения с подкреплением задача обучения агента в заданной среде может быть сформулирована в виде частично наблюдаемого

марковского процесса принятия решений (МППР). Процесс обладает свойством Маркова, если все последующие изменения в процессе возможно описать только по последнему этапу процесса. Если пространство состояний и действий конечны, то задача называется конечным марковским процессом принятия решений. МППР определяются кортежем $<\mathcal{S}, \mathcal{A}, \mathcal{T}(\cdot,\cdot), \mathcal{R}(\cdot,\cdot)>$, где:

- $\mathcal{S}$ – пространство состояний, $\forall$s$\in \mathcal{S}$
- $\mathcal{A}$ – это пространство действий, $\forall$a$\in \mathcal{A}$
- $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0,1]$ – функция перехода, заданная вероятностью того, что, выбрав действие a в состоянии s в момент времени t, система придет в состояние s' в момент времени $t+1$ такое, что $p_a(s,s') = p(s_{t+1} = s' | s_t = s, a_t = a)$
- $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$. Это функция вознаграждения, где $r_t = R_a(s,s')$ – немедленное вознаграждение, полученное агентом в момент времени t после выполнения перехода в состояние s' из состояния s.

Марковский процесс принятия решений является основным формализмом обучения с подкреплением. Обучаемая модель, называемая агентом, взаимодействует с окружающей средой и по реакции среды на свои действия формирует некоторое поведение с целью максимизировать выгоду от взаимодействия со средой или минимизировать потери. Рассматривается агент, помещенный во внешнюю среду, и имеющий конечный набор действий, каждое из которых приводит к изменению внешней среды и получению от нее обратной связи. Причем обратная связь может быть позитивной (поощрение) или негативной (наказание). В процессе взаимодействия с внешней средой агент меняет свое поведения, чтобы максимизировать поощрение и минимизировать наказание, то есть обучается, чтобы максимизировать долгосрочное накопленное вознаграждение: $R_t = r_t + \gamma(r_{t+1} + \gamma(r_{t+2})) = r_t + \gamma R_{t+1}$, где $\gamma \in [0,1]$ - это коэффициент дисконтирования, используемый для оценки эффекта будущего вознаграждения, сохраняя при этом кумулятивное вознаграждение. Цель агента – найти

оптимальный план, или стратегию, π которая максимизирует ожидаемую полезность действий из любого состояния s_t, определяемую как математическое ожидание суммы дисконтированных будущих наград:

$$R_t = E_\pi[\sum_{i=0}^{T-t} \gamma^i r_{t+i} | s_t],$$

где последовательность будущих состояний $(s_{t+1}, s_{t+2}, ..., s_T)$ индуцируется функцией перехода $\mathcal{T}$ и стратегией π.

Таким образом в любой момент времени t агент наблюдает текущее состояние $s \in \mathcal{S}$ среды и выбирает действие $a \in \mathcal{A}$. Система вероятностно эволюционирует в следующее состояние $s' \in \mathcal{S}$ в соответствии с функцией перехода $\mathcal{T}(s, a, s')$. Затем агент получает вознаграждение $r_t = R_a(s, s')$ за выполненный переход. Цель агента – определить политику $\pi : \mathcal{S} \to \mathcal{A}$, которая стремиться максимизировать функцию вознаграждения в процессе обучения. Как правило, для измерения качества политики π вводят функцию ценности $Q^\pi(s, a)$, которая обозначает наибольший возможный счет, которого можно достичь в конце игры после выбора действия a в состоянии s. Учитывая политику π, функция ценности определяется следующим образом:

$$Q^\pi(s, a) = E_\pi[R_t | s_t = s, a_t = a],$$

где E_π – математическое ожидание в соответствии с политикой π, s_t и a_t – состояние среды и действие агента в момент времени t.

Поскольку оптимальная политика π^* – это политика, которая максимизирует функцию значения, оптимальная политика каждой функции ценности может быть получена как $\pi^* = \underset{a \in A}{argmax} Q^\pi(s, a)$.

Для решения задач обучения с подкреплением предложены методы, которые представлены на Рис.7. В зависимости от того, известна ли функция перехода $\mathcal{T}$ или нет, методы RL можно разделить на model free и model based подходы. В методах, основанные на моделях, агент изучает среду, представляет модель среды и составляет решение основанное на этой модели, в то время как в подходах, не основанные на моделях, агент изучает

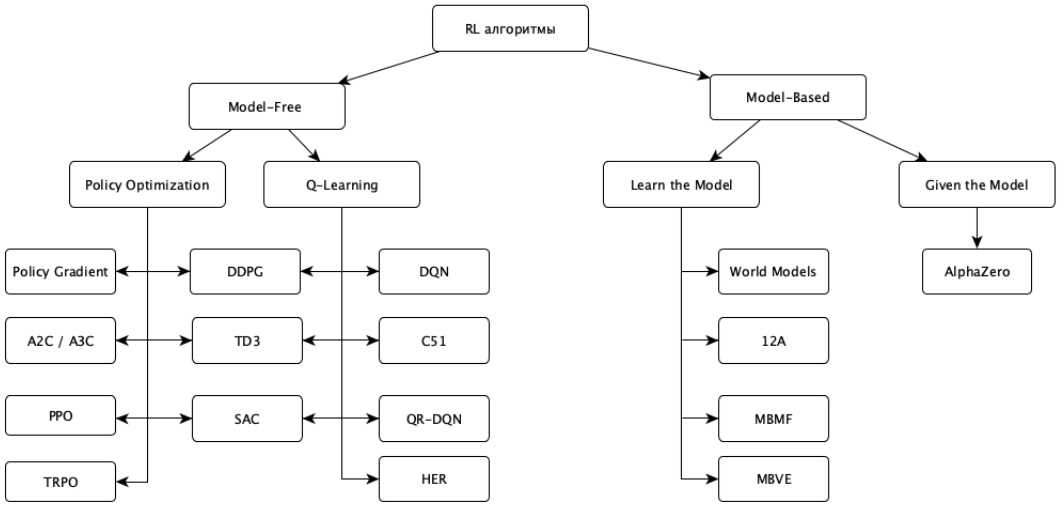


Рис. 7: Алгоритмы решения задач обучения с подкреплением

среду и определяет политику действия по сигналу награды. RL на основе модели может быть решена с помощью динамического программирования, в то время как RL без модели может быть решена методами Монте-Карло или временной разности. Методы временной разности объединяют подходы динамического программирования и идеи метода Монте-Карло, и являются одной из самых популярных схем RL. Среди методов временной разности наиболее классическим является Q-обучение, предложенное К. Уоткинсом в его докторской диссертации в 1989 году [41], где он впервые определил правило итеративного обновления для оценки функции $Q(s, a)$, которая подчиняется уравнению Беллмана:

$$Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \alpha(r_{t+1} + \gamma(\cdot \underset{a' \in A}{max}(Q_t(s_{t+1}, a')) - Q_t(s_t, a_t)),$$

где α – скорость обучения, которая определяет степень значимости новых наблюдений при обновлении оценки. К примеру, при $\alpha = 1$ предыдущие оценки не учитываются. Алгоритм 3 представляет основную идею Q-обучения.

Исходные параметры: $Q(s,a)$, ϵ,α, γ

Результат: $Q(s,a)$

для каждого $t = \overline{1,T}$ **выполнять**

  для каждого $t = \overline{1,T}$ **выполнять**

    выбрать действие a_t в состоянии s_t

    с помощью ϵ-жадной политики;

    получить награду r_t и новое состояние s_{t+1};

    обновить функцию полезности

    $Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha(r_t + \gamma(\underset{a' \in A}{max}(Q(s_{t+1},a')) - Q(s_t,a_t))$;

  конец

конец

Алгоритм 3: Q-Learning.

Метод Q-обучение, представленный выше, является алгоритмом, основанным на функциях ценности. Для решения задачи сначала необходимо определить Q функцию, а затем улучшить текущую политику π на основе полученной функции. Другой подход – методы, основанные на политике, такие как градиентная оптимизация стратегии (policy gradient) [42]. Алгоритмы оценки политики – это методы аппроксимирующие политику напрямую. Основой для всех модификаций алгоритмов этого типа является простой policy gradient алгоритм (Reinforce), особенность этого алгоритма в том, что в процессе корректировки весов модели он учитывает всю траекторию агента (последовательность переходов – кортежей $\mathcal{S}_t, \mathcal{A}_t, \mathcal{S}_{t+1}, \mathcal{R}_t$) на протяжении периода моделирования. Предположим, что политика, которую необходимо аппроксимировать с помощью параметра θ равна $\pi(s,a;\theta)$. Тогда целевая функция может быть определена как:

$$J(\pi_\theta) = E[\sum_{t=1}^{T-1} \gamma^{t-1} r_t | s_0 = s, \pi_\theta]$$

Нам нужно выбрать такой набор параметров агента θ задающий $\pi_\theta(a|s)$, чтобы максимизировать математическое ожидание суммы полученных вы-

игрышей, для этого рассчитывается градиент функции:

$$\nabla_\theta J(\pi_\theta) = E[(\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t))R_t] \approx \quad (1)$$

$$\approx \frac{1}{M} \sum_{i=1}^{M} (\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(a_t^i|s_t^i))R_{t^i}, \quad (2)$$

где a_t^i, s_t^i – значения действия агента и состояния среды в момент времени t i-го сценария $(s_1^i, a_1^i, ...s_{T^i}^i, a_{T^i}^i)$ из M сгенерированных. Для того, чтобы получить несмещенную выборку сценариев из вероятностного распределения, необходимо зафиксировать произвольно параметр θ и провести тестовые запуски. $R_{t^i} = \sum_{t=1}^{T^i} r(s_t^i, a_t^i)$ – сумма всех выигрышей, полученных в ходе сценария i.

Хотя классические методы RL достигли больших успехов во многих областях, в настоящее время их все труднее применять для решения индустриальных проблем. Так как они имеют ряд недостатков: дискретизация пространства, проклятия размерности, неэффективная обработка данных. Поэтому исследователи пытаются преодолеть эти недостатки рассматривая проблему разных точек зрения, среди которых аппроксимация функции ценности и политики является наиболее вероятным решением. В последние годы глубокое обучение (DL) все чаще используется для аппроксимации произвольных сложных функций. Глубокие нейронные сети (DNN), составленные из нескольких скрытых слоев, позволяют строить абстракции в виде иерархии признаков и благодаря отличным свойствам адаптивности, расширенному отображению ввода-вывода и нелинейности, глубокое обучение часто используется для аппроксимации функций в численных алгоритмах. В данной работе будут рассмотрены одни из наиболее популярных алгоритмов глубокого обучения с подкреплением (DRL): Глубокое Q-обучение (Deep Q-Learning (DQN)) и глубокие детерминированные

градиенты политики (Deep Deterministic Policy Gradient (DDPG)):

$$DRL = \begin{cases} \xrightarrow[states]{Input} DNN \xrightarrow[Q_\theta(s)]{Output} & \text{DQN} \\ \xrightarrow[states]{Input} DNN \xrightarrow[\pi_\theta(s)]{Output)} & \text{DDPG} \end{cases}$$

3.3 Глубокие нейронные сети

Глубокая нейронная сеть (DNN) – это искусственная нейронная сеть (ANN), которая содержит несколько слоев между входным и выходным слоями, что позволяет найти метод математических многомерных преобразований, подходящий для того, чтобы преобразовать подаваемые данные, в выходные переменные, независимо от линейной или нелинейной корреляции. Важным преимуществом искусственных нейронных сетей является способность анализа сложных зависимостей и отношений и причем в режиме автономного (автоматического) обучения или даже самообучения. Существуют различные архитектуры глубоких нейронных сетей, однако ключевыми компонентами являются следующие понятия: нейрон, вес, смещение и функция активации. Эти компоненты функционируют аналогично человеческому мозгу и могут быть обучены на решение определенных задач.

Отображение пространства входных данных в пространство выходных переменных можно осуществлять, соглавно теореме А.Н. Колмогорова [43]: каждая многомерная неприрывная функция (многомерное отображение XßY) может быть вычислена путем представления функции многих переменных в виде суперпозиции функций меньшего числа переменных. Р. Хехт-Нильсен [44] впоследствии предложил основную теорему для нейронных сетей. Из нее следует представимость любой многомерной функции нескольких переменных с помощью нейронной сети заданной размерности с нейронами скрытого слоя и функциями активации нейронов. Функциональным элементом любой ANN является нейрон Рис.8, который можно

представить как функцию следующего вида:

$$y = f(\sum_{i=1}^{n} w_i x_i + b),$$

где x_i – входные сигналы, это данные, которые поступающие из окружающей среды или от других нейронов, n – количество входов нейрона, w_i – вещественные весовые коэффициенты входа, которые определяют силу связи между нейронами, b – смещение, f – нелинейная функция активации, предназначена для вычисления выходного значения сигнала, передаваемого другим нейронам, y – выход нейрона. Определено множество различных функций активации. Наиболее эффективные и часто используемые функции это гиперболический тангенс $f(x) = \frac{2}{1+e^{-2x}} - 1$, сигмоидальная функция $f(x) = \frac{1}{1+e^{-x}}$, линейный выпрямитель $f(x) = max(0; x)$

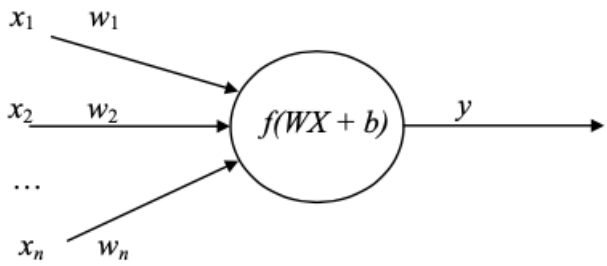


Рис. 8: Графическое изображение нейрона

При соединение нейронов в сеть создается архитектура нейронной сети, она может быть инициализирована различными способами. В настоящее время при построении моделей со сложными зависимостями выделяют такие архитектуры как: многослойный перцептрон, сверточная сеть, рекуррентная сеть и т.д. В данной работе для аппроксимации функций будет использована архитектура многослойный перцептрон, который представляет собой ряд последовательных слоев, каждый слой – это набор из некоторого количества нейронов, не имеющих связей друг с другом Рис.9.

Архитектура нейронной сети задается до начала обучения, параметры – веса w и смещения нейронов b определяются в процессе обучения. Существует множество методов обучения/тренировки ANN: метод обратного распространения ошибки, дельта-правило, метод коррекции ошибки.

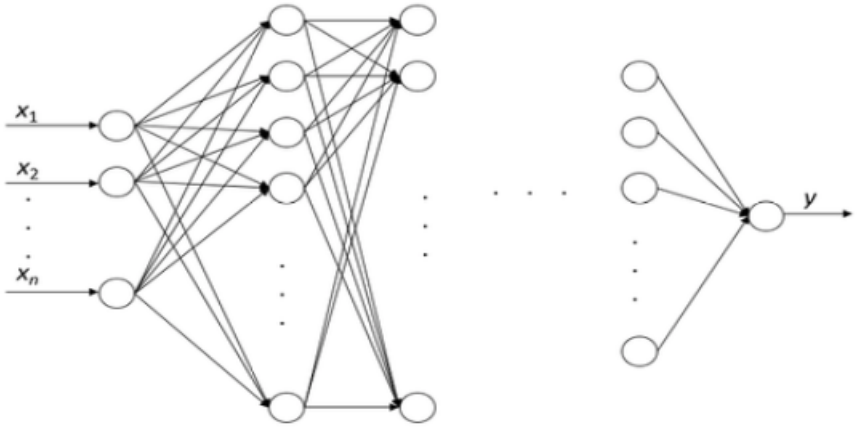


Рис. 9: Многослойный перцептрон

Наиболее распространенным является метод обратного распространения ошибки (Backpropagation). Основная идея метода обратного распространения ошибки представлена в определении отклонения (ошибки или потери – loss) выходного значения DNN для текущего обучающего примера от желаемого значения и передачи этого значения от выхода DNN через все ее слои к входу с коррекцией параметров DNN для снижения величины этого отклонения. Коррекция параметров DNN выполняется по алгоритму градиентного спуска. Условием прекращения работы алгоритма являются следующие критерии: достигается выполнение определенного количества итераций, либо по методу ранней остановки. Метод ранней остановки заключается в том, что процесс обучения останавливается, если за заданное количество эпох потери не начинают значительно уменьшаться или результат становится хуже.

3.4 Управление накоплением энергии

В этом разделе будет представлена система управления накоплением энергии на основе обучения с подкреплением.

Пространство состояний ($\mathcal{S}$). Пространство состояний $\mathcal{S}$ состоит из временной составляющей – S_t, неконтролируемой экзогенной составляющей – S_x и управляемой части – S_c:

$$\mathcal{S} = S_t \times S_x \times S_c.$$

Компонента времени S_t зависит от даты и времени и содержит информацию о состоянии энергосистемы, относящуюся к периоду времени. Используя эту информацию, агент может захватить некоторую информацию о динамике системы, релевантную для процесса обучения. Функция синхронизации определяется следующим образом:

$$S_t = S_t^d \times S_t^q,$$

где $S_t^q \in R$ – представляет четверть часа дня, а $S_t^d \in R$ – день недели. Компонента времени позволяет обучающемуся агенту получать такую информацию, как структура потребления бытовых потребителей и профиль производства фотоэлектрических приборов. Большинство бытовых потребителей и фотоэлектрических систем, как правило, следуют повторяющейся схеме ежедневного потребления и производства соответственно.

Управляемые компоненты содержат информацию о состоянии среды, которое относятся к управляемым величинам, или на которые влияют управляющие воздействия. В этом случае батарея является управляемым компонентом, которая характеризуется остаточной мощностью – $S_c \in R$.

Экзогенный признак S_x содержит наблюдаемую экзогенную информацию, которая оказывает влияние на динамику системы и функцию затрат, но не может быть подвержена влиянию управляющих воздействий. Данная работа предполагает наличие прогнозной информации на сутки вперед с шагом в 15 минут об экзогенном состоянии системы:

$$S_x = S_x^l \times S_x^{pv} \times S_x^b \times S_x^s \times S_x^{l_0} \times S_x^{pv_0} \times S_x^{b_0} \times S_x^{s_0},$$

где S_x^l вектор прогнозных значений жилой нагрузки, а S_x^{pv} – вектор прогнозных значений энергии генерируемой фотоэлектрической станцией, S_x^b и S_x^s – векторы прогнозных значений тарифов на покупку и продажу энергии соответственно. $S_x^l, S_x^{pv}, S_x^b, S_x^s \in R^{96}$, $S_x^{l_0}, S_x^{pv_0}, S_x^{b_0}, S_x^{s_0} \in R$ – актуальные значения жилой нагрузки, энергии фотоэлектрической станции, тарифов на покупку и продажу энергии на предыдущем временном шаге. Таким

образом, состояние системы энергоснабжения определяется как:

$$S = (S_t^d, S_t^q, SoC, S^l, S^{pv}, S_x^b, S_x^s, S_x^{l_0}, S_x^{pv_0}, S_x^{b_0}, S_x^{s_0}) \in \mathcal{S}.$$

Для повышения производительности предлагается нормализовать пространство состояний таким образом, чтобы все значения находились в одинаковом масштабе. Для этого пространство состояний было нормализовано с использованием нормализации min-max, где нормализованное значение x_{norm} задается уравнением:

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}},$$

где x_{max}, x_{min} — максимальное и минимальное значение признака.

Пространство действий ($\mathcal{A}$). На каждом временном шаге возможные действия осуществляемые батареей ограничены следующим набором действий: оставить батарею в режиме холостого хода, зарядить батарею и разрядить батарею, учитывая ограничения модели:

1. Режим холостого хода. Весь спрос на электроэнергию покрывается за счет использования энергии, производимой фотоэлектрической станцией и/или покупаемой из сети.

2. Зарядить батарею. Зарядка аккумулятора с использованием энергии, вырабатываемой фотоэлектрическим устройством и при покупке энергии из сети, спрос на электроэнергию также покрывается за счет использования энергии, производимой фотоэлектрической станцией и/или покупаемой из сети.

3. Разрядить батарею. Спрос на электроэнергию частично или полностью покрывается за счет использования энергии батареи, фотоэлектрической станции и/или покупаемой из сети энергии.

Таким образом в любой момент времени t агент может выполнить одно из действий $a_t \in [P_{ch}^{max}; P_{dch}^{max}]$. Однако, для DQN подхода необходимо дискретизировать пространство возможных действий, поэтому в контексте

поставленной задачи равномерно отбирается 12 значений из указанного промежутка.

Резервный контроллер. Определяется, что батарея оснащена механизмом контроля, который гарантирует, что ограничения батареи не нарушаются. Резервный контроллер – это встроенная система, которая может индуцировать зарядку или разрядку аккумулятора в зависимости от текущего SoC и предопределенной логики. Резервный контроллер действует как фильтр для каждого управляющего действия. На каждом временном шаге t алгоритм 4 представляет резервный контроллер, сопоставляющий предлагаемое управляющее действие a_t с фактическим управляющим действием a^t в зависимости от SoC батареи.

Исходные параметры: $\eta_{ch}, \eta_{dch}, SoC_{min}, SoC_{max}$
Результат: a^t
Входные данные: $SoC(t), a_t$
если $a_t > 0$ **тогда**
  если $SoC(t) + a_t\eta_{ch} < SoC_{max}$ **тогда**
    $a^t \leftarrow a_t$,
    $SoC(t) \leftarrow SoC(t) + a^t\eta_{ch}$
  иначе
    $a^t \leftarrow (SoC_{max} - SoC(t))/\eta_{ch}$,
    $SoC(t) \leftarrow SoC_{max}$
  конец
иначе
  если $SoC(t) + a_t/\eta_{dch} > SoC_{min}$ **тогда**
    $a^t \leftarrow a_t$,
    $SoC(t) \leftarrow SoC(t) + a^t/\eta_{ch}$
  иначе
    $a^t \leftarrow (SoC(t) - SoC_{min}) * \eta_{dch}$,
    $SoC(t) \leftarrow SoC_{min}$
  конец
конец

Алгоритм 4: Резервный контроллер

Настройки резервного контроллера неизвестны обучающему агенту. Однако агент может измерить результат управляющего действия по штрафу, который он получает.

Функция награды. Цель этой работы состоит в том, чтобы макси-

мизировать эффективность использования накопительной системы и минимизировать финансовые затраты пользователей, поэтому функция награды задана следующим образом:

$$\rho(s, a) = P_{buy}(t) \cdot C_{buy}(t) - P_{sell}(t) \cdot Csell(t),$$

где $C_{buy}(t)$ и $Csell(t)$ – тарифы на покупку и продажу электроэнергии в течение 15-минутного периода, P_{buy} и P_{sell} представляют собой количество энергии импортируемое из сети или экспортируемое в нее соответственно. Значения $P_{buy}(t)$ и $P_{sell}(t)$ являются следствием управляющего действия a^t и определяются с помощью алгоритмов 1 и 4.

3.5 Глубокое Q-обучение (DQN)

DQN обучается с помощью варианта алгоритма Q-обучения, использующего стохастический градиентный спуск для обновления параметров. Во-первых, функция ценности из стандартного алгоритма RL заменяется глубокой Q-сетью с параметрами θ, заданными весами и смещениями, такими, что $Q(s, a, \theta) \approx Q^{\pi}(s, a)$. Это приближение используется далее для определения целевой функции по среднеквадратичной ошибке Беллмана:

$$\mathcal{L}(\theta) = E((r + \gamma \underset{a_{t+1}}{max} Q(s_{t+1}, a_{t+1}, \theta) - Q(s_t, a_t, \theta))^2).$$

Что приводит к следующему градиенту Q-обучения:

$$\frac{\partial \mathcal{L}(\theta)}{\partial \theta} = E((r + \gamma \underset{a_{t+1}}{max} Q(s_{t+1}, a_{t+1}, \theta) - Q(s_t, a_t, \theta)) \frac{\partial Q(s_t, a_t, \theta)}{\partial \theta}).$$

Обычно стандартный DQN алгоритм колеблется или расходится, это может происходить из-за того, что данные являются последовательными. Для того чтобы преодолеть ограничение коррелированных данных и нестационарных распределений зачастую используется прием воспроизведения опыта, который случайным образом выбирает предыдущую мини-партию переходов $< s_t, a_t, r_t, s_{t+1} >$ из набора данных D размера N и, следовательно, сглаживает распределение обучения по историческим данным. Так же для

лучшей сходимости используется ϵ-жадная стратегия выбора действий, которая на каждом шаге с вероятностью $(1-\epsilon)$ выбирается действие в соответствии с правилом $a_t = \underset{a_t^* \in \mathcal{A}}{argmax} Q(s(t), a_t^*)$, а с вероятностью ϵ – случайное действие с одинаковыми вероятностями для каждого допустимого действия. Величину ϵ с каждой итерацией обучения предлагается уменьшать, чтобы пока агент имеет мало опыта, он больше искал новые варианты действий, а после изучения нескольких эпизодов, больше опирался на выработанную стратегию, и обучение сосредотачивалось на улучшении уже сформированного управления.

Исходные параметры: D, N, C, γ, θ, $\theta^* \leftarrow \theta$
Результат: $Q(s, a, \theta^*)$
для каждого $episode = \overline{1, M}$ **выполнять**
  Выходные данные:
  $S_1 = (S_t^d, S_t^q, SoC, S^l, S^{pv}, S_x^b, S_x^s, S_x^{l_0}, S_x^{pv_0}, S_x^{b_0}, S_x^{s_0})$
  для каждого $t = \overline{1, T}$ **выполнять**
    выбрать действие a_t в состоянии S_t
    с помощью ϵ-жадной политики;
    выполнить действия резервного контроллера $a_t \leftarrow$a^t;
    перейти в состояние
     $S_{t+1} = (S_t^d, S_t^q, SoC, S^l, S^{pv}, S_x^b, S_x^s, S_x^{l_0}, S_x^{pv_0}, S_x^{b_0}, S_x^{s_0})$;
    получить награду R_t;
    сохранить преход (S_t, a_t, R_t, S_{t+1}) в D;
    выбрать случайно mini batch размера N переходов
     (S_j, a_j, R_j, S_{j+1}) из D
    для каждого $(S_j, a_j, R_j, S_{j+1}) \in minibatch$ **выполнять**
      если $j = T - 1$ **тогда**
        $y(j) = R_j$
      иначе
        $y(j) = R_j + \gamma \cdot Q(s_{j+1}, \underset{a}{argmin} Q(s_{j+1}, a, \theta), \theta^*)$
      конец
    конец
    обновить веса Q-сети $E[(y_j - Q(s_j, a_j, \theta))^2]$;
    каждые C шагов переписать $\theta^* \leftarrow \theta$
  конец
конец

Алгоритм 5: Управление накоплением энергии с помощью DQN

Также одним из недостатков стандартного алгоритма Q-обучения яв-

ляется тенденция к значительному завышению оценок функции ценности. Причиной является тот факт, что Q-обучение из множества альтернативных действий выбирает то, которое максимизирует функцию ценности, однако из-за стохастической природы шума в данных приводит к тому, что ценность действия оказывается завышена. Подобное систематическое завышение приводит к изменению оптимальной стратегии поведения агента, к которой в асимптотике сходится алгоритм. Для решения данной проблемы было предложено использовать две нейронные сети. Одна из них обучается обычным алгоритмом Q-обучения на выбранном обучающем наборе, вторая же используется для оценки функции награды от выбранного действия, и обновляется значением первой нейронной сети каждые несколько шагов. Такое решение значительно уменьшило размер переоценки функции награды и привело к улучшению достигнутого результата на тестовом наборе окружений. Алгоритм 5 иллюстрирует предложенный DQN для управления накоплением энергии. Однако DQN определяется на дискретном пространстве действий, что может вести к ухудшению результатов оптимизации, в качестве альтернативы предлагается рассмотреть метод глубоких градиентов политики (DDPG).

3.6 Глубокий детерминированный градиент политики (DDPG)

Глубокий детерминированный градиент политики – это метод оценки политики, в котором параметризованная модель $\pi(s, \theta)$ основана на нейронной сети глубокого обучения. Градиент политики описывает оптимальную политику каждого состояния среды через функцию распределения вероятностей, то есть оценивается вероятность принятия действия a в заданном состоянии s: $\pi(s; \theta) = p(a|s,)a \in \mathcal{A}$.

Структура сети DDPG состоит из двух частей: сеть стратегией (актор) $\pi(s, \theta^{\pi})$ для аппроксимации функции политики $\pi(s)$ и сеть функции полезности (критик) $Q(s, a, \theta^{Q})$ для аппроксимации функции ценности $Q(s, a)$, они формируют стратегию подобно алгоритму Reinforce, описанному выше. Отличие заключается в том, что для расчета градиента исполь-

зуется функция полезности, а не траектория. При этом определяет действие агента сеть актор, а во время перехода системы в новое состояние и получении сигнала о значении награды, сеть критик вычисляет функцию полезности по методу временной разности при помощи уравнения Беллмана, которая используется для оценки действия и корректировки весов моделей.

Исходные параметры: D, N, C, γ, θ^Q, θ^π, $\theta^{Q*} \leftarrow \theta^Q$, $\theta^{\pi*} \leftarrow \theta^\pi$
Результат: $\pi(s, \theta^{\pi*})$
для каждого $episode = \overline{1, M}$ **выполнять**
  Выходные данные:
  $S_1 = (S_t^d, S_t^q, SoC, S^l, S^{pv}, S_x^b, S_x^s, S_x^{l_0}, S_x^{pv_0}, S_x^{b_0}, S_x^{s_0})$
  для каждого $t = \overline{1, T}$ **выполнять**
    выбрать действие a_t в состоянии S_t
    с помощью ϵ-жадной политики;
    выполнить действия резервного контроллера $a_t \leftarrow \mathrm{a}^t$;
    перейти в состояние
     $S_{t+1} = (S_t^d, S_t^q, SoC, S^l, S^{pv}, S_x^b, S_x^s, S_x^{l_0}, S_x^{pv_0}, S_x^{b_0}, S_x^{s_0})$;
    получить награду R_t;
    сохранить преход (S_t, a_t, R_t, S_{t+1}) в D;
    выбрать случайно mini batch размера N переходов
     (S_j, a_j, R_j, S_{j+1}) из D
    для каждого $(S_j, a_j, R_j, S_{j+1}) \in minibatch$ **выполнять**
      если $j = T - 1$ **тогда**
        $y(j) = R_j$
      иначе
        $y(j) = R_j + \gamma \cdot Q(s_{j+1}, \pi(s_{i+1}, \theta^{\pi^*}), \theta^Q)$
      конец
    конец
    обновить веса сети критиков $E((y_j - Q(s_j, a_j, \theta^Q))^2)$;
    обновить веса сети акторов
     $E(\nabla_a Q(s, a, \theta^Q)|_{s=s_i, a=\pi(s_i)} \nabla_{\theta^\pi} \pi(s, \theta^\pi)|_{s=s_i})$;
    Каждые C шагов переписать
    $\theta^{Q*} \leftarrow \theta^Q$, $\theta^{\pi*} \leftarrow \theta^\pi$;
  конец
конец

Алгоритм 6: Управление накоплением энергии с помощью DDPG

Алгоритм 6 иллюстрирует предложенный DDPG для управление накоплением энергии, где для улучшения сходимости, как и в DQN, исполь-

зуется ϵ-жадная стратегия, вводится память воспроизведения и двойные нейронные сети.

Глава 4. Проведение экспериментов

4.1 Оборудование и программное обеспечение

Для реализации моделей был выбран объектно-ориентированный и высокоуровневый язык программирования с динамической семантикой Python3, как популярный инструмент разработки с большим выбором библиотек и фреймворков, он является кроссплатформенным, что обеспечивает совместимость со многими операционными системами.

Python - это интерпретируемый язык программирования, который ориентирован на повышение производительности разработчика и читаемость кода. Он обладает масштабируемостью и продуманной модульностью, а также большим функционалом и применяется для широкого круга задач.

1. tensorflow 1.5.0 – открытая библиотека, предоставляющая возможность построения и тренировки моделей машинного обучения. Для создания мощного и интуитивно понятного алгоритма используется в связке с библиотекой keras.

2. keras 2.1.4 – открытая нейросетевая библиотека. Она нацелена на оперативную работу с сетями глубокого обучения, содержит многочисленные реализации широко применяемых строительных блоков нейронных сетей, таких как целевые и передаточные функции, слои, оптимизаторы.

3. matplotlib 2.1.1 – библиотека для визуализации данных и результатов. Позволяет построить диаграммы разброса, гистограммы, поля градиентов и многое другое. Осуществляется поддержка основных форматов изображений: PNG, JPEG, PDF, SVG.

4. pandas. 0.22.0 – библиотека для манипулирования массивами данных и первичного анализа. Данный пакет предоставляет специаль-

ные структуры данных, такие как DataFrame, и встроенные функции для обработки временных рядов и числовых таблиц.

5. numpy 1.14.0 – библиотека с открытым исходным кодом с возможностью поддержки многомерных массивов, высокоуровневых математических функций, предназначенных для работы с многомерными массивами.

6. cplex 20.1.0.1 — пакет программного обеспечения, предназначенный для решения задач линейного и квадратичного программирования, в том числе целочисленного программирования.

Для реализации программного комплекса использовался фреймворк для веб-приложений Django на языке Python, использующий шаблон проектирования MVC, который помогает реализовать быструю разработку для надежных сайтов с легкой поддержкой. Django содержит в себе обширный функционал для веб-разработки например: панель управления сайтом, аутентификация пользователей (вход, выход, регистрация), различные встроенные формы, инструменты для загрузки файлов и т.д. Также он является бесплатный и имеет подробную документацию.

Для реализации системы хранения данных было решено использовать объектно-реляционную систему управления базами данных PostgreSQL. Данная СУБД имеет широкие возможности и высокую производительность.

Эксперимент был проведен на ПК с следующими характеристиками: 2-ядерный процессор Intel Core i5, 1,8 GHz, 8 ГБ.

Репозиторий Github с реализацией протестированных моделей и визуализацией результатов можно найти в [45], в [46] представлен репозиторий Github с програмным модулем.

4.2 Построение моделей

Для того чтобы агент в заданной среде эффективно обучался, необходимо провести настройку множества гиперпараметров. Наиболее важные и чувствительные к изменениям гиперпараметры в глубоком обучении с

подкреплением это: архитектура нейронной сети, размер памяти воспроизведения и батча, скорость обучения, коэффициент дисконтирования, значение ϵ-жадной политики. Не существует определенного метода настройки этих параметров или значений, которые подходят для работы в произвольной среде. Наиболее подходящие значения и диапазоны для прогонов были взяты из литературы [47]. Ниже приведено описание гиперпараметров, которые были настроены:

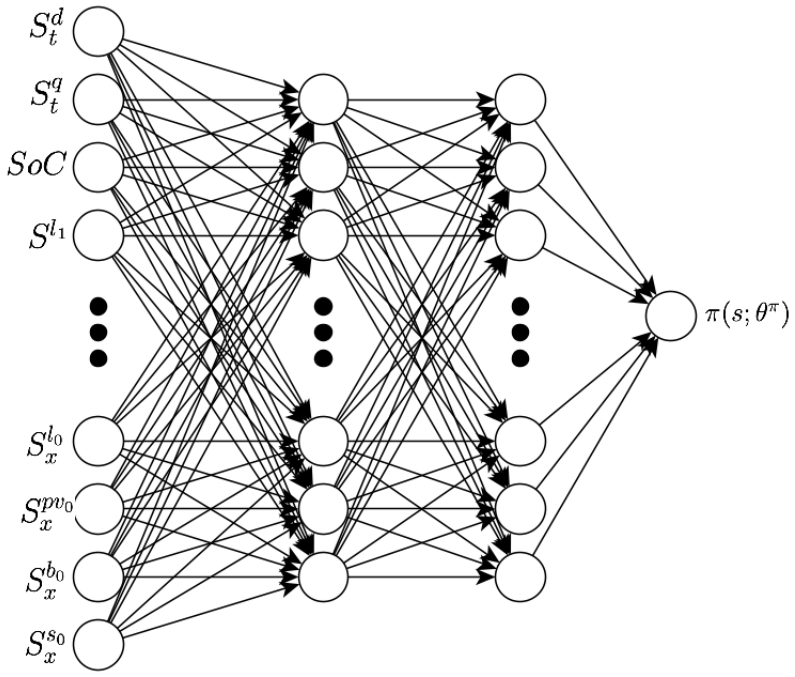


Рис. 10: Ахритектура DDPG

1. Коэффициент дисконтирования. Коэффициент дисконтирования γ находится в диапазоне $0 < \gamma < 1$ и определяет, какое значение агенту следует придавать вознаграждениям в настоящем и будущем. Когда γ приближается к нулю, агент сосредотачивается на немедленном вознаграждении по сравнению с возможной прибылью в будущем. И наоборот, по мере приближения γ к единице агенту рекомендуется планировать долгосрочные действия и учитывать ценность будущих вознаграждений. В этой работе значение коэффициента дисконтирования устанавливается равным $\gamma = 0.99$.

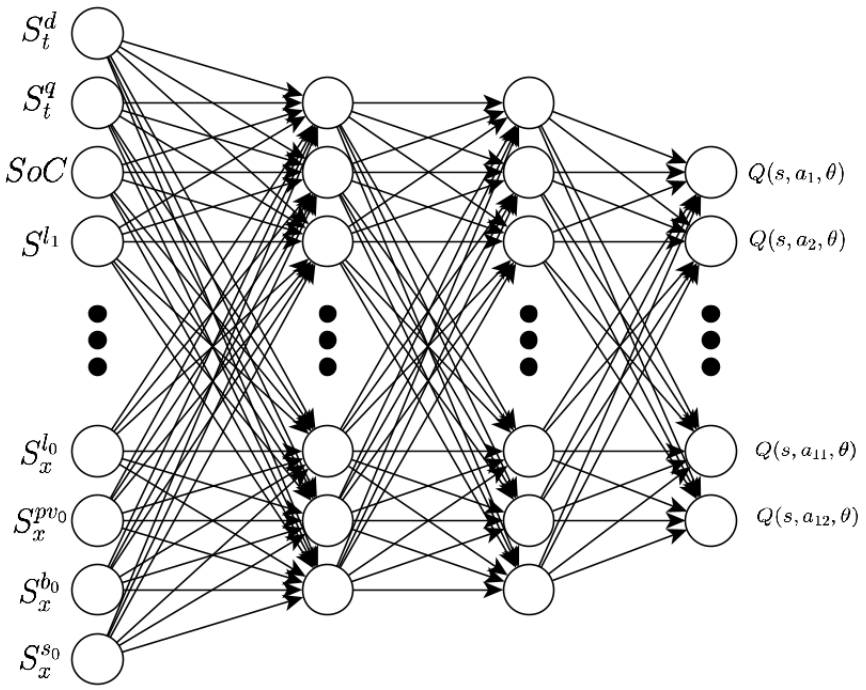


Рис. 11: Ахритектура DQN

2. Размер памяти воспроизведения. Память воспроизведения используется при обновлении весов нейронной сети и для обобщения модели требуются пакеты большого размера, особенно для пространства непрерывных действий. Поэтому устанавливается размер памяти воспроизведения на 1000 наблюдений, а размер батча, как 30% от общей памяти.

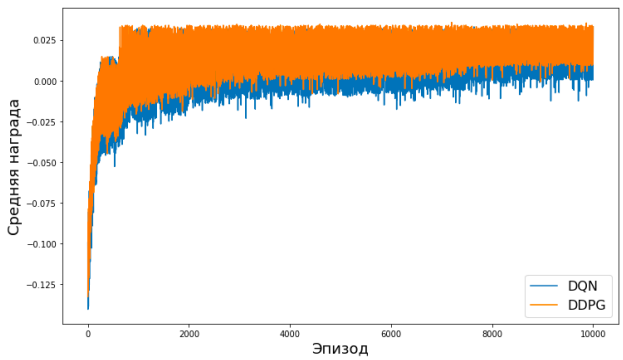
3. Скорость обучения. Скорость обучения соответствует размеру шага обновления весов нейронной сети при градиентном спуске. Как правило, более высокая скорость обучения приравнивается к более быстрому, хотя и нестабильному обучению. И наоборот, чем ниже скорость обучения, тем медленнее, но более стабильнее обучение. Устанавливается скорость обучения на $\alpha = 10^{-3}$.

4. ϵ-жадная политика. Как было описано выше, ϵ-жадная политика, на каждом шаге с вероятностью $(1 - \epsilon)$ выбирается действие в соответствии с политикой, а с вероятностью ϵ – случайное действие путем розыгрыша по жребию с одинаковыми вероятностями выбора для каждого допустимого действия. Однако исследователями часто при-

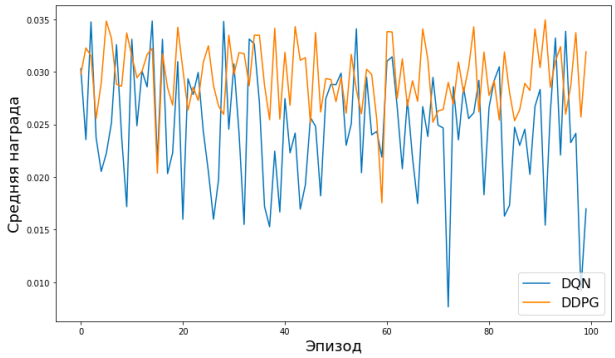
нимается стратегия, в соответствии с которой величину ϵ в процессе обучения необходимо уменьшать, чтобы пока агент имеет мало опыта, он больше «экспериментировал», искал новые варианты действий, а накопив опыт, больше доверял ему, и обучение сосредотачивалось на улучшении уже сформированного управления. Поэтому изначально устанавливается $\epsilon = 0.6$ через каждые первые 1000 итераций значение уменьшалось на 0.1, когда значение достигло $\epsilon = 0.1$ параметр не изменялся.

5. Нейронная сеть. Для того чтобы провести справедливое сравнение между DQN и DDPG, архитектура используемых глубоких нейронных сетей аналогична для обеих моделей и состоит из одного входного слоя, одного выходного слоя и двух скрытых слоев. Каждый слой состоит из узлов, которые соединены с узлами соседних слоев. Слой, в котором каждый узел имеет соединения со всеми другими узлами в соседних слоях, называется плотным слоем. В этом исследовании использовались сети, состоящие из 2 скрытых слоев, содержащие 300 узлов в каждом слое. В качестве функций активации предлагается использовать ReLU. Что касается количества узлов на слой, то входной слой имеет такое же количество узлов, что и вводимое пространство объектов и составляет 391 нейрон. Выходной слой для DQN имеет 12 нейронов, по количеству доступных для агента действий. Однако в случае с DDPG нейронная сеть имеет один выходной нейрон, соответствующий непрерывному значению действия. Иллюстрация архитектуры глубоких нейронных сетей для DDPG и DQN приведена на Рис.10 и Рис.11 соответственно.

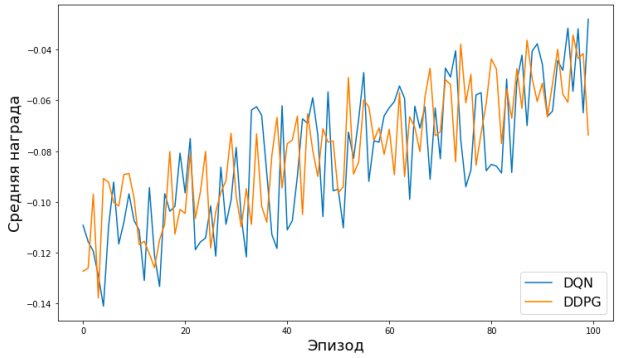
Агент изучает 10000 эпизодов в соответствии со средой, описанной в разделе 3.4. После обновления весов нейронной сети по завершению эпизода проводится испытание на 1 эпизоде и вычисляется средняя награда за это время. Сходимость среднего вознаграждения с использованием предложенных методов показана на Рис.12, также представлена сходимость алгоритма за последние 100 эпизодов учебного времени и на первых 100 эпизодах. С увеличением количества эпизодов действия агента становятся более оправ-



(а) 10000 эпизодов



(б) 100 последних эпизодов



(в) 100 первых эпизодов

Рис. 12: Значения среднего вознаграждение агента

данными и обеспечивают уменьшение затрат пользователя. В начале агент выберает действия случайным образом в соответствии с ϵ-жадная политикой. Поэтому величина среднего вознаграждения колеблется, на первых 100 эпизодах значение размаха достигает 0.10 и 0.11 для DDPG и DQN, а на последних 0.014 и 0.027 соответственно. Можно заметить, что среднее вознаграждение сходится к оптимальным точкам для обоих методов, однако результаты DDPG являются более стабильными из-за неограниченного пространства действий.

4.3 Сравнение MILP, DQN и DDPG

После завершения обучения моделей было произведено сравнение эффективности работы DQN и DDPG с детерминистическим подходом, результаты для каждого из методов приведены в таблицах 5 – 7 соответственно. Где $Test\ case$ – номер тестового случая, $Period$ – период моделирования в тестовом случае, $Money_{spent}$ – затраты пользователя с использованием

энергосберегающей батареи, $Money_{no\_batt}$ – затраты пользователя без использования энергосберегающей батареи, $Battery$ – номер батареи, которая применяется для тестирования алгоритма. Для представления результатов оптимизации сравнивается время выполнения алгоритма – $Time$ (s), как один из важных показателей в современном энергоменеджменте, и применяется метрика – $Score$, равная среднему значению относительных затрат на покупку электроэнергии с использованием батареи, к затратам на покупку электроэнергии без батареи:

$$Score = \frac{Money_{spent} - Money_{no\_batt}}{|Money_{no\_batt}|}.$$

Таблица 5: Результаты детерминистического подхода

Test case	Battery	Period	Money$_{spent}$	Money$_{no_batt}$	Score	Time(s)
1	1	1	1163	1233	-0.056	21.73
1	1	2	1075	1139	-0.055	22.35
1	1	3	1484	1547	-0.041	22.53
1	1	4	2743	2870	-0.044	22.34
1	1	5	4368	4666	-0.063	22.76
.....						
11	2	1	3173	3479	-0.087	24.84
11	2	2	2146	3060	-0.298	26.21
11	2	3	1262	2483	-0.491	21.94
11	2	4	1556	1983	-0.215	24.14
11	2	5	2282	2629	-0.132	27.40

Результаты сравнения DQN, DDPG и детерминистического подхода приведены в Таблице 8. По результатам экспериментов DDPG и DQN обеспечивают снижение пользовательских затрат на покупку электроэнергии на 19,3% и 18,6% соответственно, против 18,7% при детерминистическом подходе. Это является следствием того, что в процессе обучения методы RL могут адаптироваться и выявлять неопределенности в исторических данных, однако из-за ограниченности пространства действий DQN не так стабильно реагирует на изменения в среде, как DDPG. При этом время решения задачи в подходах обучения с подкреплением уменьшается более

Таблица 6: Результаты DQN подхода

Test case	Battery	Period	Money_{spent}	Money_{no_batt}	Score	Time(s)
1	1	1	1162	1233	-0.057	6.877
1	1	2	1079	1139	-0.052	6.28
1	1	3	1485	1547	-0.040	6.39
1	1	4	2743	2870	-0.044	6.33
1	1	5	4360	4666	-0.066	7.52
.....						
11	2	1	3175	3479	-0.087	8.10
11	2	2	2139	3060	-0.300	8.41
11	2	3	1268	2483	-0.489	7.94
11	2	4	1560	1983	-0.213	8.05
11	2	5	2274	2629	-0.135	7.09

Таблица 7: Результаты DDPG подхода

Test case	Battery	Period	Money_{spent}	Money_{no_batt}	Score	Time(s)
1	1	1	1152	1233	-0.065	7.215
1	1	2	1060	1139	-0.069	6.21
1	1	3	1480	1547	-0.043	6.42
1	1	4	2736	2870	-0.046	6.35
1	1	5	4361	4666	-0.066	7.31
.....						
11	2	1	3152	3479	-0.093	8.11
11	2	2	2115	3060	-0.310	8.21
11	2	3	1264	2483	-0.49	7.92
11	2	4	1530	1983	-0.228	8.01
11	2	5	2269	2629	-0.137	7.13

чем в 3.5 раза, что свидетельствует о возможном масштабировании методов DQN и DDPG для решения реальных производственных задач.

На Рис. 13 представлена визуализация результата работы DDPG, DQN и детерминистического подхода для 1 периода моделирования 1 тестового случая, который включает 10 дней. Как можно заметить, при обучении агент в обоих методах выявил стратегию управления зарядом аккумулятора. Прослеживается тенденция работы накопительного элемента: в периоды пика выработки фотоэлектрической станции SoC достигает своих

Таблица 8: Сравнение MILP, DQN,DDPG

Мотод	Money_{spent}	Money_{no_batt}	Score	Time(s)
MILP	309678	381847	-0.187	6240
DQN	310823	381847	-0.186	1673
DDPG	308150	381847	-0.193	1657

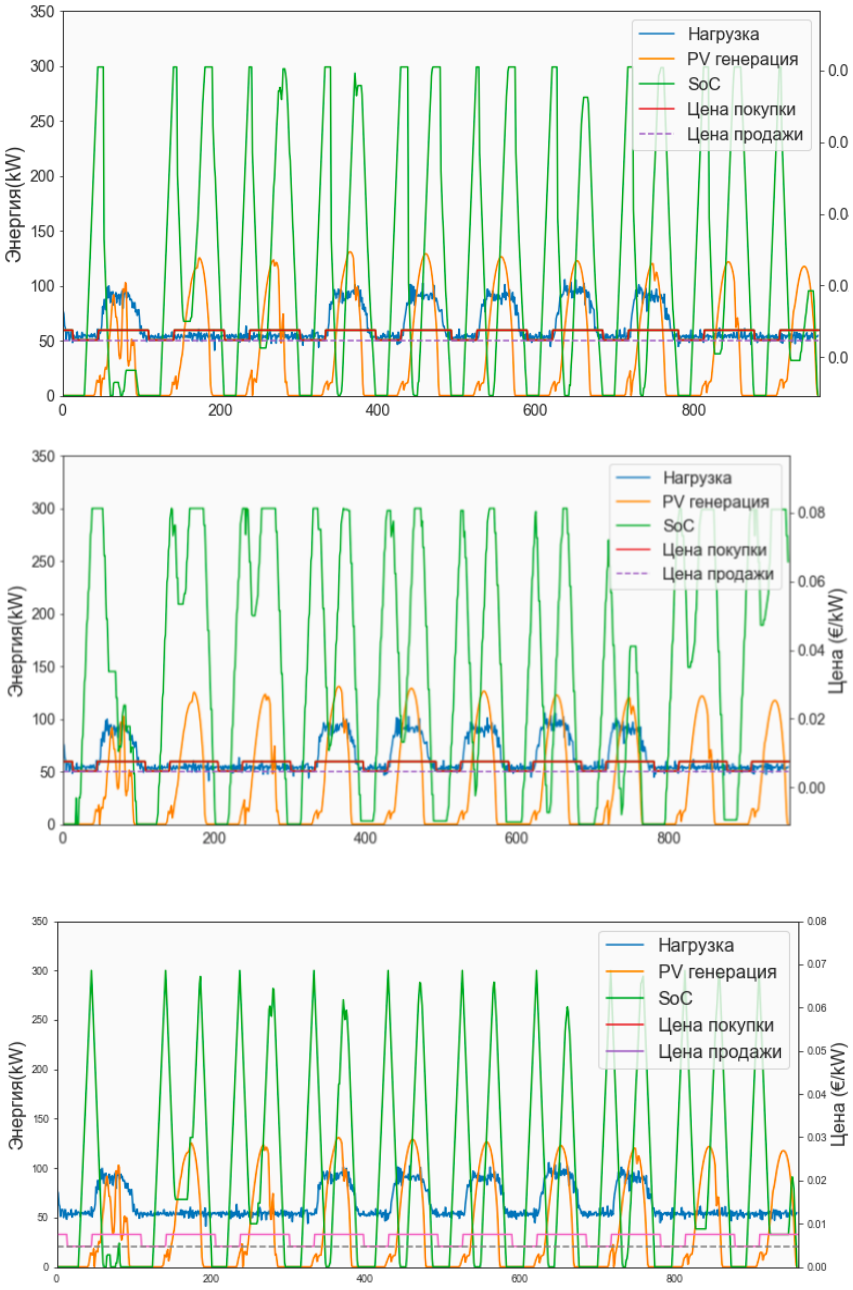
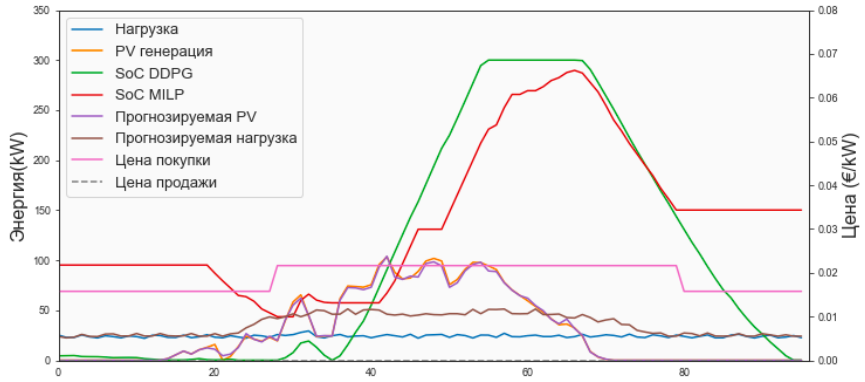
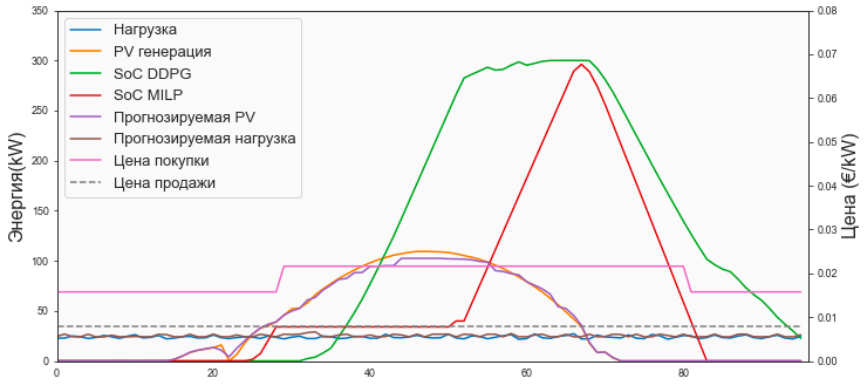


Рис. 13: Профили энергопотребления и зарядки/разрядки батареи для 1 тестового случая DDPG, DQN и MILP соответственно

максимальных значений, затем в периоды недостаточного самообеспечения энергией станции, батарея разряжается.



(а) WAPE = 10.32%



(б) WAPE = 2.14%

Рис. 14: Управление накопительной батареей DDPG и MILP подход

По сравнению с методами обучения с подкреплением, детерминистический подход дает более стабильный и предсказуемый результат, но сильно зависящий от входных данных, что в итоге влияет на результаты оптимизации энергосистемы пользователя. К примеру на Рис. 14 (а) из-за завышенного прогноза нагрузки батарея медленнее заряжается или находится в режиме холостого хода, когда актуальная выработка фотоэлектрической станции превышает спрос на энергию, при этом экспорт энергии в сеть нецелесообразен по причине очень низкой цены продажи. Случай Рис. 14 (б) с меньшей взвешенной абсолютной процентной ошибкой прогнозирования иллюстрирует преимущество детерминистического подхода, когда DDPG медленно реагирует на изменение профиля выработки энергии, действия MILP соответствуют ожиданиям.

В зависимости от характеристик батареи меняется показатель относительных затрат – *Score* (Рис.15). При увеличении вместимости батареи и ее мощности в 2 раза в среднем показатель оптимизации увеличивается в 1.5 раза. Задача определения оптимальных параметров накопительного элемента является отдельным важным этапом в проектировании энергосистемы пользователей, которая требует отдельного рассмотрения.

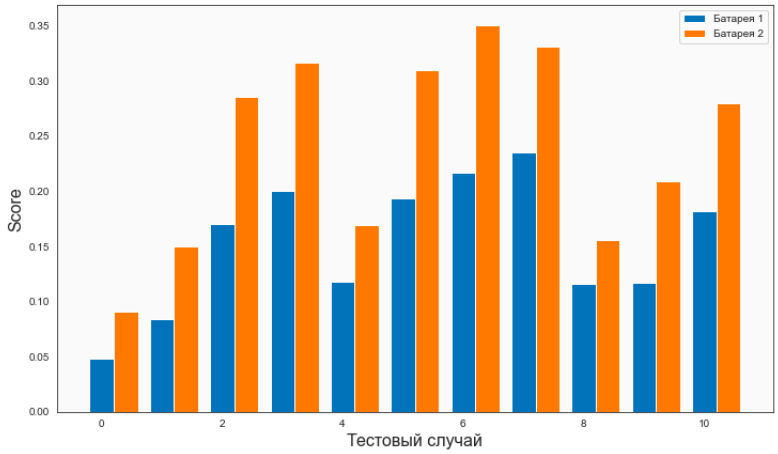


Рис. 15: Зависимость *Score* от характеристик накопительной батареи

Глава 5. Программный комплекс

В этой главе будет представлена реализация программного комплекса на языке Python. Для комфортного взаимодействия пользователя с разработанной средой необходимо клиент-серверное приложение. Программный модуль представляет собой веб-приложение, в котором клиент взаимодействует с веб-сервером при помощи браузера. Процесс описывается следующими шагами:

1. Сбор показаний датчиков (актуальная нагрузка, выработка фотоэлектрической станции за прошлый момент времени, текущая остаточная мощность батареи, тариф на покупку и продажу энергии), прогнозных данных (нагрузка, выработка фотоэлектрической станции, цена на покупку и продажу энергии), характеристик оборудования.

2. Передача данных на сервер.

3. Контроллер посылает запрос на сервер для анализа текущего состояния энергосистемы.

4. Сервер принимает запрос, формирует задачу и направляет ее вычислительному приложению.

5. Вычислительное приложение по исходным данным определяет SoC на следующий временной шаг и отправляет результат на сервер.

6. Сервер перенаправляет результат контроллеру.

7. Контроллер по показаниям с датчиков и результатом работы вычислительного приложения вырабатывает стратегию, инициализирует подключение к различным компонентам энергосистемы и отправляет соответствующие команды. Затем рассчитывает необходимые пользователю показатели и отправляет их на сервер.

8. Далее происходит перенаправление результата пользователю. Где имеется инструмент для визуализации и интерпретации работы энергосистемы.

Также реализована учетная запись пользователя с возможностью редактирования и изменения исходных данных. Архитектура программного комплекса представлена на Рис. 16.

5.1 База данных

При использовании программного комплекса, для подключения пользователя к системе, потребуется хранить некоторые пользовательские данные и системные значения. Также, необходимо хранить данные с показателями датчиков и контрольными значениями. На Рис. 17 представлена структура, полученной базы данных.

Таблица **user** необходима для хранения персональных данных пользователей, они используются при процедуре аутентификации. При реги-

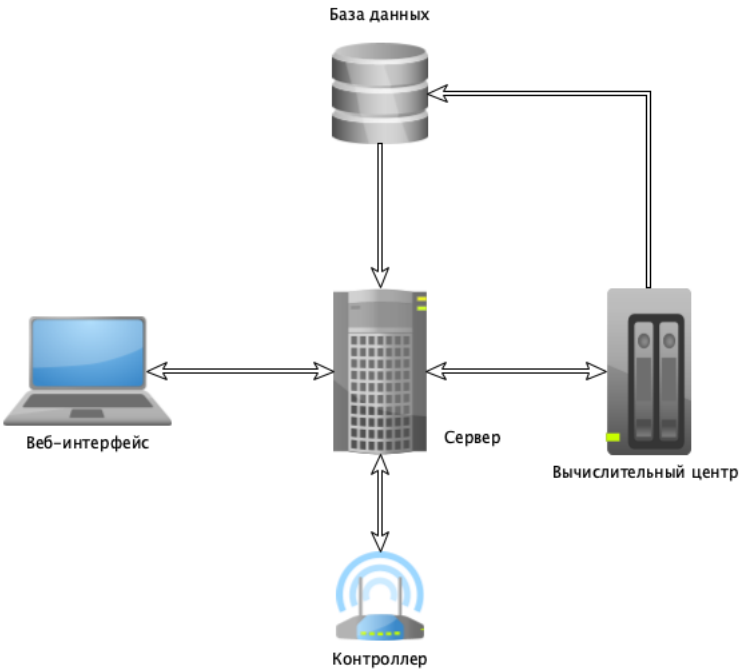


Рис. 16: Архитектура программного комплекса

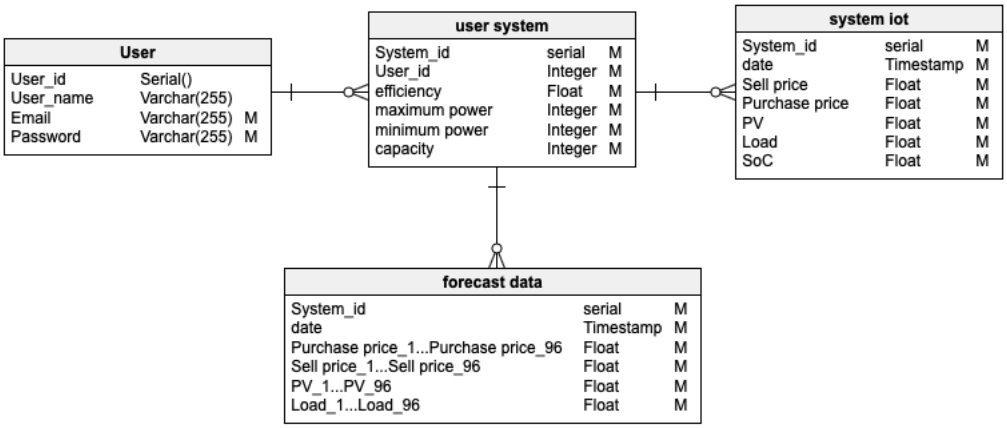


Рис. 17: Схема структуры базы данных

страции нового пользователя в данной таблице создается запись, которая содержит фамилию и имя пользователя, почту и захешированный пароль.

Таблица **user system** содержит информацию с техническими характеристиками оборудования: эффективность зарядки/разрядки аккумулятора, максимальная зарядная/разрядная мощность батареи, минимальное/максимальное состояние заряда батареи.

Таблица **system iot** содержит информацию о показаниях датчиков:

тариф на покупку/продажу энергии, жилая нагрузка, мощность фотоэлектрической станции, состояние заряда батареи. Каждое показание ссылается на запись в user system для идентификации системы, с которой было снято показание, и пользователя системы.

Таблица **forecast data** содержит информацию о прогнозных значениях: тариф на покупку/продажу энергии, жилая нагрузка, мощность фотоэлектрической станции на следующие сутки с шагом в 15 минут. Они необходимы для работы DDPG модели. Как и в таблице system iot каждое показание ссылается на запись в user system и пользователя системы.

5.2 Функционал приложения

Для комфортного взаимодействия пользователя с разработанной средой реализованы различные запросы, облегчающие контроль за энергосистемой. Программный комплекс имеет следующий функционал:

- **Регистрация нового пользователя.** При реализации этого запроса создается новая запись в таблице user. Запрос содержит фамилию и имя пользователя, почту и пароль.

- **Добавление технические характеристики оборудования.** Данный запрос создает новую запись в таблице user system. Запрос содержит эффективность зарядки/разрядки аккумулятора, максимальную зарядную/разрядную мощность батареи, минимальное/максимальное состояние заряда батареи.

- **Обучение модели.** Данный запрос запускает процесс обучения модели. Это необходимо, если в системе нет модели подходящей для пользователя. Запрос содержит массив исторических данных и технические характеристики оборудования пользователя.

- **Удаление имеющейся системы.** Данный запрос удаляет данные о выбранной системе из таблицы user system. Запрос содержит идентификационный номер системы.

- **Подключение управление системой.** Данный запрос запускает скрипт, который каждые 15 минут собирает данные из таблиц user system, forecast data, system iot и формирует стратегию для управления энергосистемой пользователя.

- **Визуализация работы энергосистемы.** Данный запрос выполняет динамическую визуализацию работы энергосистемы и предоставляет отчет о текущих процессах: затраты на энергию, состояние заряда батареи, выработка фотоэлектрической станции, нагрузка энергосистемы за последние сутки.

Выводы

В данной выпускной квалификационной работе объектом исследований были выбраны системы электроснабжения с распределенными энергетическими ресурсами, включающие генерацию возобновляемой энергии от фотоэлектрической станции и систему хранения энергии в виде аккумуляторных батарей, как одни из самых востребованных типов микросетей среди пользователей. Для эффективной работы микрорешетки был предложен метод планирования графика хранения энергии с целью минимизации финансовых затрат пользователей. В процессе реализации были решены поставленные задачи и получены следующие результаты:

1. обзор существующих подходов и практик для оптимизации, стабилизации и управления электропотреблением, а также существующие индустриальные решения энергоменеджмента;

2. описана математическая постановка задачи в терминах смешанного целочисленного линейного программирования;

3. проведен первичный анализ данных, учитывая специфику поставленной задачи;

4. рассмотрены математические методы, применяемые к решаемой задаче, их преимущества и недостатки;

5. описаны подходы на основе обучения с подкреплением: глубокое Q-обучение (DQN), глубокий детерминированный градиент политики (DDPG);

6. разработана среда для обучения агента в рамках задачи энергоменеджмента, каждое состояние которой включает 391 параметр и состоит из: временной составляющей (2), неконтролируемой экзогенной составляющей (388), управляемой части (1). Действия агента в DQN дискретизированы до 12 возможных вариантов. Также введен резервный контроллер, который адаптирует агента к ограничениям системы. В качестве функции награды выбрано значение пользовательских затрат на энергию в течение 15 минут. Подробнее о постановке

задачи в терминах обучения с подкреплением написано в разделе 3.4, о реализации — в разделах 4.1 и 4.2;

7. произведено сравнение реализованных моделей на одном и том же наборе данных по метрике, определяющей относительное снижение пользовательских затрат: DQN (18,6%), DDPG (19,3%), детерминированный подход (18,7%). Также предложенные методы продемонстрировали большое преимущество по времени решения задачи, которое в среднем оказалось меньше в 3.5 раз по сравнению с решением задачи программным обеспечением cplex. Что свидетельствует о возможном масштабировании методов DQN и DDPG для решения реальных производственных задач. Подробнее сравнение описано в разделе 4.3;

8. для внедрения разработанного подхода реализован программный модуль, который упрощает взаимодействие пользователя с контролирующей системой и предлагает функционал для визуализации работы микрорешетки;

По результатам исследований был получен программный комплекс, способный управлять энергосистемой пользователя в условиях реального времени.

Заключение

В эпоху интеллектуальных сетей и умных домов потребность в имплементации эффективной управляющей компоненты в энергосистему пользователей возрастает с каждым годом. Домохозяйства и предприятия потребляют все больше энергии для обеспечения своих нужд, при этом цены на энергоресурсы продолжают расти. Ведь в последние годы несоответствие между спросом и предложением в энергетической сфере становится все более напряженным: ископаемая энергия истощается и становится менее доступной, к тому же многие экологи акцентируют внимание общественности на проблемах загрязнения окружающей среды при выработке энергоресурсов. В качестве решения проблемы были предложены возобновляемые ис-

точники энергии, но нестабильный характер выработки дестабилизирует энергосистему и делает ее менее эффективной, так как разработанные ранее алгоритмы контроля не могут учитывать все факторы, влияющие на динамику системы. Поэтому целью моего исследования была разработка быстрого и эффективного инструмента, отвечающего требованиям современного энергоменеджмента.

Наиболее перспективным направлением в контексте решаемой задачи является обучение с подкреплением, которое обладает преимуществом самообучения и исследует оптимальные стратегии с помощью механизма проб и ошибок в динамической среде. Алгоритмы RL предлагается использовать для решения различных проблем принятия решений в области управления в условиях неопределенности. В данной работе на основе имеющихся данных было предложено решение для гибкого планирования работы энергосистемы пользователя с целью минимизации финансовых затрат, оно включает в себя программный комплекс, способный в режиме реального времени управлять зарядом накопительной батареи и рассчитывать объемы энергетических потоков, необходимых для удовлетворения потребностей пользователя.

В качестве дальнейших исследований для увеличения энергоэффективности моделируемой системы будут рассмотрены технологии распределенной энергетики, такие как системы когенерации. Это требует дополнительных исследований и более детального изучения процесса совместной выработки электрической и тепловой энергии, а также большего массива исторических данных.

Список литературы

[1] Официальный сайт schneider electric [Электронный ресурс] / SE. Режим доступа: `https://www.se.com/ru/ru/`, свободный. (дата обращения: 6.05.21)

[2] Репозиторий соревнования [Электронный ресурс] / GitHub. Режим доступа: `https://github.com/drivendataorg/power-laws-optimization`, свободный. (дата обращения: 6.05.21)

[3] Sharma V., Bowden S. Peak load offset and the effect of dust storms on 10 MWp distributed grid tied photovoltaic systems installed at Arizona State University // 38th IEEE Photovoltaic Specialists Conference, 2012, P. 590–595.

[4] Yu Y., Cai Z., Huang Y. Energy Storage Arbitrage in Grid-Connected Micro-Grids Under Real-Time Market Price Uncertainty: A Double-Q Learning Approach // IEEE Access, 2020. Vol. 8, P. 54456–54464.

[5] Cardona E., Piacentino A. Optimal design of CHCP plants in the civil sector by thermoeconomics // Applied Energy, 2007. Vol. 84, No. 7, P. 729–748.

[6] Georgilakis P. S., Hatziargyriou N. D. Optimal distributed gener-ation placement in power distribution networks: models, methods, andfuture research // IEEE Transactions on power systems, 2013. Vol. 28, No. 3, P. 3420–3428.

[7] Anatone M., Panone V. A Model for the Optimal Management of a CCHP Plant // Energy Procedia, 2015. Vol. 81, No. 69, P. 399–411.

[8] Perez A., Moreno R. Effect of Battery Degradation on Multi-Service Portfolios of Energy Storage // IEEE Transactions on Sustainable Energy, 2016. Vol. 7, P. 1718–1729.

[9] Gengo T., Kobayashi Y. Development of Grid-stabilization Power-storage System with Lithium-ion Secondary Battery // Mitsubishi Heavy Industries Technical Review, 2009. Vol. 46, No. 2, P. 36–42.

[10] Atia R., Yamada N. Sizing and analysis of renewable energy and battery systems in residential microgrids // IEEE Transactions on Smart Grid, 2016. Vol. 7, No. 3, P. 1204—1213.

[11] Bahramirad S., Reder W., Khodaei A. Reliability-constrained optimal sizing of energy storage system in a microgrid // IEEE Transactions on Smart Grid, 2012. Vol. 3, No. 4, P. 2056–2062.

[12] Dulout J., Hernandez L. Optimal Scheduling of a Battery-based Energy Storage System for a Microgrid with High Penetration of Renewable Sources // ELECTRIMACS Conference, 2017. P. 1–6.

[13] Wang J., Liu J. Optimal scheduling of gas and electricity consumption in a smart home with a hybrid gas boiler and electric heating system // Energy, 2020. Vol. 204, P. 117951.

[14] Hatziargyriou N. Special issue on microgrids and energy management // Eur Trans Electr Power, 2011. Vol. 21, P. 1139–1141.

[15] Reddy P.P., Veloso M.M. Strategy learning for autonomous agents in smart grid markets // Twenty-second International Joint Conference on Artificial Intelligence, 2011. P. 1446–1451.

[16] Chen C., Duan S., Cai T. Smart energy management system for optimal microgrid economic operation // Renewable Power Generation, 2011. Vol. 5, No. 3, P. 258–267.

[17] Mohamed F.A., Koivo H.N. System modelling and online optimal management of MicroGrid with battery storage // International Journal on Electrical Power and Energy Systems, 2010. Vol. 32, No. 5, P. 398–407.

[18] Colson C.M., Nehrir M.H., Pourmousavi S.A. Towards real-time microgrid power management using computational intelligence methods // IEEE, 2010. P. 1–8.

[19] Abdirahman M.A., Mustafa M.W. Autonomous Integrated Microgrid (AIMG) System // International Journal of Education and Research, 2014. Vol. 2, No. 1, P. 77–82.

[20] Chaouachi A., Rashad M., Kamel M. Multiobjective Intelligent Energy Management for a Microgrid // IEEE Transactions on Industrial Electronics, 2013. Vol. 60, No. 4, P. 1688–1699.

[21] Wang H., Huang T., Liao X. Reinforcement Learning for Constrained Energy Trading Games With Incomplete Information // IEEE Trans. Cybern, 2017. Vol. 47, P. 3404–3416.

[22] Kim B., Zhang Y. Dynamic Pricing and Energy Consumption Scheduling With Reinforcement Learning // IEEE Trans. Smart Grid, 2016. Vol. 7, P. 2187–2198.

[23] Ruelens F., Claessens B.J., Vandael S. Residential Demand Response of Thermostatically Controlled Loads Using Batch Reinforcement Learning // IEEE Trans. Smart Grid, 2017. Vol. 7, P. 2149–2159.

[24] Xiong R., Cao J., Yu Q. Reinforcement learning-based real-time power management for hybrid energy storage system in the plugin hybrid electric vehicle // Appl. Energy, 2018. Vol. 211, P. 538—548.

[25] Wei C., Zhang Z., Qia W. Reinforcement learning based intelligent maximum power point tracking control for wind energy conversion systems // IEEE Trans. Ind. Electron, 2015. Vol. 62, No. 10, P. 6360–6370.

[26] Xi L., Yu L., Fu Y. Automatic generation control based on deep reinforcement learning with exploration awareness // Proc. CSEE, 2019. Vol. 39, No. 14, P. 4150–4162.

[27] Wang B., Zhou M., Xin B. Analysis of operation cost and wind curtailment using multi-objective unit commitment with battery energy storage // Energy, 2019. Vol. 178, P. 101–114.

[28] Wan Z., Li H., He H. Model-Free Real-Time EV Charging Scheduling Based on Deep Reinforcement Learning // IEEE Transactions on Smart Grid, 2019. Vol. 10, No. 5, P. 5246–5257.

[29] Gao Y., Yang J., Yang M. Deep Reinforcement Learning Based Optimal Schedule for a Battery Swapping Station Considering Uncertainties // IEEE Transactions on Industry Applications, 2020. Vol. 56, No. 5, P. 5775–5784.

[30] Wu Y.K., Tan H.C., Peng J.K. Deep reinforcement learning of energy management with continuous control strategy and traffic information for a series-parallel plug-in hybrid electric bus // Energy, 2019. Vol. 247, P. 454–466.

[31] Бердников Р.Н., Дементьев Ю.А., Моржин Ю.И. Основные положения концепции интеллектуальной электроэнергетической системы россии с активно-адаптивной сетью // Энергия единой сети, 2012. № 4, С. 4–11.

[32] Задорожний А.В., Окороков Р.В. Основные эффекты реализации технологической платформы «Интеллектуальная энергетическая система России // Вестник ИГЭУ. 2013. №2, С. 1–7.

[33] Федеральный закон от 27.12.2019 г. № 471-ФЗ "О внесении изменений в Федеральный закон "Об электроэнергетике"в части развития микрогенерации"

[34] Официальный сайт WattDepot [Электронный ресурс] / wattdepot. Режим доступа: `http://wattdepot.org`, свободный. (дата обращения: 6.05.21)

[35] Официальный сайт Generac Power Systems [Электронный ресурс] / Generac. Режим доступа: `https://www.generac.com/home`, свободный. (дата обращения: 6.05.21)

[36] Официальный сайт openHAB [Электронный ресурс] / openhab. Режим доступа: `https://www.openhab.org`, свободный. (дата обращения: 6.05.21)

[37] Официальный сайт Home Assistant [Электронный ресурс] / home-assistant. Режим доступа: `https://www.home-assistant.io`, свободный. (дата обращения: 6.05.21)

[38] Официальный сайт Spectrum Power [Электронный ресурс] / siemens. Режим доступа: `https://new.siemens.com/ru/ru/produkty/energetika/avtomatizaciya-v-energetike/`

upravlenie-elektricheskimi-setyami/vysokoe-napryazhenie.html, свободный. (дата обращения: 6.05.21)

[39] Официальный сайт EcoStruxure [Электронный ресурс] / se. Режим доступа: https://www.se.com/ru/ru/work/solutions/for-business/electric-utilities/energy-management-system-ems/, свободный. (дата обращения: 6.05.21)

https://www.se.com/ru/ru/work/solutions/for-business/electric-utilities/energy-management-system-ems/

[40] Портал Driven Data [Электронный ресурс] / drivendata. Режим доступа: https://www.drivendata.org/competitions/53/optimize-photovoltaic-battery/page/105/, свободный. (дата обращения: 6.05.21)

[41] Watkins, C. J. C. H. Learning from Delayed Rewards, Ph.D. thesis, Cambridge University, 1989.

[42] Pilarski P. M., Sutton R. S. Between instruction and reward: human-prompted switching // AAAI Fall Symposium Series: Robots Learning Interactively from Human Teachers, 2012. P. 45–52.

[43] Колмогоров А.Н. О представлении непрерывных функций нескольких переменных суперпозициями непрерывных функций меньшего числа переменных // ДАН СССР, 1956. Т. 108, № 2, С. 179–182.

[44] Hecht-Nielsen R. Kolmogorov's mapping neural network existence theorem // IEEE First Annual Int. Conf. on Neural Networks, 1987. Vol. 3, P. 11–13.

[45] Репозиторий исследования [Электронный ресурс] / GitHub. Режим доступа: https://github.com/NastiyaM/Diploma-smart-battery, свободный. (дата обращения: 12.05.21)

[46] Репозиторий ПО [Электронный ресурс] / GitHub. Режим доступа: https://github.com/NastiyaM/Smart-energy-software-package, свободный. (дата обращения: 12.05.21)

[47] Yoshua B. Practical recommendations for gradient-based training of deep architectures // Neural networks: Tricks of the trade, 2012. P. 437–478.