

DOI:

УДК 004.42:004.94

О. Р. Магомедов, С. А. Чернышев

## МУЛЬТИАГЕНТНАЯ СИСТЕМА ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ ДЛЯ МИНИМИЗАЦИИ СТОИМОСТИ ГРУППИРУЕМЫХ ТОВАРОВ

**Аннотация.** В статье приводится решение реальной задачи из предметной области доставки товаров с подробным описанием предложенной архитектуры и принципами работы мультиагентной системы (МАС) поддержки принятия решений для минимизации стоимости отправки заказа посредством группировки товаров по разным посылкам. Рассмотрены преимущества МАС над переборными алгоритмами, особенности спроектированной мультиагентной системы, GUI для анализа, различные модификации работы системы и возможные пути ее улучшения.

*Ключевые слова:* мультиагентная система, система поддержки принятия решений, оптимизация, логистика, группировка товаров.

O.R. Magomedov, S.A. Chernyshev

## MULTI-AGENT DECISION SUPPORT SYSTEM FOR MINIMIZING THE COST OF GROUPED GOODS

**Abstract.** The article describes solution of a real problem from the subject area of goods delivery with a detailed description of the proposed architecture and the principles of multi-agent system (MAS) decision-making support to minimize the cost of sending an order by grouping goods into different parcels. The advantages of MAS over enumerative algorithms, features of the designed multi-agent system, GUI for analysis, different modifications of the system operation and possible ways of its improvement are considered.

*Keywords:* multi-agent system, decision support system, optimization, logistics, goods grouping.

## *Введение*

Согласно данным Главного таможенного управления КНР, товарооборот между Россией и Китаем за 2021 год достиг рекордных показателей - 146,88 млрд долл. [1]. Это дополнительно связано с тем, что в Китае начало открываться всё больше компаний, которые предоставляют посреднические услуги для оптовых покупателей, организуя полный цикл взаимодействия с производителем продукции: выезд на завод, покупка пробных образцов, закупка партии товаров, доставка из Китая в Россию и т. д. Основной заработок таких компаний – процент с продаж и логистические услуги по доставке товара заказчику. Доставлять каждый товар по отдельности – довольно дорогое удовольствие в виду чего осуществляется их объединение с целью уменьшения стоимости отправки всего заказа. При наличии нескольких позиций в заказе, такое объединение товаров по группам не представляет ничего сложного, но, когда речь идет о 30 и более товарах, операторы не в силах справиться с расчетами, а переборные алгоритмы затрачивают на это довольно много времени. При единичных отправках это не критично, в то же самое время озвученные проблемы являются «узким горлышком» при попытке масштабирования бизнеса и увеличения товарооборота.

Перед авторами статьи стояла задача разработать систему поддержки принятия решений, способную находить приемлемые для бизнеса варианты распределения товаров по группам для уменьшения стоимости отправки заказов. Несмотря на то, что переборные алгоритмы будут давать более точный результат, их применение при увеличении количества товаров в заказе не представляется возможным из-за значительного увеличения времени расчетов, даже с учетом распараллеливания. Ввиду этих обстоятельств было принято решение использовать мультиагентный подход, который в последнее время начал снова набирать популярность [2-5], а так как имелись ограничения на использование различных зависимостей в виде фреймворков мультиагентного моделирования, архитектура данной системы и протокол взаимодействия между агентами реализовывались без их использования.

## *Базовые понятия*

Рассмотрим некоторые базовые понятия, связанные с построением мультиагентных систем. Мультиагентная система (МАС) - система, состоящая из одной или более групп агентов, конкурирующих или сотрудничающих друг с другом с целью выполнить общую задачу таким образом, чтобы

увеличить ценность принимаемых решений для всей системы [6]. Под агентом понимается автономный программный объект, способный достигать поставленных целей в условиях неопределенности путем выработки и анализа вариантов принятия решений и согласованного взаимодействия с другими агентами. По уровню поведения агентов можно разделить на проактивные и реактивные. Проактивные агенты анализируют свои знания об окружающей среде и сами проявляют инициативу, вступая в переговоры с другими агентами. Реактивные агенты реагируют на события окружающей среды и непосредственное взаимодействие с ними. Реакцией агента на событие также может быть вступление в переговоры с другими агентами, принятие или отказ от соглашений с агентами. Необходимо отметить, что мультиагентная система, основанная на реактивных агентах, может проявлять свойства МАС, построенной на проактивных агентах [7]. В таких случаях, хоть изначально импульсом к действию агентов и является событие, агенты обретают способность анализировать окружающую среду и предпринимать действия по улучшению состояния системы. Таким образом, мультиагентная система является социальным пространством взаимодействующих друг с другом агентов для достижения общей цели системы.

### *Описание предметной области*

Для того, чтобы лучше понять задачу, рассмотрим предметную область. Целью разрабатываемого алгоритма является минимизация стоимости перевозки заказа, который содержит определенное количество товаров. Каждый товар обладает следующими характеристиками: идентификатор, вес и объем. Товары распределяются по посылкам. Как правило отправить все товары в заказе одной посылкой дороже, чем распределить товары по нескольким посылкам. После того, как товары распределены по посылкам, подсчитывается стоимость каждой посылки. Совокупная стоимость отправки всех посылок и составляет стоимость отправки заказа.

Для расчета цены посылки вычисляется ее плотность путем деления суммы весов всех товаров на сумму объемов всех товаров в посылке. Затем находится тариф по тарифной сетке (табл. 1), который зависит от плотности посылки. Тариф содержит диапазон плотности посылок, для которого он работает, тарифную ставку и тип расчета (по весу или по объему). Как правило, чем больше плотность посылки, тем меньше тарифная ставка. Теперь, если тип расчета в тарифе по весу, то для получения стоимости

посылки производится умножение ее веса на тарифную ставку, в противном случае на тарифную ставку умножается объем.

Таким образом, от группировки товаров по посылкам, зависит цена заказа, то есть задача сводится к распределению товаров по посылкам. Мультиагентный подход хорошо применим к поставленной задаче. Во-первых, четко выделяется критерий эффективности решения задачи — итоговая цена заказа, которую необходимо минимизировать. Во-вторых, цена посылки меняется при перемещениях продуктов из одних посылок в другие, поэтому имеет смысл задать продуктам и группам продуктов социальное поведение и выделить два основных класса агентов — продукт и группа продуктов.

*Таблица 1*

### Пример тарифной сетки

№	Плотность кг/м. куб	Ставка \$	Тип расчета (1 - по весу, 2 - по объему)
1	1000 и более	3,30	2
2	801-999	3,40	2
...			
18	101-110	4,70	2
19	Менее 100..	520,00	1

### *Стек используемых технологий*

Для разработки мультиагентной системы был выбран компилируемый высокоуровневый язык программирования Dart [8]. Язык разработан компанией Google и в основном используется в связке с Flutter [9] для мультиплатформенной разработки клиентских приложений. Тем не менее он неплохо подходит для написания мультиагентных систем, где особенно важна параллельная работа агентов. Dart представляет возможность параллельного выполнения программы посредством изолятов, а наличие фреймворка Flutter позволяет написать мультиплатформенный графический клиент для работы с системой. Использование одного языка программирования для разработки мультиагентной системы и графического интерфейса для взаимодействия с ней позволяет легко использовать сущности и бизнес-логику мультиагентной системы из кода графического интерфейса. Изоляты — это потоки выполнения программы, которые могут

работать конкурентно. При старте программы запускается главный изолят, который выполняет функцию `main`, являющуюся точкой входа программы. Далее главный изолят может создавать и управлять жизненным циклом других изолятов, что и использовалось в архитектуре разработанной системы. Изоляты не разделяют память как потоки в таких языках программирования как C#, Python, Java и для их взаимодействия используются сообщения. При разработке системы возникла необходимость, чтобы все изоляты имели одинаковые знания об окружении системы, и это вызвало необходимость дублирования данных, которого можно было бы избежать, если бы в Dart была разделяемая память между потоками выполнения.

### *Структура разработанной мультиагентной системы*

Главным объектом системы является Environment (среда окружения). Он хранит состояние всей системы, отвечает за создание, удаление агентов, является посредником в переговорах между агентами и фиксирует результаты соглашений агентов, изменяя состояние системы [10]. В среде окружения действуют три типа агентов: Order Agent (агент заказа), Product Group Agent (группа продуктов), Product Agent (одиночный продукт). Product Group представляет из себя посылку, состоящую из более чем одного продукта. Product представляет посылку, состоящую из одного продукта. Order Agent отвечает за хранение информации о заказе, а именно продуктов, групп продуктов и тарифной сетки. Также Order Agent отвечает за обновление продуктов и групп продуктов, когда агенты продуктов и групп продуктов заключают соглашение, выполняя как бы роль брокера.

Жизненный цикл мультиагентной системы можно условно разделить на 3 этапа: инициализация, работа и завершение работы. Последовательно рассмотрим каждый из этапов.

**Инициализация.** Сначала в Environment передаются входные данные: перечень продуктов в заказе и тарифная сетка. Также Environment принимает конфигурацию, с помощью которой возможно задать начальное разбиение продуктов на несколько групп продуктов поровну, либо задать разбиение, при котором все продукты изначально находятся в разных посылках. Далее Environment создает агента заказа, агентов продуктов для одиночных продуктов и агентов групп продуктов для групп продуктов.

**Работа.** Когда Environment инициализирован, начинается этап работы системы, состоящий из последовательных переговорных этапов, называемых эпохами. В ходе эпохи идут переговоры агентов и принимаются соглашения

между ними. Основной концепцией разработанной мультиагентной системы является проведение переговоров между агентами и принятие ими соглашений об изменении системы, но только тех, которые улучшают результат (уменьшается конечная стоимость заказа). Эпохи продолжают создаваться до тех пор, пока не будет достигнуто состояние, которое не улучшается в течение определенного количества эпох, задаваемого конфигурацией Environment (рис. 1).

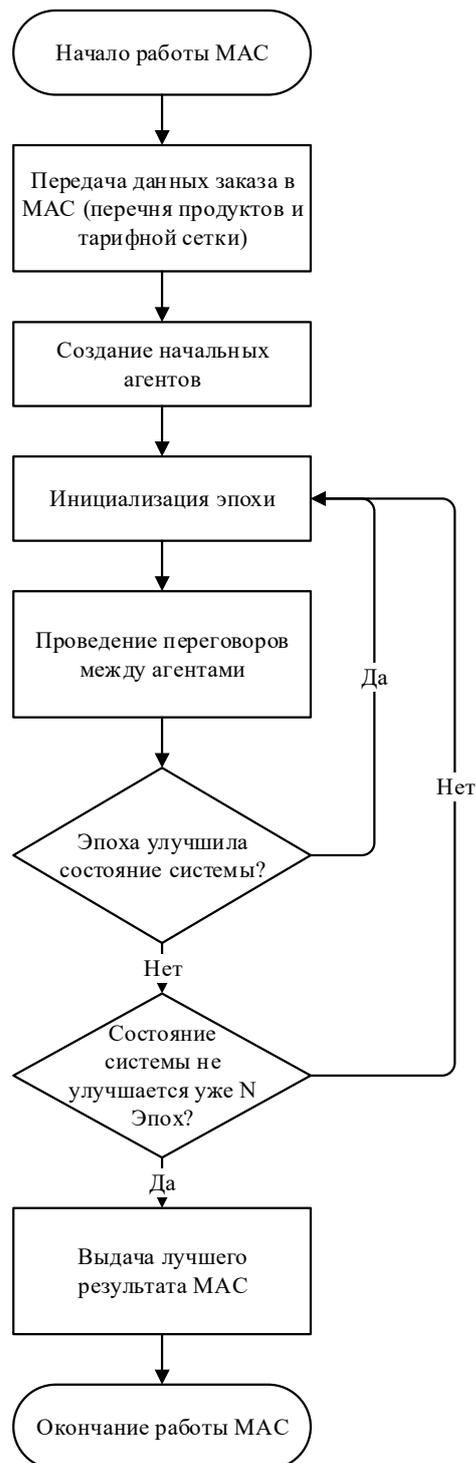


Рисунок 1. Блок схема жизненного цикла МАС

Рассмотрим соглашения, которые могут заключать агенты:

1. Объединение продуктов и групп продуктов. Объединяться могут как продукт с продуктом, продукт с группой продуктов, группа продуктов с продуктом, группа продуктов с группой продуктов. В результате объединения продукта с продуктом создается новая группа продуктов, которая содержит 2 объединившихся продукта. При объединении продукта и группы продуктов продукт добавляется к группе, а в случае объединения двух групп продуктов в первую из них добавляются продукты второй. В сообщении соглашения содержатся идентификаторы двух агентов (продуктов и групп продуктов), заключивших соглашение.
2. Удаление продукта из группы продуктов. Продукт удаляется из группы продуктов и добавляется в Environment. Сообщение о соглашении содержит идентификатор группы продуктов, из которой удаляется продукт, и идентификатор продукта, который удаляется из группы.
3. Удаление продукта из группы продуктов с заменой. Продукт удаляется из группы продуктов и добавляется в Environment и в то же время к группе продуктов добавляется другой продукт, взятый из Environment. Соглашение является составным и содержит два соглашения: удаление продукта из группы продуктов, а также объединение продукта с группой продуктов.
4. Обмен продуктами между группами продуктов. В результате продукт из первой группы перемещается во вторую группу, а продукт из второй группы перемещается в первую группу. Соглашение содержит идентификаторы групп, между которыми происходит обмен, а также продукты, которые требуется обменять.

Данные соглашения не являются единственно возможными. Система разработана таким образом, что поддерживает внедрение других соглашений. Главным требованием для которых является способность уменьшить стоимость заказа.

Рассмотрим жизненный цикл эпохи. В начале каждой эпохи происходит её инициализация, в ходе которой агенты получают знание о состоянии системы. Далее Environment определяет тип эпохи. Было реализовано 4 типа эпох:

1. Regular. В данном типе эпохи агенты продуктов и групп продуктов обмениваются сообщениями и могут прийти к соглашению об объединении, удалении продукта из группы продуктов и удалении продукта из группы продуктов с заменой.

2. OnlyProductsToGroups. На данном шаге отправляют сообщения только продукты и сообщения идут только группам. Агенты могут прийти к соглашению об объединении продукта с группой продуктов, либо об удалении продукта из группы с заменой.
3. Exchange. На данном шаге производится переговоры по поводу обмена продуктами между группами.
4. GroupDisbandment. Шаг, на котором происходит расформирование одной из групп продуктов по заданному признаку (наивысшей плотности, наименьшей плотности, наивысшему весу, наименьшему весу, наивысшей цене, наименьшей цене), после чего эпоха заканчивается. Это единственный механизм в системе, который не соответствует концепции применения только тех изменений, которые уменьшают стоимость заказа. Применение данного механизма осуществляется только в случае, когда состояние системы перестает улучшаться в течение нескольких эпох. Расформирование одной из групп позволяет активизировать переговорный процесс, тем самым повышая вероятность нахождения более оптимального решения, хоть и изначально ухудшает состояние системы. Количество возможных расформирований на протяжении работы системы определяется соответствующим параметром конфигурации.

После определения типа эпохи Environment отправляет агентам продуктов и групп продуктов сообщение о начале переговоров, которое является сигналом к действию и содержит дополнительную информацию как номер эпохи и тип эпохи. В зависимости от типа эпохи агенты выстраивают свое поведение. Если агент хочет объединиться с каким-либо из других агентов, то он отправляет сообщение в Environment. Environment рассылает сообщение агентам. Агенты проверяют, улучшит ли операция положение системы и возвращают соглашение в случае, если стоимость заказа уменьшается, иначе агенты посылают сообщение об отказе. При получении соглашения Environment сразу применяет соглашение, изменяя состояние системы. При этом Environment проверяет допустимость соглашений. Некоторые соглашения могут быть недействительными. Если агент продукта А отправил сообщение агентам продуктов Б и В на объединение и оба агента согласились, но Б прислал сообщение первым, то агенты А и Б объединяются, а агенты А и В не объединяются из-за того, что при вынесении соглашения В не знал, что агенты А и Б будут объединены, вследствие чего соглашение между агентами А и В объявляется недействительным. Таким образом, при применении соглашения его стороны объявляются неактивными до конца эпохи и дальнейшие соглашения, в

которых участвует хотя бы один неактивный агент, будут отменены. Применение соглашения сразу после его получения Environment имеет недостаток. В примере выше соглашение между агентами А и В могло оказаться выгоднее соглашения А и Б, но в любом случае применяется соглашение между А и Б, так как оно пришло в Environment первым. Тем не менее в ходе испытаний оказывалось, что в большинстве случаев такой недостаток компенсируется в следующих эпохах. Так в примере выше агенты А и Б при объединении образуют агента группы продуктов, который содержит продукты А и Б. Но если группа А и В лучше для системы, чем группа А и Б, то на типе эпохи exchange произойдет удаление продукта Б из группы и замена его на продукт В. В результате получаем состояние идентичное тому, которое бы получилось при изначальном объединении продуктов А и В в группу.

Такой же принцип используется и при обмене продуктами между группами продуктов. Агент группы рассылает сообщение остальным группам на обмен, остальные группы проверяют выгодность обмена и формируют либо соглашения, либо отказы. Применяются только те соглашения, в которых оба агента активны, то есть не участвовали в других соглашениях на протяжении данной эпохи. После каждой эпохи происходит фиксация результата эпохи с сохранением информации о составе заказа и его цене.

**Завершение работы.** После того, как достигнуто состояние, которое не улучшается в течение определенного количества эпох, мультиагентная система завершает работу и возвращает наилучший результат.

### *Результаты и обсуждение*

Главным преимуществом мультиагентной системы по сравнению с переборными алгоритмами является скорость выполнения работы. Следует отметить, что при увеличении количества продуктов в заказе скорость выполнения их группировки переборными алгоритмами падает в несколько раз. Были проведены замеры времени выполнения переборного алгоритма и мультиагентной системы на выборках входных данных с постепенным увеличением количества товаров в заказе в формате часы:минуты:секунды.миллисекунды (табл. 2).

Таблица 2

### Сравнение времени работы переборного алгоритма с МАС

Количество продуктов	Время группировки (мультиагентная система)	Время группировки (переборный алгоритм)
----------------------	--	---

5	0:00:04.880135	0:00:00.112994
6	0:00:04.525058	0:00:00.174604
7	0:00:04.947600	0:00:00.217338
8	0:00:04.455448	0:00:01.408652
9	0:00:06.053171	0:00:21.815098
10	0:00:09.406804	0:08:36.785325
30	0:00:17.780205	-
50	0:00:43.907667	-
70	0:01:03.459164	-
100	0:02:06.676889	-

Переборный алгоритм, который учитывает при переборе все комбинации продуктов, не справляется при заказе, большем 10 продуктов. В то время как время выполнения мультиагентной системы при увеличении продуктов пропорционально увеличению продуктов. При этом во всех проведенных испытаниях мультиагентная система показала себя наравне с переборным алгоритмом, то есть достигла самого оптимального результата. Однако при увеличении количества продуктов в заказе повышается вероятность того, что мультиагентная система составит не самую лучшую группировку продуктов. Данное явление прослеживается из-за вероятностного фактора, присутствующего в мультиагентной системе, который обусловлен тем, что порядок переговоров между агентами продуктов и групп продуктов не определен и применяются те соглашения между агентами, которые первыми попали в Environment. При росте продуктов значительно повышается количество возможных линий переговоров, вследствие чего повышается шанс того, что переговоры приведут к разным результатам. Для того, чтобы максимально улучшить результат группировки заказа, программа последовательно запускает много мультиагентных систем на одном и том же заказе с разной конфигурацией и затем выбирает результат с минимальной итоговой ценой заказа. Также возможным способом улучшения мультиагентной системы является добавление новых соглашений между агентами. Система хорошо справляется с задачей группировки товаров, хотя определенно имеет потенциал к дальнейшему улучшению.

Стоит отметить, что МАС разрабатывалась как полностью автономный сервис, так и как система поддержки принятия решений. Важным дополнением к мультиагентной системе является графический интерфейс пользователя, который имеет двойную ценность. Во-первых, графический интерфейс упрощает исследование поведения системы для ее последующего улучшения. Во-вторых, за счет упрощения взаимодействия с мультиагентной системой становится возможным ввести роль оператора заказов, который

будет работать с системой: выбирать конфигурацию и контролировать оптимальность результатов с возможностью их ручной модификации.

### *GUI для работы с MAC*

Разработанный графический интерфейс позволяет загрузить необходимый заказ из JSON файла. На экране отображаются продукты, группы продуктов и тарифная сетка заказа (рис. 2). Интерфейс позволяет запустить MAC на заданном заказе и далее оценить полученные результаты (получившиеся продукты и группы, время выполнения MAC). Для ручного редактирования продуктов и групп продуктов реализована возможность перетаскивать их с помощью мыши, тем самым добавляя продукт к группе продуктов или соединяя группы продуктов [11]. Для более удобного просмотра продуктов и групп добавлена возможность их сортировки по различным критериям (цене, весу, объему, плотности).

The screenshot displays a web-based interface for managing MAC configurations. It is divided into several sections:

- Products:** A card for product 'VM-134-0511-1' with details: Price: 55.44, Weight: 1980.0, Volume: 9.0, Density: 220.0.
- Product groups:** Two cards for 'Group-2' and 'Group-3'. Group-2 details: Price: 3293.68, Weight: 126680.0, Volume: 422.0, Density: 300.19, Items: 5. Group-3 details: Price: 631.4, Weight: 22550.0, Volume: 108.0, Density: 208.8, Items: 4.
- Prices:** A table with columns 'Id', 'Range', and 'Price'.

Id	Range	Price
0	0 - 100	4.5
1	100 - 110	0.038
2	110 - 120	0.037
3	120 - 130	0.036
4	130 - 140	0.035
5	140 - 150	0.034
6	150 - 160	0.033
7	160 - 170	0.032
8	170 - 180	0.031
9	180 - 190	0.03
10	190 - 200	0.029
11	200 - 250	0.028
12	250 - 300	0.027
13	300 - 350	0.026
14	350 - 400	0.025
- Control Panel:** Includes buttons for 'Upload from file', 'Specify system path' (with a file path: C:\Users\smart\Desktop\dart\_isolate\_age nt\dart\_isolate\_agent\bin\build\dev), 'View prices list', 'Run configuration' (with settings: Iterations: 25, Runners: 2, Max groups: 5), 'Start system', and 'Run results' (with summary: Best price: 3980.52\$, Current price: 3980.52\$, Time to complete: 16.23s).

Рисунок 2. GUI для работы с MAC

### *Заключение*

Мультиагентный подход позволил разработать систему, способную быстро решать довольно сложную оптимизационную задачу в сфере логистики, с которой плохо справлялись переборные алгоритмы,

рассматривающие все возможные решения задачи. Мультиагентная система дала возможность использовать более высокий уровень абстракции — вместо алгоритма, который отбирает и оценивает эффективность решений, была создана среда и выделены агенты, которые взаимодействуя по определенному протоколу, в ходе переговоров улучшают систему по оптимизационному критерию, заключая соглашения. Также в ходе разработки и тестирования решения было выявлено важное свойство мультиагентных систем: при правильном подборе соглашений между агентами система восстанавливается и продолжает улучшаться, достигая необходимого результата, даже после принятия неудачных соглашений. Система почти в 100% случаев не уступает переборным алгоритмам и показывает хорошие результаты на заказах с больших количеством продуктов. Узким местом таких систем является нечестная конкуренция между агентами, ведь применяются те соглашения, которые пришли в систему первыми и отклоняются соглашения, которые пришли позже и являются несовместимыми с предыдущими. Вследствие этого возможным улучшением систем является создание механизма приема и оценки входящих соглашений с целью отбора и применения лучших. Также улучшения результата системы можно попробовать достичь сменой реактивной архитектуры агентов на проактивную [12]. И третьим возможным улучшением является настройка конфигурации системы. Для лучшего анализа системы к ней можно подключить статистический модуль, который будет собирать данные о работе мультиагентной системы. Собранные данные можно будет использовать и как данные для изучения человеком, и как выборку данных для обучения нейронной сети, ответственной за выбор конфигурации системы в зависимости от входящего заказа.

Также следует отметить высокую конфигурируемость предлагаемого решения, что придает гибкость, необходимую для проведения экспериментов с целью анализа поведения системы и достижения ее наилучшей работы. Структура МАС вводит некоторые ограничения, но вместе с этим упрощает поведение агентов и оставляет возможность внедрения новых способов переговоров и соглашений между ними.

#### Литература

1. *Болдова К.* Товарооборот России и Китая в 2021 году побил рекорд [Электронный ресурс] URL: <https://journal.open-broker.ru/radar/rekord-tovarooborota-rossii-i-kitaya/> (дата обращения: 14.08.2022).

2. *M. Elgamal, E. Akram, J.M. Guerrero* An adaptive multiagent control system for autonomous economic operation and resilience assurance in a hybrid-energy islanded microgrid // *International Journal of Electrical Power and Energy Systems*, v. 140, 2022, DOI: 10.1016/j.ijepes.2022.108070
3. *W. Xiangyu, L. Weiming, W. Quanwei, L. Shihua* A Modular Optimal Formation Control Scheme of Multiagent Systems With Application to Multiple Mobile Robots // *IEEE Transactions on Industrial Electronics*, v.69, № 9 (2022), p. 9331 – 9341, DOI: 10.1109/TIE.2021.3114732
4. *Кузин. А. Ю.* Мультиагентная система управления распределенной энергосистемой / А. Ю. Кузин, Д. В. Лукичев, Г. Л. Демидова // *Вестник Государственного университета морского и речного флота имени адмирала С. О. Макарова*. — 2020. — № 5. — С. 945-954, DOI: 10.21821/2309-5180-2020-12-5-945-954.
5. *Клименко. Д. Н.* Мультиагентная система управления фондовым рынком. *Инновации и инвестиции*. — 2021 — №5. — С. 99-101.
6. *Г. А. Ржевский, П. О. Скобелев* Как управлять сложными системами? Мультиагентные технологии для создания интеллектуальных систем управления предприятиями : перевод с английского. // Самара : Офорт, 2015. - 290 с.
7. *Kendall, E.A., Murali Krishna, P.V., Pathak, C.V., Suresh, C.B.*: Patterns of intelligent and mobile agents // *Proceedings of the 2nd International Conference on Autonomous Agents*, pp. 92–99. ACM (1998)
8. Dart [Электронный ресурс] URL: <https://dart.dev/> (дата обращения: 1.09.2022).
9. Flutter [Электронный ресурс] URL: <https://flutter.dev/> (дата обращения: 1.09.2022).
10. *Чернышев С.А., Магомедов О.Р.* Мультиагентная система поддержки принятия решений для минимизации стоимости группируемых товаров на основе реактивных агентов // Свидетельство о государственной регистрации программы для ЭВМ № 2022619068 от 19 мая 2022 г.
11. *Чернышев С.А., Магомедов О.Р.* Мультиагентная система поддержки принятия решений для минимизации стоимости группируемых товаров. Графический пользовательский интерфейс // Свидетельство о государственной регистрации программы для ЭВМ № 2022619625 от 24 мая 2022 г.
12. *Чернышев С.А., Магомедов О.Р.* Мультиагентная система поддержки принятия решений для минимизации стоимости группируемых товаров на основе проактивных агентов // Свидетельство о государственной регистрации программы для ЭВМ № 2022619067 от 19 мая 2022 г.

---

## References

1. *Boldova K.* Tovaroorot Rossii i Kitaya v 2021 godu pobil rekord [Trade turnover between Russia and China hit a record in 2021]. Available at: <https://journal.open-broker.ru/radar/record-tovaroorota-rossii-i-kitaya/> (accessed: 14.08.2022).
2. *M. Elgamal, E. Akram, J.M. Guerrero* An adaptive multiagent control system for autonomous economic operation and resilience assurance in a hybrid-energy islanded microgrid // *International Journal of Electrical Power and Energy Systems*, v. 140, 2022, DOI: 10.1016/j.ijepes.2022.108070
3. *W. Xiangyu, L. Weiming, W. Quanwei, L. Shihua* A Modular Optimal Formation Control Scheme of Multiagent Systems With Application to Multiple Mobile Robots // *IEEE Transactions on Industrial Electronics*, v.69, № 9 (2022), p. 9331 – 9341, DOI: 10.1109/TIE.2021.3114732
4. *Kuzin, Aleksey Yu., Dmitry V. Lukichev, and Galina L. Demidova.* “Multi-agent system for distributed energy system control.” *Vestnik Gosudarstvennogo universiteta morskogo i rechnogo flota imeni admirala S. O. Makarova* 12.5 (2020): 945–954. DOI: 10.21821/2309-5180-2020-12-5-945-954. (in Russian)
5. *Klimenko D.N.* Multi-agent stock market management system // *Innovations and Investments magazine*, №5 (2021), p. 99-101 (in Russian)
6. *George Rzevski, Petr Skobelev* *Managing Complexity* // WIT Press, 2014, DOI: 10.1002/inst.12050
7. *Kendall, E.A., Murali Krishna, P.V., Pathak, C.V., Suresh, C.B.:* Patterns of intelligent and mobile agents // *Proceedings of the 2nd International Conference on Autonomous Agents*, pp. 92–99. ACM (1998)
8. Dart. Available at: <https://dart.dev/> (accessed: 1.09.2022).
9. Flutter. Available at: <https://flutter.dev/> (accessed: 1.09.2022).
10. *Chernyshev S.A., Magomedov O.R.* Multi-agent decision support system for minimizing the cost of grouped goods based on reactive agents // State Registration Certificate for Computer Software No. 2022619068 dated May 19, 2022.
11. *Chernyshev S.A., Magomedov O.R.* Multi-agent decision-making support system for minimization of grouped goods cost. Graphic User Interface // State Registration Certificate № 2022619625 of May 24, 2022.
12. *Chernyshev S.A., Magomedov O.R.* Multi-agent decision support system for minimizing the cost of grouped goods based on proactive agents // State Registration Certificate for Computer Software № 2022619067 dated May 19, 2022.

---

**Магомедов Олег Русланович**

Бакалавр

Санкт-Петербургский государственный экономический университет, Санкт-Петербург.

Сфера научных интересов: имитационное моделирование, мультиагентные системы, системы поддержки принятия решений, разработка распределенных систем. Автор 6 опубликованных научных работ.

Электронный адрес: [smart7even@yandex.ru](mailto:smart7even@yandex.ru)

**Чернышев Станислав Андреевич**

кандидат технических наук, доцент кафедры информатики. Санкт-Петербургский государственный экономический университет;

доцент кафедры прикладной информатики. Санкт-Петербургский государственный университет аэрокосмического приборостроения, Санкт-Петербург.

Сфера научных интересов: имитационное моделирование, мультиагентные системы, системы поддержки принятия решений, разработка распределенных систем. Автор более 40 опубликованных научных работ.

Электронный адрес: [chernyshev.s.a@bk.ru](mailto:chernyshev.s.a@bk.ru)