

Министерство образования и науки Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ»**

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ**

**«Глубокие самообучающиеся агенты для мультиагентной системы
маршрутизации»**

Автор: Мухутдинов Дмитрий Вадимович _____

Направление подготовки (специальность): 01.03.02 Прикладная математика и
информатика

Квалификация: Бакалавр

Руководитель: Фильченков А.А., к.ф.-м.н _____

К защите допустить

Зав. кафедрой Васильев В.Н., докт. техн. наук, проф. _____

« ____ » _____ 20 ____ г.

Санкт-Петербург, 2017 г.

Студент Мухутдинов Д.В. **Группа** М3438 **Кафедра** компьютерных технологий **Факультет** информационных технологий и программирования

Направленность (профиль), специализация Математические модели и алгоритмы разработки программного обеспечения

Консультанты:

а) Вяткин В.В., PhD, Luleå University of Technology _____

Квалификационная работа выполнена с оценкой _____

Дата защиты « ____ » _____ 20 ____ г.

Секретарь ГЭК _____

Листов хранения _____

Демонстрационных материалов/Чертежей хранения _____

**«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ»**

УТВЕРЖДАЮ

Зав. каф. компьютерных технологий

докт. техн. наук, проф.

_____ Васильев В.Н.

« ____ » _____ 20 ____ г.

**ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ**

Студент Мухутдинов Д.В. **Группа** М3438 **Кафедра** компьютерных технологий **Факультет** информационных технологий и программирования
Руководитель Фильченков Андрей Александрович, к.ф.-м.н, доцент кафедры КТ

1 Наименование темы: Глубокие самообучающиеся агенты для мультиагентной системы маршрутизации

Направление подготовки (специальность): 01.03.02 Прикладная математика и информатика

Направленность (профиль): Математические модели и алгоритмы разработки программного обеспечения

Квалификация: Бакалавр

2 Срок сдачи студентом законченной работы: «31» мая 2017 г.

3 Техническое задание и исходные данные к работе.

Требуется разработать алгоритм решения обобщенной задачи маршрутизации, основанный на идее мультиагентного обучения нейронных сетей с подкреплением. Требуется применить метод к решению конкретных задач маршрутизации, таких как маршрутизация сетевых пакетов и управление конвейерной системой транспортировки багажа, и сравнить его с существующими алгоритмами.

4 Содержание выпускной квалификационной работы (перечень подлежащих разработке вопросов)

- а) Обзор предметной области
- б) Реализация среды для эмуляции задачи сетевой маршрутизации
- в) Реализация среды для эмуляции конвейерной системы транспортировки багажа
- г) Разработка алгоритма маршрутизации на основе мультиагентного обучения нейронных сетей с подкреплением
- д) Реализация существующих алгоритмов решения задач маршрутизации
- е) Проведение экспериментов, интерпретация результатов

5 Перечень графического материала (с указанием обязательного материала)

Не предусмотрено

6 Исходные материалы и пособия

- а) Richard S. Sutton and Andrew G. Barto. Reinforcement Learning: An Introduction. The MIT Press, 2012
- б) Mnih et al. Human-level control through deep reinforcement learning. Nature, 518(7540):529–533, 2015.

7 Календарный план

№№ пп.	Наименование этапов выпускной квалификационной работы	Срок выполнения этапов работы	Отметка о выполнении, подпись руков.
1	Ознакомление с предметной областью	11.2016	
2	Чтение статей, посвященных алгоритмам маршрутизации	12.2016	
3	Чтение статей, посвященных задаче обучения с подкреплением	01.2017	
4	Разработка сред для эмуляции задач маршрутизации	03.2017	
5	Разработка алгоритма маршрутизации, реализация существующих алгоритмов	04.2017	
6	Проведение экспериментов, написание пояснительной записки	05.2017	

8 Дата выдачи задания: «01» сентября 2016 г.

Руководитель _____

Задание принял к исполнению _____ «01» сентября 2016 г.

Министерство образования и науки Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
**«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ»**

**АННОТАЦИЯ
ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ**

Студент: Мухутдинов Дмитрий Вадимович

Наименование темы работы: Глубокие самообучающиеся агенты для мультиагентной системы маршрутизации

Наименование организации, где выполнена работа: Университет ИТМО

ХАРАКТЕРИСТИКА ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

1 Цель исследования: Разработка распределенного алгоритма решения задачи маршрутизации, подходящего для эффективного применения в различных условиях, включая киберфизические системы.

2 Задачи, решаемые в работе:

- а) Формальная постановка обобщенной задачи маршрутизации.
- б) Разработка алгоритма решения поставленной задачи.
- в) Сравнение работы алгоритма с существующими решениями на практике.

3 Число источников, использованных при составлении обзора: _____

4 Полное число источников, использованных в работе: 10

5 В том числе источников по годам

Отечественных			Иностранных		
Последние 5 лет	От 5 до 10 лет	Более 10 лет	Последние 5 лет	От 5 до 10 лет	Более 10 лет

6 Использование информационных ресурсов Internet: _____

7 Использование современных пакетов компьютерных программ и технологий:

Для решения задачи использовался язык программирования Python. При разработке систем эмуляции работы компьютерной сети и системы управления багажными конвейерами была использована библиотека Thespian. Для реализации и обучения нейронных сетей были использованы библиотеки TensorFlow и Keras.

8 Краткая характеристика полученных результатов: Разработана формальная постановка обобщенной задачи маршрутизации в терминах мультиагентного обучения с подкреплением. Разработан алгоритм решения обобщенной задачи маршрутизации. Разработанный алгоритм по результатам экспериментов на практике не уступает современным специализированным алгоритмам решения задач сетевого роутинга и управления конвейерной системой, но благодаря обобщенной постановке задачи и применению нейронных сетей имеет широкий потенциал применения для решения

задач, сводимых к задаче маршрутизации, в сложных условиях (таких, как киберфизические системы).

9 Гранты, полученные при выполнении работы: При выполнении работы грантов получено не было.

10 Наличие публикаций и выступлений на конференциях по теме работы: По теме данной работы публикаций и выступлений на конференциях нет.

Выпускник: Мухутдинов Д.В. _____

Руководитель: Фильченков А.А. _____

« ____ » _____ 20 ____ г.

ОГЛАВЛЕНИЕ

[illegible]

ВВЕДЕНИЕ

Задача пакетной маршрутизации - это задача поиска кратчайшего пути в графе в условиях, когда за каждый узел графа отвечает отдельный вычислительный процесс. Это означает, что каждый отдельный узел графа должен принять решение о том, какому из соседей следует отправить очередной пакет, чтобы тот достиг пункта назначения как можно быстрее.

Задача пакетной маршрутизации впервые обрела актуальность с появлением компьютерных сетей. Первые алгоритмы сетевой маршрутизации появились в процессе разработки сети ARPANet. Именно тогда были изобретены такие подходы к пакетной маршрутизации, как distance-vector[1] и link-state[2], которые и по сей день лежат в основе таких стандартных и широко применяемых алгоритмов сетевой маршрутизации, как Routing Information Protocol (RIP)[3] и Open Shortest Path First (OSPF)[4].

Оба этих подхода основаны на идее вычисления кратчайшего пути между текущим узлом сети и пунктом назначения пакета. Однако существуют и другие подходы к решению задачи маршрутизации, основанные на идее обучения с подкреплением (reinforcement learning). Первым таким алгоритмом стал алгоритм Q-routing[5], основанный на методе Q-learning[6]. Этот алгоритм, как и его модификации[7, 8], благодаря обучению с подкреплением оказался способен лучше адаптироваться к изменениям в интенсивности сетевого трафика, чем алгоритмы, основанные на вычислении кратчайшего пути, но из-за использования большего количества служебных сообщений применение таких алгоритмов в реальных сетях ограничено.

Но стоит отметить, что задача маршрутизации встречается не только в компьютерных сетях, но также и в более сложных средах, таких как киберфизические системы. Примером такой задачи, частично решаемой с помощью алгоритмов маршрутизации, является управление конвейерной системой (в частности, системой распределения багажа в аэропорту). Задача управления такой системой в большинстве случаев решается с помощью централизованных алгоритмов, и децентрализованное решение (основанное на алгоритма Беллмана-Форда) было пред-

ложено совсем недавно[9]. Такая задача отличается от задачи сетевого роутинга, с одной стороны, тем, что служебные сообщения и целевые сообщения являются разными сущностями (“целевое сообщение” – это чемодан, а служебные сообщения являются цифровыми), и служебные сообщения передаются мгновенно по сравнению с целевыми. С другой стороны, состояние каждого конвейера и всей системы в целом задается большим количеством параметров – такими как скорости конвейеров, положение, количество и масса чемоданов на каждом конвейере, и так далее. С третьей стороны, желательно, чтобы система оптимизировала не только скорость доставки чемоданов до точки назначения, но и, к примеру, собственное энергопотребление.

Первое обстоятельство снимает технические ограничения на количество служебных сообщений, что позволяет в полной мере применять алгоритмы на основе обучения с подкреплением. Второе и третье обстоятельства усложняют написание оптимального детерминированного алгоритма и наталкивают на идею реализации приближенного решения, например, с использованием нейронных сетей.

В настоящее время в области обучения с подкреплением с применением нейронных сетей достигнуты впечатляющие успехи. Всплеск активности в этой области произошел после выхода статьи команды Google DeepMind об обучении глубокой сверточной нейронной сети игре на консоли Atari 2600[10]. Применению полученной модели к различным задачам в различных условиях посвящено множество исследований, часть из них посвящена проблеме мультиагентного обучения с подкреплением (multi-agent reinforcement learning). Так как задачу маршрутизации можно сформулировать как задачу мультиагентного обучения с подкреплением, имеет смысл применить данные наработки для ее решения.

В данной работе будет предложен алгоритм маршрутизации, основанный на методе Q-routing, но использующий нейронную сеть в качестве обучающегося агента. На данный момент не существует алгоритма маршрутизации, построенного по такому принципу.

В главе 1 будет сформулирована обобщенная постановка задачи маршрутизации в терминах мультиагентного обучения с подкреплением.

ем. Будут рассмотрены существующие алгоритмы маршрутизации, их сильные и слабые стороны. Также будут рассмотрены существующие методы обучения нейронных сетей с подкреплением, в том числе в мультиагентном случае.

В главе 2 будет рассмотрен предложенный алгоритм и обоснованы решения, принятые в ходе его разработки.

В главе 3 будут приведены экспериментальные результаты работы алгоритма для задач пакетной маршрутизации в компьютерной сети и управления системой багажных конвейеров. Будут приведены результаты работы в условиях неравномерной нагрузки на сеть (или конвейерную систему) и изменения топологии сети. Также будет проведено сравнение с существующими алгоритмами маршрутизации.

ГЛАВА 1. ОБЗОР ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. Таблицы

Таблица 1 – Таблица умножения (фрагмент)

–	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
2	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34
3	3	6	9	12	15	18	21	24	27	30	33	36	39	42	45	48	51
4	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68

Есть еще такое окружение `tabu`, его можно аккуратно растянуть на всю страницу. Приведем пример (таблица 2).

Таблица 2 – Таблица умножения с помощью `tabu` (фрагмент)

–	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
2	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34
3	3	6	9	12	15	18	21	24	27	30	33	36	39	42	45	48	51
4	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68

1.2. Рисунки

Пример рисунка (с помощью `TikZ`) приведен на рисунке 1. Под `pdflatex` можно также использовать `*.jpg`, `*.png` и даже `*.pdf`, под `latex` можно использовать `Metapost`. Последний можно использовать и под `pdflatex`, для чего в стилевике продекларированы номера картинок от 1 до 20.

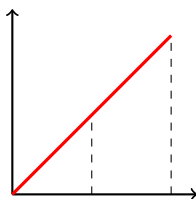


Рисунок 1 – Пример рисунка

1.3. Листинги

В работах студентов кафедры «Компьютерные технологии» часто встречаются листинги. Листинги бывают двух основных видов — исходный код и псевдокод. Первый оформляется с помощью окружения

`lstlisting` из пакета `listings`, который уже включается в стилевике и немного настроен. Пример Hello World на Java приведен на листинге 1.

Листинг 1 – Пример исходного кода на Java

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, world!");
    }
}
```

1.3.1. Тест

Псевдокод можно оформлять с помощью разных пакетов. В данном стилевике включается пакет `algorithmicx`. Сам по себе он не генерирует флоатов, поэтому для них используется пакет `algorithm`. Пример их совместного использования приведен на листинге 2. Обратите внимание, что флоаты разные, а нумерация — общая!

Листинг 2 – Пример псевдокода

```
function IsPrime( $N$ )
  for  $t \leftarrow [2; \lfloor \sqrt{N} \rfloor]$  do
    if  $N \bmod t = 0$  then
      return false
    end if
  end for
  return true
end function
```

Наконец, листинги из `listings` тоже можно подвешивать с помощью `algorithm`, пример на листинге 3.

Листинг 3 – Исходный код и флоат `algorithm`

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, world!");
    }
}
```

ГЛАВА 2. ПРОВЕРКА СКВОЗНОЙ НУМЕРАЦИИ

Листинг 4 должен иметь номер 4.

Листинг 4 – Исходный код и флюид algorithm

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, world!");
    }
}
```

Рисунок 2 должен иметь номер 2.

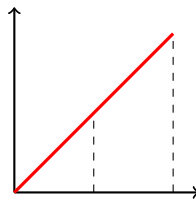


Рисунок 2 – Пример рисунка

Таблица 3 должна иметь номер 3.

Таблица 3 – Таблица умножения с помощью tabu (фрагмент)

–	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
2	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34
3	3	6	9	12	15	18	21	24	27	30	33	36	39	42	45	48	51
4	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68

Выводы по главе 2

В конце каждой главы желательно делать выводы. Вывод по данной главе — нумерация работает корректно, ура!

ЗАКЛЮЧЕНИЕ

В данном разделе размещается заключение.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 *McQuillan J. M., Walden D. C.* The ARPA network design decisions // *Computer Networks*. — 1977. — 1(5). — С. 243–289.
- 2 *McQuillan J. M., Richer I., Rosen E. C.* The New Routing Algorithm for the ARPANet // *IEEE Trans. on Comm.* — 1980. — 28(5). — С. 711–719.
- 3 *Hendrick C.* Routing Information Protocol: RFC. — Июнь 1988. — № 1058.
- 4 *Moy J.* OSPF Version 2: RFC. — Апр. 1998. — № 1058.
- 5 *Boyan J. A., Littman M. L.* Packet routing in dynamically changing networks: a reinforcement learning approach // *Advances in Neural Information Processing Systems*. — 1994. — No. 6. — P. 671–678.
- 6 *Watking C.* Learning from Delayed Rewards: дис. ... канд. / Watking C. — Cambridge : King's College, 1989.
- 7 *Choi S. P. . M., Yeung D.-Y.* Predictive Q-Routing: A Memory-based Reinforcement Learning Approach to Adaptive Traffic Control // *Advances in Neural Information Processing Systems*. — 1996. — No. 8. — P. 945–951.
- 8 *Kumar S., Miikkulainen R.* Dual reinforcement Q-routing: An on-line adaptive routing algorithm // *Artificial Neural Networks in Engineering*. — 1997. — No. 7. — P. 231–238.
- 9 *Yan J., Vyatkin V.* Distributed Software Architecture Enabling Peer to Peer Communicating Controllers // *IEEE Transactions on Industrial Informatics*. — 2013. — 9(4). — P. 2200–2209.
- 10 Human-level control through deep reinforcement learning / V. Mnih [et al.] // *Nature*. — 2015. — No. 518. — P. 529–533.

ПРИЛОЖЕНИЕ А. ПРИМЕР ПРИЛОЖЕНИЯ

В приложениях рисунки, таблицы и другие подобные элементы нумеруются по приложениям с соответствующим префиксом. Проверим это.

Листинг А.1 должен иметь номер А.1.

Листинг А.1 – Исходный код и флюид `algorithm`

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, world!");
    }
}
```

Рисунок А.1 должен иметь номер А.1.

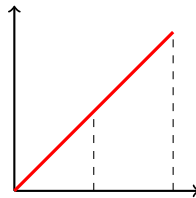


Рисунок А.1 – Пример рисунка

Таблица А.1 должна иметь номер А.1.

Таблица А.1 – Таблица умножения с помощью `tabu` (фрагмент)

–	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
2	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34
3	3	6	9	12	15	18	21	24	27	30	33	36	39	42	45	48	51
4	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68

Заодно проверим нумерованные и нумерованные перечисления. Нумерованные:

- пункт А;
- пункт Б;
- пункт В.

Нумерованные списки нескольких уровней:

- а) первый элемент;
- б) второй элемент с подэлементами:

- 1) первый подэлемент;
 - 2) второй подэлемент;
 - 3) третий подэлемент.
- в) третий элемент;
 - г) четвертый элемент;
 - д) пятый элемент;
 - е) шестой элемент;
 - ж) седьмой элемент;
 - и) восьмой элемент;
 - к) девятый элемент;
 - л) десятый элемент.

**ПРИЛОЖЕНИЕ Б. ЕЩЕ ОДИН ПРИМЕР ПРИЛОЖЕНИЯ С
НЕИМОВЕРНО ДЛИННЮЩИМ НАЗВАНИЕМ ДЛЯ ТЕСТИРОВАНИЯ
ПЕРЕНОСОВ**

Проверим на примере таблиц, что нумерация в приложениях — по приложениям. Таблица Б.1 должна иметь номер Б.1.

Таблица Б.1 – Таблица умножения с помощью `tabu` (фрагмент)

–	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
2	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34
3	3	6	9	12	15	18	21	24	27	30	33	36	39	42	45	48	51
4	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68