

УДК 004.8

**MULTIAGENT REINFORCEMENT DEEP LEARNING ROUTING ALGORITHM  
DQN-ROUTING GENERALIZATION APPROACH**

**Барбахан И. (National Research University ITMO)**

**Фильченков Андрей Александрович, доцент факультета информационных технологий  
и программирования.**

(National Research University ITMO)

This work is about enhancing the generalization capability of DQN-Routing algorithm when it is applied to a new topology, using an approach based on domain adaptation technique.

**Введение.**

DQN-Routing is an algorithm based on deep reinforcement learning networks combined with link-state protocol and preliminary supervised learning. The algorithm is designed to solve the problem of routing in a network (Graph) and originally was tested using manually constructed simulation models of computer network and baggage handling system. It outperforms state-of-the-art routing algorithms. However, in its current situation, the only way to apply this algorithm on a new topology (network/graph) is to re-run the whole training process.

This is not time efficient due to the structure of the algorithm in which every node is an agent that needs to be trained from scratch and it will take a lot of computational time.

Since DQN-Routing is a novel method, there has not been any previous work done towards solving the problem of generalization, and this work will construct the base towards enhancing the overall performance of the algorithm.

**Основная часть.**

The proposed solution is a Domain Adaptation solution, which basically is a sub-category of transfer Learning. In general domain adaptation is used when an algorithm is performing well on a task, and there is a need to transfer that knowledge to a new different and yet similar task, the first task is called "Source Domain" and the Second task is called "Target Domain".

Transferring the knowledge means that there is no need to re-train and re-run the whole procedure for the new task, but to use the knowledge gained from the source to solve the task of the target.

For DQN-algorithm, what we have is different topologies, which means different layout of the graph and then new agents in new nodes, this basically is a domain shift since the task is the same, routing from point a to point B in the most efficient time and power consumption.

This will make an approach like Adversarial Discriminative Domain Adaptation is the best for solving the problem, Adversarial learning methods are a promising approach to training robust deep networks and can generate complex samples across diverse domains. They also can improve recognition despite the presence of domain shift or dataset bias.

Based on this we proposed a solution as follows:

First, we build an encoder that will take as an input the source domain representations, and its output will be the input to the policy training part of the algorithm which will perform as efficient as it does in its current situation, but a huge amount of computational time is expected.

The first step is called Pre-Training.

Then we build the second step which is Adversarial Adaptation, in this step we align the source domain with its encoder with the target domain and its not-trained encoder, and both outputs will enter a discriminator module which will try to classify which input is from the source domain and which is from the target, we train an Adversarial Loss Function to try to discriminate as best as possible and in return the encoders goal is to make the discriminator's job as hard as possible.

Encoders will do so by minimizing the distance between the source and target domains under their respective mappings which will contain basically weights of the agents between source and target domains.

The third step is Testing, in this step we use the target domain samples and target encoder, but its output will enter the policy trained from the source, by doing so we want to use the same trained policy since in step two we made the target distribution very similar to the source.

### **Выводы.**

We created 500 random topologies, each has between 10 and 100 nodes, we choose 20 topologies with the most dissimilar embeddings.

Our performance indicators are firstly the solution search time, to test this indicator we run the algorithm on one of the new created topologies and recorded the time needed to present results, and then we run the algorithm after adding our approach on the same topology and compared results.

The second performance indicator is the composition of time and power consumption for the solution on each topology.

Solving the problem of generalization of the DQN-Routing will boost the algorithm's power to solve more complicated routing problems and their many applications in the vast field of multiagent reinforcement learning.

Барбахан И. (автор)

Фильченков А.А. (научный руководитель)

## УДК 004.8

### Мультиагентное подкрепление алгоритм маршрутизации глубокого обучения DQN-Routing generalization approach

Барбахан И. (Национальный исследовательский университет ИТМО)  
Фильченков Андрей Александрович, доцент факультета информационных технологий и программирования.  
(Национальный исследовательский университет ИТМО)

Эта работа посвящена расширению возможностей обобщения алгоритма DQN-Routing, когда он применяется к новой топологии с использованием подхода, основанного на методе адаптации домена.

Введение.

DQN-Routing - это алгоритм, основанный на сетях глубокого обучения с подкреплением в сочетании с Link-государственный протокол и предварительное обучение с учителем. Алгоритм предназначен для решения задачи

маршрутизации в сети (График) и первоначально была протестирована с использованием имитационных моделей, построенных вручную

компьютерной сети и системы обработки багажа. Он превосходит современные алгоритмы маршрутизации. Однако в нынешней ситуации единственный способ применить этот алгоритм к новой топологии (сеть / график) - перезапустить весь тренировочный процесс.

Это неэффективно по времени из-за структуры алгоритма, в которой каждый узел является агентом, который

необходимо обучать с нуля, и это займет много вычислительного времени.

Поскольку DQN-Routing - новый метод, ранее не проводилось никаких работ по решению проблема обобщения, и эта работа создаст основу для улучшения общей производительности алгоритма.

Основная часть.

Предлагаемое решение представляет собой решение для адаптации домена, которое в основном представляет собой подкатегорию передачи

Учусь. Обычно адаптация предметной области используется, когда алгоритм хорошо выполняет задачу, и необходимо перенести эти знания на новую, но похожую задачу, первая задача называется «Исходный домен», а вторая задача называется «Целевой домен».

Передача знаний означает, что нет необходимости повторно обучать и повторно запускать всю процедуру для

новая задача, но использовать знания, полученные из источника, для решения задачи цели.

Для DQN-алгоритма у нас разные топологии, что означает разное расположение графа.

а затем новые агенты в новых узлах, это в основном сдвиг домена, поскольку задача та же, маршрутизация из точки А в точку Б за наиболее эффективное время и потребление энергии.

Это сделает подход, подобный Adversarial Discriminative Domain Adaptation, лучшим для решения

Проблема, состязательные методы обучения - многообещающий подход к обучению надежных глубоких сетей и может создавать сложные образцы в различных областях. Они также могут улучшить узнаваемость, несмотря на наличие сдвига домена или смещения набора данных.

Исходя из этого, мы предложили следующее решение:

Сначала мы создаем кодировщик, который будет принимать в качестве входных данных представления исходного домена, а его выходные данные.

будет входом в часть алгоритма обучения политике, который будет работать так же эффективно, как и в текущей ситуации, но ожидается огромное количество вычислительного времени.

Первый шаг называется предварительным обучением.

Затем мы создаем второй шаг - состязательную адаптацию, на этом шаге мы выравниваем исходный код. домен с его кодировщиком с целевым доменом и его не обученным кодировщиком, и оба выхода будут введите модуль дискриминатора, который попытается классифицировать, какие входные данные поступают из исходного домена и который исходит от цели, мы обучаем функцию состязательных потерь, чтобы попытаться различать как можно лучше возможно, а взамен цель кодировщиков - максимально усложнить работу дискриминатора.

Кодеры будут делать это, минимизируя расстояние между исходным и целевым доменами под их соответствующие сопоставления, которые будут содержать в основном веса агентов между источником и целью домены.

Третий шаг - это тестирование, на этом шаге мы используем образцы целевого домена и целевой кодировщик, но его вывод войдет в политику, обученную из источника, при этом мы хотим использовать те же обученные policy, поскольку на втором шаге мы сделали целевой дистрибутив очень похожим на исходный.

### **Выводы.**

Мы создали 500 случайных топологий, в каждой от 10 до 100 узлов, мы выбрали 20 топологий с самыми непохожими вложениями.

Наши показатели эффективности - это, во-первых, время поиска решения, чтобы проверить этот индикатор, мы запускаем

алгоритм на одной из вновь созданных топологий и записал время, необходимое для представления результатов, и

затем мы запускаем алгоритм после добавления нашего подхода к той же топологии и сравнения результатов.

Второй показатель эффективности - это соотношение затрат времени и энергии на решение на каждой топологии.

Решение проблемы обобщения DQN-маршрутизации повысит мощность алгоритма для решения более сложных задачи маршрутизации и их многочисленные приложения в обширной области многоагентной обучение с подкреплением.

Барбахан И. (автор)

Фильченков А.А. (научный руководитель)