# Reinforcement Learning Based Routing in Networks: Review and Classification of Approaches

A comprehensive review of the literature of RL-based protocols.

Master’s Degree in Computer Science

**Alessio Bandiera** (1234567)

SAPIENZA
UNIVERSITÀ DI ROMA

# Table of Contents

## Introduction

# Motivation

## Introduction

Modern networks have become far more complex, dynamic, and diverse than early manually configured systems. This has made *human* management insufficient.

# Motivation

## Introduction

Modern networks have become far more complex, dynamic, and diverse than early manually configured systems. This has made *human* management <u>insufficient</u>.

As a result, **Machine Learning (ML)** is increasingly used to handle tasks such as

- traffic prediction
- fault and configuration management
- congestion control

# Motivation

## Introduction

Modern networks have become far more complex, dynamic, and diverse than early manually configured systems. This has made *human* management insufficient.

As a result, **Machine Learning (ML)** is increasingly used to handle tasks such as

- traffic prediction
- fault and configuration management
- congestion control

The goal is to **automatically** learn network conditions in order to improve the user experience, while optimizing network resources.

# The routing problem
## Introduction

In networks, **routing** is the problem of sleecting paths to send packets from source(s) to destination(s), while

- meeting **Quality of Service (QoS)** requirements
- optimizing network resources

# The routing problem
## Introduction

In networks, **routing** is the problem of sleecting paths to send packets from source(s) to destination(s), while

- meeting **Quality of Service (QoS)** requirements
- optimizing network resources

However, whenever multiple metrics are required, the routing problem becomes NP-complete.

# The routing problem
## Introduction

In networks, **routing** is the problem of sleecting paths to send packets from source(s) to destination(s), while

- meeting **Quality of Service (QoS)** requirements
- optimizing network resources

However, whenever multiple metrics are required, the routing problem becomes NP-complete.

This is the reason why ML is seen as a strategy to revolutionize current **routing** techniques.

# Reinforcement Learning

## Introduction

As already discussed throughout the lectures of this course, **Reinforcement Learning (RL)** is an ML technique inspired by behavioral psychology that provides system modeling based on *agents* that interect with their *environment*

As already discussed throughout the lectures of this course, **Reinforcement Learning (RL)** is an ML technique inspired by behavioral psychology that provides system modeling based on *agents* that interect with their *environment*

For the purpose of finding better solutions for **routing** in networks, RL-based algorithms seem particularly promising because TODO

# Q-learning
## Introduction

In 1989 Watkins et al. [Wat+89] proposed the most-widely adopted flavour of Reinforcement Learning, called **Q-learning**.

Q-learning is a **model-free** approach that aims at estimating the action function $Q_{\pi^*}(s, a)$, where $\pi^*$ is the optimal policy

# Q-learning
## Introduction

In 1989 Watkins et al. [Wat+89] proposed the most-widely adopted flavour of Reinforcement Learning, called **Q-learning**.

Q-learning is a **model-free** approach that aims at estimating the action function $Q_{\pi^*}(s, a)$, where $\pi^*$ is the optimal policy

Very importantly, his approximation of the action function is *independent of the policy* followed by the agents, making Q-learning applicable in a wide variety of contexts.

# Q-learning

## Introduction

In 1989 Watkins et al. [Wat+89] proposed the most-widely adopted flavour of Reinforcement Learning, called **Q-learning**.

The action-value is updates through the following formula:

$$Q_n(s_n, a_n) = (1 - \alpha) \cdot Q_{n-1}(s_n, a_n) + \alpha \cdot \left[ R_n + \gamma \cdot \max_{a \in \mathcal{A}} Q_{n-1}(s_{n+1}, a) \right]$$

where $\alpha$ is the **learning factor**, and $\gamma$ is the **discount rate**.

The initial Q-values $Q_0(s, a)$ are assumed given.

# Q-learning

## Introduction

In 1989 Watkins et al. [Wat+89] proposed the most-widely adopted flavour of Reinforcement Learning, called **Q-learning**.

Moreover, this function can be rewritten in its more common **discrete time** $t$ form:

$$Q(s_t, a_t) = (1 - \alpha) \cdot Q(s_t, a_t) + \alpha \cdot \left[ R_{t+1} + \gamma \cdot \max_{a \in \mathcal{A}} Q(s_{t+1}, a) \right]$$

# Q-learning

## Introduction

In 1989 Watkins et al. [Wat+89] proposed the most-widely adopted flavour of Reinforcement Learning, called **Q-learning**.

Most importantly, Watkins showed that Q-laerning **converges** to the optimum action-values with probability 1, as long as all actions are repeatedly sampled in all states.

Indeed, this is the reason why Q-learning is the most popular and effective larning technique in the field of delayed reinforcement.

# Q-routing
## Introduction

In 1993 Boyan and Littman [BL93] proposed a hop-by-hop routing algorithm based on Q-learning, called **Q-routing**.

# Q-routing
## Introduction

In 1993 Boyan and Littman [BL93] proposed a hop-by-hop routing algorithm based on Q-learning, called **Q-routing**.

After their seminal work, tens of works followed the original idea of using RL to optimize routing, while also considering the evolution of commuinication networks and users requirements.

# Q-routing

## Introduction

In 1993 Boyan and Littman [BL93] proposed a hop-by-hop routing algorithm based on Q-learning, called **Q-routing**.

After their seminal work, tens of works followed the original idea of using RL to optimize routing, while also considering the evolution of commuinication networks and users requirements.

In fact, most of the existing RL-based routing protocols today are extensions of their original work.

# The work in exam

## Introduction

The paper that will be discussed today was published in 2019 by Mammeri [Mam19].

# The work in exam
## Introduction

The paper that will be discussed today was published in 2019 by Mammeri [Mam19].

Their work is a review of **60 papers**, which essentially covers the literature of RL-based routing algorithms completely.

# The work in exam
## Introduction

The paper that will be discussed today was published in 2019 by Mammeri [Mam19].

Their work is a review of **60 papers**, which essentially covers the literature of RL-based routing algorithms completely.

In particular, their work has 2 main objectives:

1. provide a **comprehensive peresentation** of the main characteristics of RL-based routing protocols

# The work in exam
## Introduction

The paper that will be discussed today was published in 2019 by Mammeri [Mam19].

Their work is a review of **60 papers**, which essentially covers the literature of RL-based routing algorithms completely.

In particular, their work has 2 main objectives:

1. provide a **comprehensive peresentation** of the main characteristics of RL-based routing protocols
2. provide **classification criteria** to enable analysis and comparison of existing protocols

# Table of contents

## Introduction

This presentation will first provide a general idea of **Q-routing**, which has been a very influential idea and most of the current literature is based on this RL-based algorithm.

# Table of contents

## Introduction

This presentation will first provide a general idea of **Q-routing**, which has been a very influential idea and most of the current literature is based on this RL-based algorithm.

Subsequently, the most important segment of this review will be presented: the **classification criteria** that the authors defined in order to categorize the literature.

This presentation will first provide a general idea of **Q-routing**, which has been a very influential idea and most of the current literature is based on this RL-based algorithm.

Subsequently, the most important segment of this review will be presented: the **classification criteria** that the authors defined in order to categorize the literature.

To their knowledge, the authors state that their work is the first in the literature that proposed classification criteria to help comparing all available RL-based routing protocols in the literature.

# Table of Contents

TODO intro slide

# Q-routing

## Q-routing

**function** Qrouting( )
    Initialize $Q_i$ matrix randomly
    **while** termination condition holds **do**
        **if** packet $P$ is ready to be sent to $d$ **then**
            Determine node $j^* \leftarrow \arg\min_{j \in \mathcal{N}(i)} Q_i(d,j)$
            Send packet to node $j^*$
            Collect estimate $\theta_{j^*}(d)$ from node $j^*$
            Update $Q_i(d,j^*) \leftarrow (1-\alpha) \cdot Q_i(d,j^*) + \alpha \cdot \left[W_i^q(P) + T_{ij^*} + \theta_{j^*}(d)\right]$
        **end if**
    **end while**
**end function**

- $i$ is the node that is currently running the algorithm
- $P$ is a packet that node $i$ needs to forward to destination $d$
- $Q_i(d,j)$ is the *delivery delay* that $i$ estimates it takes, for node $j$, to deliver the packet $P$ at destination $i$
- $\mathcal{N}(j)$ is the set of $j$'s neighbors
- $\theta_j(d)$ is $j$'s estimate for the time remaining in the trip to destination $d$ of packet $P$
- $W_i^q(P)$ is the time spent by packet $P$ in node $i$'s queue
- $T_{ij}$ is the transmission time between nodes $i$ and $j$

# Q-routing

## Q-routing

**function** Qrouting( )
Initialize $Q_i$ matrix randomly
**while** termination condition holds **do**
**if** packet $P$ is ready to be sent to $d$ **then**
Determine node $j^* \leftarrow \arg\min_{j \in \mathcal{N}(i)} Q_i(d, j)$
Send packet to node $j^*$
Collect estimate $\theta_{j^*}(d)$ from node $j^*$
Update $Q_i(d, j^*) \leftarrow (1 - \alpha) \cdot Q_i(d, j^*) + \alpha \cdot \left[W_i^q(P) + T_{ij^*} + \theta_{j^*}(d)\right]$
**end if**
**end while**
**end function**

Upon sending packet $P$ to node $j^*$, node $i$ receives back from node $j^*$ the estimate

$$\theta_{j^*}(d) = \min_{k \in \mathcal{N}(j^*)} Q_{j^*}(d, k)$$

# Q-routing

## Q-routing

**function** Qrouting( )
  Initialize $Q_i$ matrix randomly
  **while** termination condition holds **do**
    **if** packet $P$ is ready to be sent to $d$ **then**
      Determine node $j^* \leftarrow \arg\min_{j \in \mathcal{N}(i)} Q_i(d,j)$
      Send packet to node $j^*$
      Collect estimate $\theta_{j^*}(d)$ from node $j^*$
      Update $Q_i(d,j^*) \leftarrow (1-\alpha) \cdot Q_i(d,j^*) + \alpha \cdot \left[W_i^q(P) + T_{ij^*} + \theta_{j^*}(d)\right]$
    **end if**
  **end while**
**end function**

Then, node $i$ updates $Q_i(d,j^*)$ based on the *update formula* for Q-learning described earlier:

$$Q(s_t, a_t) = (1-\alpha) \cdot Q(s_t, a_t) + \alpha \cdot \left[R_{t+1} + \gamma \cdot \max_{a \in \mathcal{A}}(s_{t+1}, a)\right]$$

# Flaws of Q-learning

## Q-routing

Despite the wide adoption, Q-routing has some flaws. Some problems are direct consequences of Q-learning such as

- *slow convergence*
- *high parameter setting sensitivity*

# Flaws of Q-learning

## Q-routing

Despite the wide adoption, Q-routing has some flaws. Some problems are direct consequences of Q-learning such as

- *slow convergence*
- *high parameter setting sensitivity*

However, there are also problems arising from the algorithm itself, for instance the **Q-value freshness**: $\theta_j(d)$ is evaluated only upon packet transmission on a route, therefore if a route is not used for a long time its estimate becomes *outdated*.

TODO conclusive slide

# Table of Contents

## Classification criteria

# TODO

## Classification criteria

TODO intro slide

# Classification criteria

## Classification criteria

TODO paragraph

These criteria are divided into 3 groups:

1. **Context of use**: criteria based on the *target applications*
2. **Design characteristics**: criteria based on the *design* of the protocols
3. **Performance**: criteria based on qualitative evaluation on *overhead* and *metrics*

# Context of use

## Classification criteria: Context of use

TODO intro slide

# Network class and assumptions

## Classification criteria: Context of use

TODO

# Routing optimization context

## Classification criteria: Context of use

A *good* protocol should be able to determine and select the optimal paths to convey data from sources to destinations. This can be TODO

# Unicast or Multicast

## Classification criteria: Context of use

Categorizing between **unicast or multicast** approaches is a natural choice, given the inherent *overhead* that multicast routing protocols require.

# Unicast or Multicast

## Classification criteria: Context of use

Categorizing between **unicast or multicast** approaches is a natural choice, given the inherent *overhead* that multicast routing protocols require.

Indeed, RL should be applied in multicasting scenarios only when links are sufficiently stable and/or partial delivery is allowed, otherwise convergence may be outright *impossible*.

# QoS metrics for optimization

## Classification criteria: Context of use

The choice of the metrics is one of the most important aspects of a protocol. When multiple metrics are utilized, they are *weighted* based on the importance — which depends on the target application.

# QoS metrics for optimization

## Classification criteria: Context of use

The choice of the metrics is one of the most important aspects of a protocol. When multiple metrics are utilized, they are *weighted* based on the importance — which depends on the target application.

QoS metrics that have been addressed as objectives for RL-based routing include:

- **delivery rate**: average time to deliver a packet
- **delivery ratio**: proportion of packets successfully delivered
- **hop count**: average number of hops from source to destination
- **loss ratio**: proportion of packets not delivered

# QoS metrics for optimization

## Classification criteria: Context of use

The choice of the metrics is one of the most important aspects of a protocol. When multiple metrics are utilized, they are *weighted* based on the importance — which depends on the target application.

QoS metrics that have been addressed as objectives for RL-based routing include:

- **bandwidth**: average bandwidth provided to sources
- **throughput**: average amount of bytes delivered in the entire network per time unit
- **path stability**: it indicates how a path between source and destination changes over time
- **energy consumption**: average energy consumption of the network

# QoS metrics for optimization

## Classification criteria: Context of use

The choice of the metrics is one of the most important aspects of a protocol. When multiple metrics are utilized, they are *weighted* based on the importance — which depends on the target application.

QoS metrics that have been addressed as objectives for RL-based routing include:

- **network lifetime**: average time over which the network is still alive
- **transmission power**: power for performing a transmission
- **hit delay**: average delay to return requrested data in peer-to-peer networks
- **hit ratio**: proportion of satisfied requrests in peer-to-peer networks

# QoS metrics for optimization

## Classification criteria: Context of use

The choice of the metrics is one of the most important aspects of a protocol. When multiple metrics are utilized, they are *weighted* based on the importance — which depends on the target application.

QoS metrics that have been addressed as objectives for RL-based routing include:

- **gain**: average revenue (in $) received by the agent — in business contexts
- **overhead**: average *cost* to deliver data packets at destination — the *cost* definition depends on the application

# QoS guaranteeing

## Classification criteria: Context of use

Lastly, a few routing protocols are aimed at providing QoS guarantees, regarding delivery delay to meet some requirements of **delay-sensitive applications**.

# QoS guaranteeing

## Classification criteria: Context of use

Lastly, a few routing protocols are aimed at providing QoS guarantees, regarding delivery delay to meet some requirements of **delay-sensitive applications**.

For instance, QoS guarantees are essential in *multimedia applications*.

# Design characteristics

## Classification criteria: Design characteristics

TODO intro slide

# Learning model

## Classification criteria: Design characteristics

In RL there are two possible approaches, **model-free** and **model-based**.

# Learning model

## Classification criteria: Design characteristics

In RL there are two possible approaches, **model-free** and **model-based**.

However a few algorithms are actually **model-based**, in particular

- some of them use *offline*-collected information of the environment model
- some others calculate and improve the environment model in an *online* fashion

# Learning model

## Classification criteria: Design characteristics

In RL there are two possible approaches, **model-free** and **model-based**.

Model based approaches are known to converge quickly, and thus can offer an interesting opportunity when the **speed of convergence** is a crucial requirement.

# Agent states and Actions spaces

## Classification criteria: Design characteristics

TODO

# Solution space exploration

## Classification criteria: Design characteristics

In RL the **Exploration vs Exploitation dilemma** is a well-known problem. Indeed, the *speed of convergence* strictly depends on the approach utilized to balance between *exploring* and *exploiting* the solution space.

# Solution space exploration

## Classification criteria: Design characteristics

In RL the **Exploration vs Exploitation dilemma** is a well-known problem. Indeed, the *speed of convergence* strictly depends on the approach utilized to balance between *exploring* and *exploiting* the solution space.

The *action selection* strategies in RL-based routing include:

- **Greedy strategy**: only the highes Q-value is used for selection — this strategy may take a very long time to converge

# Solution space exploration

## Classification criteria: Design characteristics

In RL the **Exploration vs Exploitation dilemma** is a well-known problem. Indeed, the *speed of convergence* strictly depends on the approach utilized to balance between *exploring* and *exploiting* the solution space.

The *action selection* strategies in RL-based routing include:

- $\varepsilon$**-greedy strategy**: in addition to the greedy strategy, the learner uses a small amount of randomness (that depends on $\varepsilon$) to explore new solutions — the most used form of selection

# Solution space exploration

## Classification criteria: Design characteristics

In RL the **Exploration vs Exploitation dilemma** is a well-known problem. Indeed, the *speed of convergence* strictly depends on the approach utilized to balance between *exploring* and *exploiting* the solution space.

The *action selection* strategies in RL-based routing include:

- **Proabability based strategy**: similar to $\varepsilon$-greedy, but the value of $\varepsilon$ is calculated from the history of learning

# Solution space exploration

## Classification criteria: Design characteristics

In RL the **Exploration vs Exploitation dilemma** is a well-known problem. Indeed, the *speed of convergence* strictly depends on the approach utilized to balance between *exploring* and *exploiting* the solution space.

The *action selection* strategies in RL-based routing include:

- **Bayesian network decision strategy**: the action selection uses *Bayesian networks* to better explore the solution space

# Solution space exploration

## Classification criteria: Design characteristics

In RL the **Exploration vs Exploitation dilemma** is a well-known problem. Indeed, the *speed of convergence* strictly depends on the approach utilized to balance between *exploring* and *exploiting* the solution space.

The *action selection* strategies in RL-based routing include:

- **Devaluation of solutions based strategy**: the Q-values are periodically decayed in order to enforce exploration of the solution space

# Solution space exploration

## Classification criteria: Design characteristics

In RL the **Exploration vs Exploitation dilemma** is a well-known problem. Indeed, the *speed of convergence* strictly depends on the approach utilized to balance between *exploring* and *exploiting* the solution space.

The *action selection* strategies in RL-based routing include:

- **New neighbors first strategy**: newly discovered nodes are favored in next hop selection — this approach is particularly useful in *mobile networks*

# Agents collaboration
## Classification criteria: Design characteristics

The original version of RL defines each agent as *independent*, and only able to interact with the environment.

# Agents collaboration

## Classification criteria: Design characteristics

The original version of RL defines each agent as *independent*, and only able to interact with the environment.

Indeed, collaboration is so prevalent among the protocols in the literature that it is possible to categorize them w.r.t. how the nodes cooperate:

- **Reactive collaboration**: nodes only provide feedback upon reception of packet
- **Proactive collaboration**: similar to the *reactive* approach, but nodes additionally broadcast their link-state information through *Hello packets* to their neighbors

# Hybridization with other optimization techniques

## Classification criteria: Design characteristics

Most of RL-based routing algorithms involve *pure* RL approaches, however some algorithms combine RL with other **optimization techniques** to speed up convergence.

# Hybridization with other optimization techniques

## Classification criteria: Design characteristics

Most of RL-based routing algorithms involve *pure* RL approaches, however some algorithms combine RL with other **optimization techniques** to speed up convergence.

Approaches include:

- Gradient methods
- Game Theory approaches
- *Bayesian network* methods
- Least square policy iteration

# Hybridization with other optimization techniques

## Classification criteria: Design characteristics

Most of RL-based routing algorithms involve *pure* RL approaches, however some algorithms combine RL with other **optimization techniques** to speed up convergence.

Approaches include:

- Neural Networks
- Genetic algorithms
- Ants optimization

# Numbers of parameters to tune

## Classification criteria: Design characteristics

A well-designed protocol should be **easily tunable**.

# Numbers of parameters to tune

Classification criteria: Design characteristics

A well-designed protocol should be **easily tunable**.

However, in addition to $\alpha$ and $\gamma$ a multitude of protocols utilize many more tunable parameters in their algorithms.

# Numbers of parameters to tune

Classification criteria: Design characteristics

A well-designed protocol should be **easily tunable**.

However, in addition to $\alpha$ and $\gamma$ a multitude of protocols utilize many more tunable parameters in their algorithms.

Additionally, weights must be assigned whenever there are **multiple metrics** to consider.

# Numbers of parameters to tune

## Classification criteria: Design characteristics

A well-designed protocol should be **easily tunable**.

However, in addition to $\alpha$ and $\gamma$ a multitude of protocols utilize many more tunable parameters in their algorithms.

Additionally, weights must be assigned whenever there are **multiple metrics** to consider.

Therefore the authors categorized the routing protocols also based on the number of tunable QoS metrics and parameter each paper offers.

## Reward functions

### Classification criteria: Design characteristics

The authors outline that the **reward function** is the most distinctive feature of existing RL-based routing protocols.

# Reward functions

## Classification criteria: Design characteristics

The authors outline that the **reward function** is the most distinctive feature of existing RL-based routing protocols.

Reward functions may be categorized into 3 classes:

- **Test-based reward functions**: the reward is assigned a constant value, depending the outcome of some *test*

  The most common test is checking if the packet was actually delivered to destination, which yields a *binary outcome* for the reward

# Reward functions

## Classification criteria: Design characteristics

The authors outline that the **reward function** is the most distinctive feature of existing RL-based routing protocols.

Reward functions may be categorized into 3 classes:

- **Linear reward functions**: they have the following general form

$$R = C + \sum_{k=1}^{H} \omega_k \cdot M_k$$

  - $C$ is a constant factor that depends on the test chosen by the protocols
  - $H$ is the number of metrics of the protocol
  - $\omega_k$ is the weight of the $k$-th metric
  - $M_k$ is the value of the $k$-th metric

# Reward functions

## Classification criteria: Design characteristics

The authors outline that the **reward function** is the most distinctive feature of existing RL-based routing protocols.

Reward functions may be categorized into 3 classes:

- **Nonlinear reward functions**: this type is less common among RL-protocols, and they are designed with different forms of combinations of metrics depending on the specific application

# Q-value updating rule forms

## Classification criteria: Design characteristics

Over half of proposed RL-based routing algorithms are direct applications of Q-learning as originally proposed by Watkins.

# Q-value updating rule forms

## Classification criteria: Design characteristics

Over half of proposed RL-based routing algorithms are direct applications of Q-learning as originally proposed by Watkins.

However the remaining half of the protocols use procedures that either sued a modified Q-value updating rule, or do not rely on Q-learning at all.

# Performance aspects

## Classification criteria

TODO intro slide

# Communication overhead

## Classification criteria: Performance aspects

**Communication overhead** is a crucial part of the design of a routing protocol, which depends on how the protocols defines the exchange of relvant information between nodes of the network.

# Communication overhead

## Classification criteria: Performance aspects

**Communication overhead** is a crucial part of the design of a routing protocol, which depends on how the protocols defines the exchange of relvant information between nodes of the network.

Therefore, the overhead of the reviewed protocols have been categorized from a *qualitative* point of view into:

- **null overhead**: there is no exchange of information between agents

# Communication overhead

## Classification criteria: Performance aspects

**Communication overhead** is a crucial part of the design of a routing protocol, which depends on how the protocols defines the exchange of relvant information between nodes of the network.

Therefore, the overhead of the reviewed protocols have been categorized from a *qualitative* point of view into:

- **low overhead**: the chosen next hop returns a feedback in an explicit ACK packet, or it includes its feedback when, in turn, it (re)forwards the packet

  Half of the reviewed ptorocols fall under this category

# Communication overhead

## Classification criteria: Performance aspects

**Communication overhead** is a crucial part of the design of a routing protocol, which depends on how the protocols defines the exchange of relvant information between nodes of the network.

Therefore, the overhead of the reviewed protocols have been categorized from a *qualitative* point of view into:

- **medium overhead**: this is the case of protocols in which the feedback from the destination is propagated to all hops through an explicits ACK packet

# Communication overhead

## Classification criteria: Performance aspects

**Communication overhead** is a crucial part of the design of a routing protocol, which depends on how the protocols defines the exchange of relvant information between nodes of the network.

Therefore, the overhead of the reviewed protocols have been categorized from a *qualitative* point of view into:

- **medium overhead**: this is the case of protocols in which the feedback from the destination is propagated to all hops through an explicits ACK packet

# Communication overhead

## Classification criteria: Performance aspects

**Communication overhead** is a crucial part of the design of a routing protocol, which depends on how the protocols defines the exchange of relvant information between nodes of the network.

Therefore, the overhead of the reviewed protocols have been categorized from a *qualitative* point of view into:

- **high overhead**: these protocols require that nodes periodically exchange link-state information

# State space overhead

## Classification criteria: Performance aspects

RL-based algorithms requires *memory* to store the **states of the agents**, and the number of states may be very high.

# State space overhead

## Classification criteria: Performance aspects

RL-based algorithms requires *memory* to store the **states of the agents**, and the number of states may be very high.

Hence, the protocols can be *qualitatively* grouped based on the **state space overhead**:

- **very low overhead**: then when state space is states of a packet TODO WHAT

# State space overhead

## Classification criteria: Performance aspects

RL-based algorithms requires *memory* to store the **states of the agents**, and the number of states may be very high.

Hence, the protocols can be *qualitatively* grouped based on the **state space overhead**:

- **low overhead**: when the state space is the node IDs — most of the reviewed papers fall under this category

# State space overhead

## Classification criteria: Performance aspects

RL-based algorithms requires *memory* to store the **states of the agents**, and the number of states may be very high.

Hence, the protocols can be *qualitatively* grouped based on the **state space overhead**:

- **limited overhead**: when the state space depends on external factors — e.g. the number of transmission power leves, the maximum number of available channels, etc.

# State space overhead

## Classification criteria: Performance aspects

RL-based algorithms requires *memory* to store the **states of the agents**, and the number of states may be very high.

Hence, the protocols can be *qualitatively* grouped based on the **state space overhead**:

- **high overhead**: when the state space is a list of whole paths withthier current characteristics

# Action space overhead

## Classification criteria: Performance aspects

Additionally, RL-based algorithm also require *memory* store all the **possible actions** that agents can perform.

# Action space overhead

## Classification criteria: Performance aspects

Additionally, RL-based algorithm also require *memory* store all the **possible actions** that agents can perform.

Therefore again, the protocols can be *qualitatively* grouped based on the **action space**:

- **low overhead**: when the action space depends on external factors TODO WHAT WHY

# Action space overhead

## Classification criteria: Performance aspects

Additionally, RL-based algorithm also require *memory* store all the **possible actions** that agents can perform.

Therefore again, the protocols can be *qualitatively* grouped based on the **action space**:

- **medium overhead**: when the action space depends on the number of nodes in the neighborhood

# Action space overhead

## Classification criteria: Performance aspects

Additionally, RL-based algorithm also require *memory* store all the **possible actions** that agents can perform.

Therefore again, the protocols can be *qualitatively* grouped based on the **action space**:

- **high overhead**: when the action space depends on either
  - the number of *dynamic paths*
  - the number of or *predefined paths*
  - the number of *grids* in the network — WHAT ARE THESE

# Action space overhead

## Classification criteria: Performance aspects

Additionally, RL-based algorithm also require *memory* store all the **possible actions** that agents can perform.

Therefore again, the protocols can be *qualitatively* grouped based on the **action space**:

- **very high overhead**: when the state space depends on combinations of channels subsets or paths

# Proof of convergence

## Classification criteria: Performance aspects

In the optimization field, the **convergence** to optimal solutions is an *expected* property.

# Proof of convergence

## Classification criteria: Performance aspects

In the optimization field, the **convergence** to optimal solutions is an *expected* property.

Nevertheless, many existing techniques to solve multicriteria optimization problems are not guarantted to reach the optimal solution.

# Proof of convergence

## Classification criteria: Performance aspects

In the optimization field, the **convergence** to optimal solutions is an *expected* property.

Nevertheless, many existing techniques to solve multicriteria optimization problems are not guarantted to reach the optimal solution.

Regarding RL-based algorithms, from the original work of Watkins et al. [Wat+89] it is possible to derive proofs of convergence of RL algorithms, however:

- not all RL-based approaches are based on the standard implementation of Q-learning
- many papers rely on **additional assumptions** that guarantee convergence

# Proof of convergence

## Classification criteria: Performance aspects

In the optimization field, the **convergence** to optimal solutions is an *expected* property.

Nevertheless, many existing techniques to solve multicriteria optimization problems are not guarantted to reach the optimal solution.

Regarding RL-based algorithms, from the original work of Watkins et al. [Wat+89] it is possible to derive proofs of convergence of RL algorithms, however:

- not all RL-based approaches are based on the standard implementation of Q-learning
- many papers rely on **additional assumptions** that guarantee convergence

## Protocol performance (in simulations)

Classification criteria: Performance aspects

It would be clearly important to categorize the protocols based on the **performance** based on the simulations.

# Protocol performance (in simulations)

## Classification criteria: Performance aspects

It would be clearly important to categorize the protocols based on the **performance** based on the simulations.

However, given the

- large number of reviewed protocols
- wide variety of reward functions
- wide range of metric weights

the authors found impratical to even provide *qualitative* evaluation of the performance of each single protocol.

# Protocol performance (in simulations)

## Classification criteria: Performance aspects

It would be clearly important to categorize the protocols based on the **performance** based on the simulations.

However, given the

- large number of reviewed protocols
- wide variety of reward functions
- wide range of metric weights

the authors found impratical to even provide *qualitative* evaluation of the performance of each single protocol.

Therefore, they limited their effor to present the data reported by the origianl authors *as-is*.

# Table of Contents

## Conclusion and challenges

# TODO

## Conclusion and challenges

TODO