Telecommunication Systems
https://doi.org/10.1007/s11235-019-00600-9

Efficient admission control and resource allocation mechanisms for public safety communications over 5G network slice

Anuar Othman[1,2] · **Nazrul Anuar Nayan**[1]

Abstract
The deployment of a broadband public safety (PS) mobile network can be undertaken in different ways. One method involves combining commercial networks with private ones to reduce their deployment cost and time to market. In shared commercial networks, priorities should be defined to differentiate traffic not only between consumers and PS users but also between different PS organizations and type of services. Prioritization must also ensure that emergency calls are always served under normal conditions and during disasters. The recent advent of the fifth generation (5G) wireless standard introduces new technologies, such as network slicing (NS), which allows the provision of logical PS networks in a shared 5G system wherein each slice can be dedicated to an organization or to a type of service. However, 5G management and orchestration become a challenging task with NS, e.g., in handling resource allocation between slices with diverse requirements. Therefore, efficient solutions for slice resource allocation are required to facilitate this task. In this paper, we present a review of adaptive and dynamic resource allocation leveraging on heuristic and reinforcement learning-based algorithms that have been proposed in the recent literatures. The challenge in implementing these algorithms is to find the most suitable one for our problem, i.e., an algorithm that is highly scalable, able to solve problems immediately, and exhibits the best convergence properties in terms of speed and ability to find the global optimum.

Keywords Public safety · 5G network slicing · Resource allocation · Reinforcement learning

1 Introduction

1.1 Broadband public safety (PS) network

Until recently, Public Safety (PS) organizations have been relying on dedicated networks and specialized technologies, such as Project 25 (P25) and Terrestrial Trunked Radio (TETRA), for their mobile communications. However, all these technologies are narrowband; hence, their capabilities for broadband applications are limited [1]. Mobile broadband, such as Long-Term Evolution (LTE) and 5G, will enable the mobile use of applications and allow smart devices, including tablets, smartphones, and laptops, to be used by PS users. In [2], the authors present a comparative survey on these technologies, discuss their convergence, and highlight the benefits of broadband services for mission-critical data communications.

A broadband PS network can be delivered in various ways. One method involves combining commercial networks with private ones to reduce their deployment cost and time to market [3]. A commercial network, operated by a mobile network operator (MNO) and shared between consumers and critical users, should be able to guarantee PS requirements, e.g., high quality of service (QoS) [1, 3]. Many PS organizations worldwide, such as those in Belgium, Finland, and the USA, are already using this approach instead of purely private LTE networks for their broadband services while 5G test network activities are ongoing globally.

Furthermore, a broadband PS network must perform efficiently in different scenarios, including during disasters when network infrastructure is frequently degraded or destroyed. These issues are addressed in [4], in which the authors extensively review device-to-device and dynamic wireless networks as complementary techniques to support

✉ Nazrul Anuar Nayan
nazrul@ukm.edu.my

[1] Centre for Integrated Systems Engineering and Advanced Technology, Faculty of Engineering and Built Environment, Universiti Kebangsaan Malaysia, UKM, 43600 Bangi Selangor, Malaysia

[2] Arubaito Secure Networks, Kuala Lumpur, Malaysia

Published online: 27 July 2019

the aforementioned requirements. In [5], the authors discuss open research issues in the next-generation broadband PS network.

1.2 Quality of service (QoS) control in LTE and 5G

In a shared network, QoS control plays a key role in prioritizing PS traffic during normal operations and when the network is congested. Priorities should be defined to differentiate traffic not only between users (e.g., PS and consumers) but also between PS organizations (e.g., police and fire departments) and service types (e.g., voice and data). Prioritization must also ensure that emergency calls are always served under normal conditions and during disasters.

The concept of QoS in LTE is based on a bearer, i.e., an information transmission path of defined capacity, latency, and reliability. Therefore, QoS control in LTE is responsible for the authorization and enforcement of maximum QoS that is authorized for a service data flow or an Internet Protocol (IP) connectivity access network bearer [6]. Allocation and retention priority (ARP) and QoS class identifier (QCI) are among the QoS attributes that are associated with an LTE bearer [7]. The former defines the admission and preemption characteristics of a bearer, whereas the latter is used to prioritize bearer packets in the queuing and scheduling process. Standardized characteristics associated with QCI values are resource type, priority, packet delay budget, and packet error loss rate [6]. Consequently, the QoS of an LTE service is achieved by allocating a dedicated bearer with specific ARP definitions, QCI values, maximum bit rate (MBR), minimum reserved traffic rate, and aggregated MBR for a group of bearers of a single user.

Furthermore, access class barring can be activated during congestion in a specific area or base station to enable a prioritized user to gain access to the network by barring or holding low-priority users. By contrast, QoS control for 5G is based on flows and uses several tools, including ARP and QoS flow identifier (QFI), to prioritize traffic. The concept is still being refined by the 3rd Generation Partnership Project (3GPP) and is expected to be continuously enhanced in the next few years. Although LTE and 5G QoS control can prioritize PS traffic in a shared network, they pose a risk of slowing down the overall call control setup because QCIs and QFIs are requested on a per-session basis [8].

1.3 Network slicing (NS) in 5G

The advent of the 5G standard introduces new technologies, such as software-defined networking (SDN), network function virtualization (NFV), and multi-access edge computing, which benefit broadband PS networks by reducing their deployment cost and time to market. Both SDN and NFV enable NS, which is a network concept that allows an MNO to create logical networks or slices over a shared infrastructure and radio spectrum. In particular, 3GPP defines a slice as "a logical network that provides specific network capabilities and network characteristics" [9]. Slices are customized to provide optimized solutions for different use cases with specific requirements in terms of functionality, performance, and isolation [10]. As illustrated in Fig. 1, slice or logical network 1 is deployed to deliver enhanced mobile broadband (eMBB) service while logical network 2 is delivering smart factory service. For the case of broadband PS network use, NS allows the provision of logical PS networks in a shared 5G system where each slice can be dedicated to an organization or type of service.

Recently, NS has been eliciting increasing interest from academia and industries [12–15]. In [12], the authors introduce the concept of NS as a service (NSaaS) and describe its service orchestration and service level agreement (SLA) mappings to assist MNOs in providing NSaaS to tenants. In [13], the authors provide a summary of preliminary research effort on NS and discuss cases in different categories, namely, eMBB, massive machine type communications, and ultra-reliable and low-latency communications. In [14, 15], the authors focus on the architectural aspects of 5G networks, which include radio access network (RAN) and core network (CN), to enable the efficient implementation of NS and provide realization options and deployment examples.

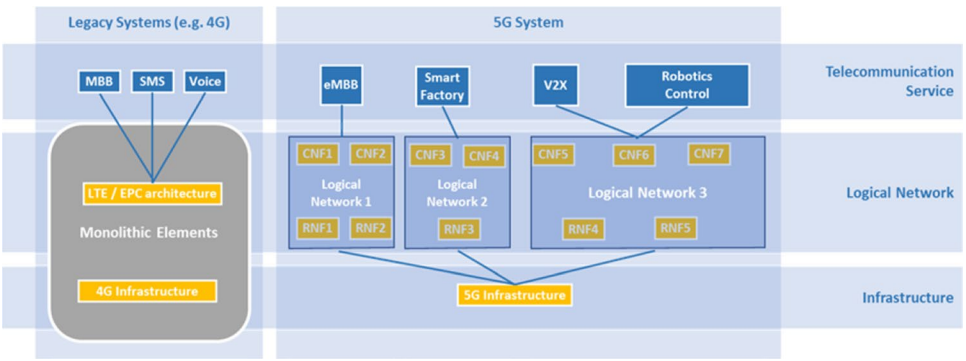


Fig. 1 Multi-tenancy in legacy LTE and slicing-enabled 5G networks [11]

In [16], the authors propose synchronous slice admission control to efficiently manage inter-slice resource allocation.

The European Union-funded 5GPPP projects provide an overall architecture vision of 5G in [11]. As illustrated in Fig. 2, the architecture is composed of several layers, namely, the service, management and orchestration (MANO), control, and data layers. The service layer contains end-to-end orchestration, business support systems, and all existing applications and services, such as voice and data communications. The MANO layer extends the MANO functions of NFV technology described in [17] with an inter-slice broker (ISB) that is responsible for managing cross-slice resource allocation and interacting with the service layer through the service management function. The control layer contains common and dedicated control functions that interact with the corresponding common and dedicated data functions of the data layer. The separation of the control and data (user) planes of virtual network functions and physical network functions are enabled via the SDN technology described in [18]. In normal operations, an ISB performs slice admission control based on predefined policies upon receiving a request for a new slice from a tenant. The admitted slice is then provisioned with the required resources, which are continuously monitored during the slice operational period based on their SLAs.

The aforementioned layered architecture improves 5G flexibility and scalability in supporting different use cases. However, 5G management and orchestration becomes a challenging task with NS, such as when handling resource allocation between slices with diverse requirements. As identified in [14], one of the challenges is to define an optimization policy that enables automated resource allocation to different slices in a multi-tenant 5G network where resource demands vary considerably in relatively short timescales. In [15], the authors highlight the challenge in defining an optimal slice admission policy to prevent the overprovisioning of scarce resources (i.e., radio spectrum). In [8], the authors indicate the need for dynamic NS to address challenging network conditions and to provide a better alternative to the resource-consuming static NS approach.

Therefore, efficient solutions for slice resource allocation are required to facilitate 5G management and orchestration tasks. In [14], the authors emphasize the need for automated policy validation and computationally efficient algorithms to perform the corresponding resource orchestration actions in a timely manner. In [15], the authors call for novel algorithms and solutions to allocate network resources among different tenants, thereby allowing an MNO to achieve specific objectives, such as maximizing its overall network utility. In [8], the authors identify the need for machine learning-based control strategies to enable dynamic NS as one of the open research challenges.

In this paper, we present a review of dynamic resource allocation solutions that leverage on heuristic and machine learning-based algorithms. To the best of our knowledge, this study is the first to emphasize on the public safety use case in 5G NS resource allocation. The remainder of this paper is structured as follows. Section 2 provides an overview of reinforcement learning (RL), which is a machine learning paradigm that automates decision-making tasks via a goal-directed learning agent. Section 3 presents a comparative analysis of various studies in terms of mechanism, objective, and approach. Section 4 discusses the algorithm, strengths, and weaknesses of each proposed solution.

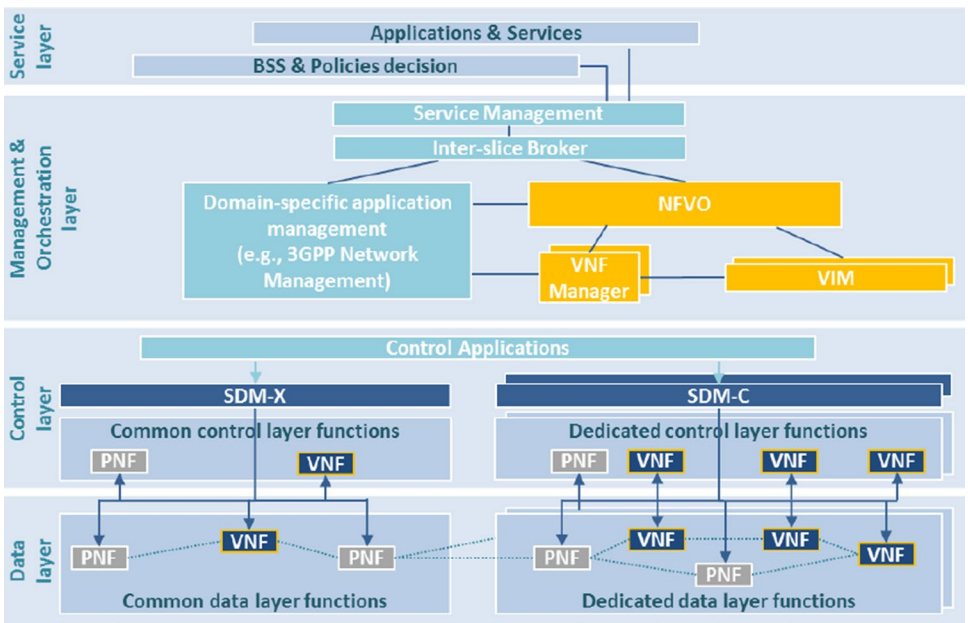


Fig. 2 5G architecture functional layers [11]

The research challenges in automated resource slicing are described in Sect. 5. Finally, concluding remarks are provided in Sect. 6.

2 Overview of reinforcement learning

Reinforcement Learning (RL) is a machine learning paradigm that automates the decision-making tasks of an agent directly from experience gained through interaction with everything outside it, i.e., its environment [19]. A reward is a feedback signal from the environment that indicates an agent's performance, and a state provides information about the environment that helps an agent determine its action. The explicit goal of an agent is to select actions that can maximize the total reward that it will receive over the long run without having complete visibility of its environment. Therefore, one of the challenges of an agent is to achieve trade-off between exploration and exploitation. The former involves selecting known actions that are proven to yield good rewards, whereas the latter involves trying new actions to discover good ones that may yield better rewards.

The major components of an RL agent include policy and value function. The former describes an agent's behavior in selecting actions, whereas the latter refers to a prediction of the total received reward. The value function indicates the quality of an action in the long run, as opposed to a reward, which is an immediate quality indicator. A third optional component is an environment model that can predict the next reward and next state resulting from a given action and state. Nearly all RL problems can be formalized as a Markov decision process (MDP). An MDP is a sequence of random states, actions, and rewards wherein all the states have a Markov property; that is, each state captures all relevant information from its history. A finite MDP is a special case in which the numbers of states, actions, and rewards are all finite. As shown in Fig. 3, an RL agent learns over a sequence of discrete time steps by selecting an action At based on its current state S_t at time step t and then receiving in return reward $Rt+1$ and state $St+1$ at time step $t+1$.

Methods for solving RL problems can be categorized as value-based, policy-based, or evolutionary. In a value-based method, such as Monte Carlo and Q-learning, an agent selects actions based on value functions that it learns in accordance with a deterministic (i.e., greedy) or near-deterministic (i.e., ϵ-greedy) policy. By contrast, in a policy-based method, value functions are not used for action selection but only the for parameter learning of a stochastic policy, which is defined as a probability distribution over actions. Then, actions are selected by the parameterized policy, which can be stochastic. Evolutionary methods, such as genetic algorithms, evaluate the behavior of many non-learning agents for every complete sequence of states or generations. Each agent uses a different policy for interacting with its environment, and the one with the most reward is selected as the optimal policy for each generation. In [19], stand-alone evolutionary methods are not considered well-suited for RL problems.

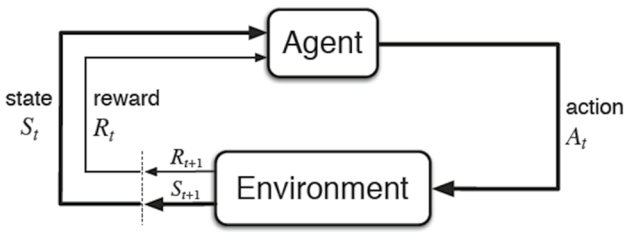


Fig. 3 Agent–environment interaction in an MDP [19]

3 Efficient resource allocation solutions

Table 1 presents a comparative analysis of the various literatures in terms of objective, mechanism, and method. The latter is classified into heuristic-based, value-based, policy-based, or evolutionary RL.

3.1 Heuristic-based

A heuristic-based slice admission control with a two-tier priority level and dynamic resource allocation based on traffic load is proposed in [20]. Its objective is to maximize user quality of experience (QoE) while ensuring that the requirements of all the slices are met. As shown in Fig. 4, when a new user arrives, the algorithm first considers the constraints in terms of the intra-slice priority and QoE of the currently provisioned users in the slice. Then, user admission is decided based on the availability of sufficient resources to meet at least its minimum data rate requirement. The next step consists of allocating resources to the accepted user to maximize its QoE by considering inter- and intra-slice priority, current traffic load, and constraints in terms of physical resource availability and user channel conditions.

The same algorithm can be used for slice admission control with appropriate parameter adaptation. In such case, a tenant submits a request to the control entity (e.g., MANO) by specifying its QoS requirements (e.g., minimum and maximum data rates) and number of users to be served. The performance of the proposed algorithm is evaluated and compared with the LTE single-tier algorithm that considers only priorities between users (intra-slice). The simulation results show that the algorithm increases user QoE while providing better fairness among different slices and improving the overall utilization of network resources.

Table 1 Comparative analysis

No.	Literature	Objective	Mechanism	Method
1	NS management and prioritization in 5G mobile systems [20]	Maximize user data rate Guarantee slice throughput requirements	User and slice admission control User and slice resource allocation (RAN)	Heuristic-based RL
2	Optimizing 5G infrastructure markets: Business of NS [21]	Maximize MNO revenue Guarantee slice throughput requirements	Slice admission control Slice resource allocation (RAN)	Value-based RL
3	Machine learning-aided orchestration in multi-tenant networks [22]	Maximize efficiency of resource orchestration	Service admission control Service network resource allocation (CN, transmission)	Policy-based RL
4	Slice as an evolutionary service genetic optimization for inter-slice resource management in 5G networks [23]	Maximize long-term network utility (network throughput, latency, or reliability)	Slice admission control Slice resource allocation (abstract)	Evolutionary RL
5	Deep reinforcement learning for resource management in NS [24]	Optimize spectrum efficiency Optimize slice QoE	Slice resource allocation (RAN, CN)	Value-based RL
6	Adaptive virtual resource allocation in 5G NS using constrained MDP [25]	Guarantee slice QoS requirements Jointly optimize power allocation and channel assignment	Slice resource allocation (RAN)	ADP (value-based RL)

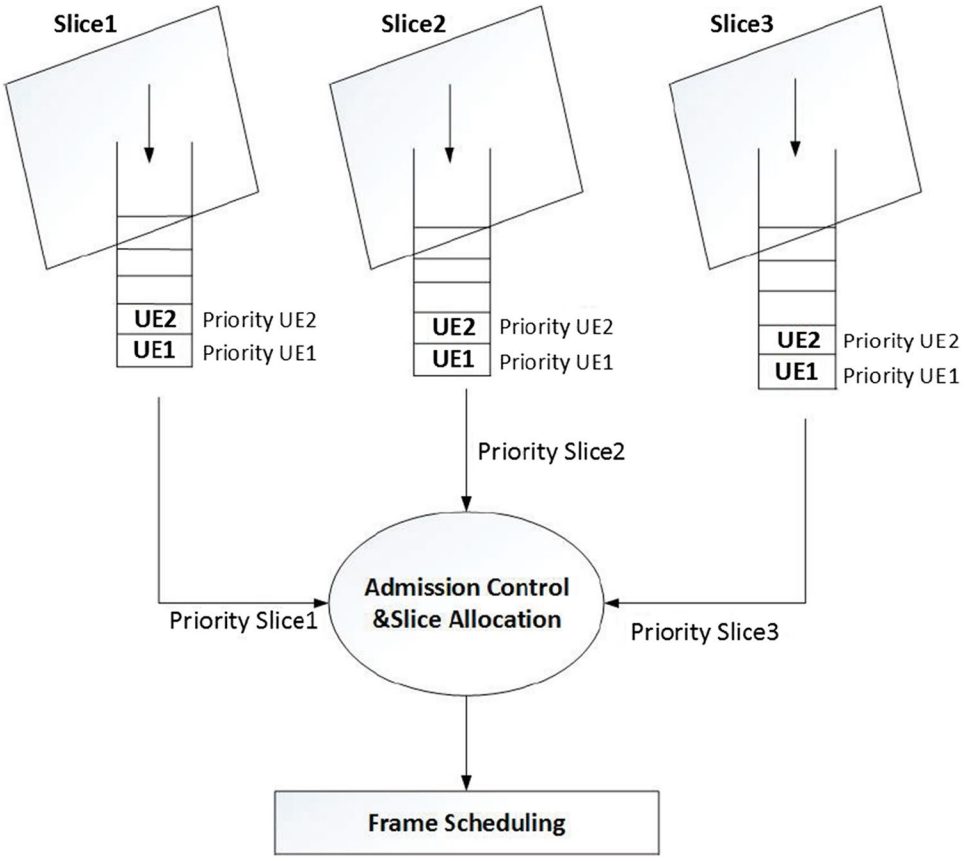


Fig. 4 Reference scenario with inter- and intra-slice priorities [20]

3.2 RL-based

In [21], the authors address the issue of efficient resource allocation by proposing a slice admission and allocation algorithm that can maximize an MNO's revenue while guaranteeing its tenants' data throughput requirements. The work considers a bidding system that receives slice requests submitted by tenants and decides on their admission based

on current load and the slice's duration, size, traffic elasticity type, and price. The system is modeled as semi-MDP, and the proposed algorithm is based on the Q-learning framework [26]. The system works by successively updating its estimation of expected reward over the long run (i.e., Q-value) after taking an action at each time step. The Q-values for all possible states and actions are stored in a lookup table, and the optimal policy at each time step is the action that maximizes the expected reward.

To ensure convergence to the global optimum, Q-learning depends on a parameter known as learning rate, which emphasizes recent estimations (i.e., more accurate estimations). Q-learning also depends on another parameter to balance between its exploitation step, which leads to maximum overall rewards, and exploration step, which may cause suboptimal results. Therefore, Q-learning learns by constantly exploring unchartered states and exploiting known policies for already visited states. The proposed algorithm's performance is evaluated and compared with random policies that randomly reject slices and naïve policies that either accept or reject all slices. The simulation results show that revenue improvement is up to 100% over naïve policies and approximately 20% over random policies.

In [22], the authors leverage policy-based RL to enable efficient resource orchestration in a multi-tenant 5G network. The proposed solution can maximize the efficiency of resource orchestration to meet the dynamic requirements of tenants while increasing the revenue of the MNO. All type of resources including RAN, CN, and transmission resources are managed by an orchestrator (i.e., MANO). All service requests from tenants are placed in the MANO's buffer prior to being processed by an RL agent. As illustrated in Fig. 5, the neural network (NN), which is a programming paradigm that generalizes nearly similar states, is used as a function approximator to learn policy parameters. Therefore, NN takes holding time, amount of required resources for each service in the buffer, and current system state as inputs, and provides an indicator on whether to accept or reject the service as output.

The reward function is proportional to the sum of penalties associated with waiting and provisioned services. At each time step, the agent uses an unbiased Monte Carlo method to estimate the discounted reward. To mitigate the high variance issue, a baseline is subtracted from the estimated value. An optimal policy is then learned by conducting the gradient descent method on policy parameters. The key concept in this optimization method is to estimate the gradient by observing the trajectories of executions that are obtained by following the policy [27]. The proposed algorithm is evaluated against simple deterministic heuristic approaches, and the simulation results show that it provides better resource orchestration and subsequently increases MNO's revenue by up to 17%.

In [23], the authors present an efficient inter-slice resource management strategy based on evolutionary RL by focusing on optimizing the slice admission policy using a genetic algorithm (GA) [28]. The objective is to maximize long-term network utility that can be flexibly defined as revenue, network throughput, delay, or reliability. As illustrated in Fig. 6, a strategy (i.e., the slice admission policy) for a sequence of arriving slice requests during an operation period (i.e., generation) is mapped onto an individual binary sequence by an encoder. A set of candidate strategies for a generation is known as a population, and all feasible strategies are represented by a codebook. Fitness is the value of the objective function to be optimized (e.g., network utility). A new population is subsequently produced for the next generation based on these values.

To approach an optimal policy, a GA performs the following three processes to its population at every generation. First, a new population is reproduced by copying and shuffling old strategies based on their fitness, thereby allowing better strategies to proliferate and worst ones to be eliminated. Then, population strategies are randomly paired in a crossover step where each pair has a chance to randomly swap a portion of their codes, which allows advanced "genes" to be passed on to the next generation of population. The final step, mutation, allows codebook exploration

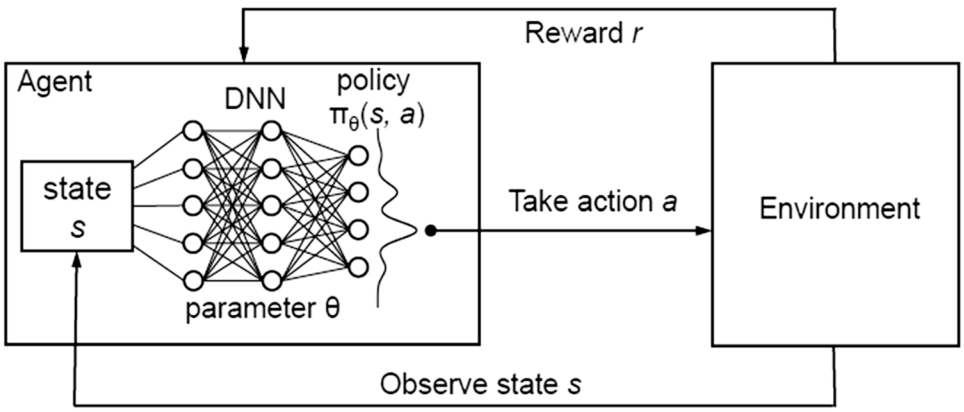


Fig. 5 RL with policy represented via NN [27]

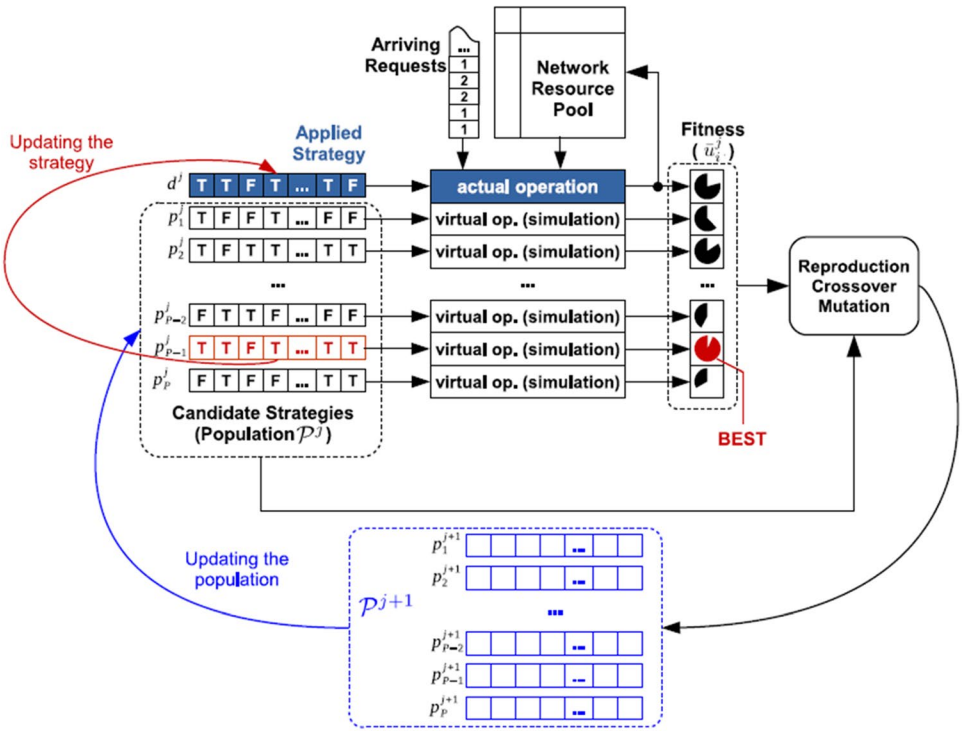


Fig. 6 Diagram of the proposed genetic slicing strategy optimizer [23]

through the random inversion of a strategy's codes. Therefore, the optimizer begins with an initial population and strategy, and both are selected randomly.

During each generation, slice requests are handled on the basis of the currently selected strategy while the population is evaluated in the background. Then, the best candidate strategy with the highest fitness is selected, and a new population is generated through reproduction, crossover, and mutation for the next generation. The proposed algorithm is evaluated on a system with two slice types with different priorities and three naïve strategies. The first strategy accepts all the requests (greedy), the second one accepts only low-priority slices (conservative), and the third one accepts only high-priority slices (opportunistic). The simulation results show that the proposed algorithm outperforms naïve strategies with long-term network utility by over 90% with respect to the global optimum. Its convergence improves as population size increases.

In [24], the authors investigate the application of deep Q-learning (DQL) [29] to providing a dynamic allocation of RAN and CN resources for admitted and provisioned slices with volatile demand variations. For RAN, the objective is to maintain acceptable spectral efficiency (SE) while providing an appropriate data rate and delay for all slices. Therefore, the goal of a DQL agent is to maximize the weighted summation of these two values by dynamically allocating a bandwidth to each slice based on the current state, i.e., the number of arrived packets in each slice within a specific time window. By contrast, the goal for CN is to minimize packet scheduling delay by balancing the resource utilization (RU) and waiting time of a service flow, with each service belonging to a slice. In such case, the goal of a DQL agent is to minimize the weighted sum of average time in all services by dynamically allocating a chain of connected network services, i.e., a service function chain (SFC) for the flow at the current time stamp based on the priority and time stamp of last-arriving flows in each SFC.

Similar to [21], the DQL agent selects an action at a time step using a near greedy policy on the current estimated Q-value, receives a reward, and updates the Q-value for the next time step. In addition, the agent stores that sequence of action, reward, and state as an experience in a memory dataset. Additionally, to cater to a large system space, DQL uses a function approximator, e.g., an NN, to approximate the Q-values to as close as possible to the target Q-values values instead of storing them in a table similar to that in [26]. The target Q-values are approximated from an episode of random samples (e.g., mini batch) of past experience from the memory dataset using a copy of the evaluation NN. This process is known as experience replay, and the copied NN is called the target NN. Then, the evaluation NN parameters are adjusted by optimizing the mean square error between the target and the approximated Q-values using stochastic gradient descent (Fig. 7). To enhance learning stability, the target NN is copied from an older version of the evaluation NN.

The proposed DQL is evaluated on hard slicing, in which each service has the same amount of allocated resources,

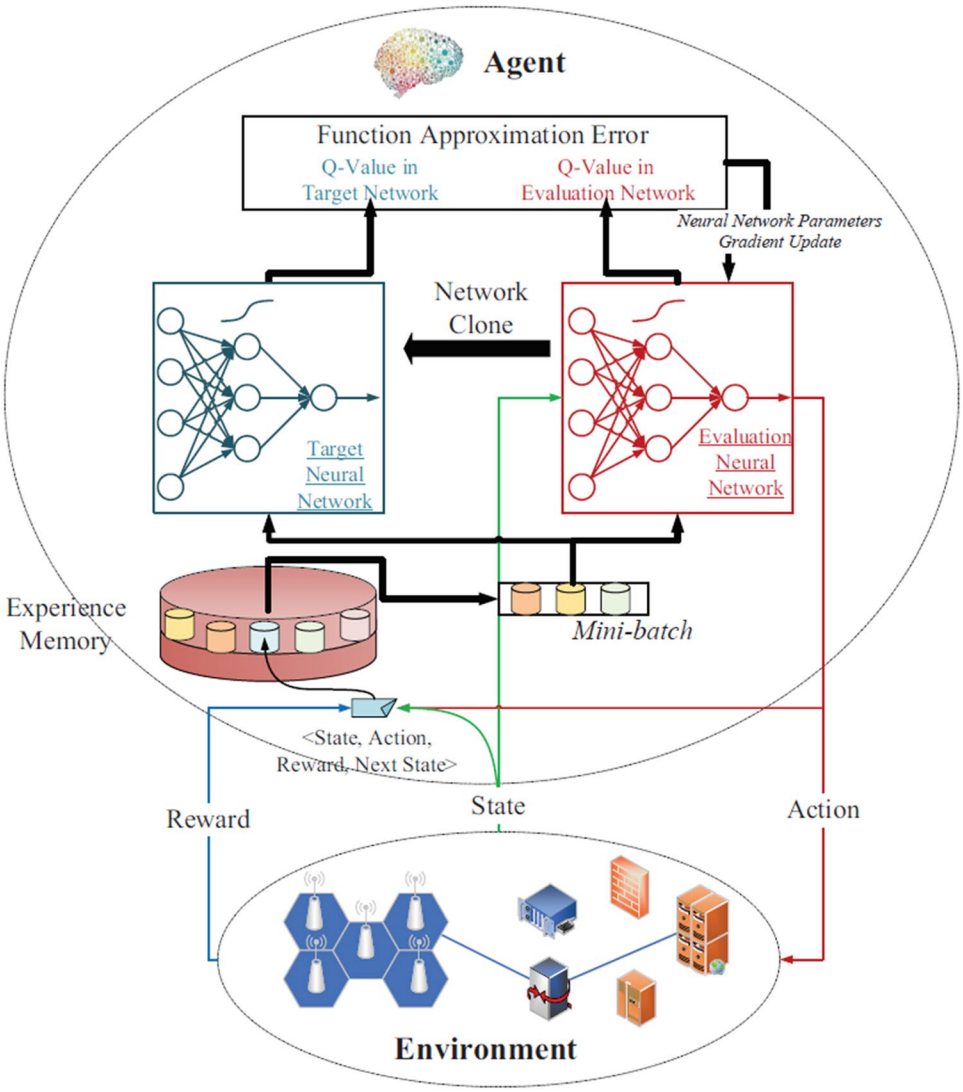


Fig. 7 Illustration of DQL [24]

and on no slicing, in which round-robin scheduling is conducted within all services. The simulation results show that DQL outperforms other strategies with a QoE of 81% compared with 15% and 41% for hard slicing and no slicing, respectively, for RAN resource management. For CN, DQL reduces flow average waiting time by up to 10.5% shorter than no-priority solutions (e.g., those that allocate SFC yielding minimum waiting time to flow) and increases CPU usage by up to 27.9%.

In [25], the authors propose an adaptive resource allocation based on the approximate dynamic programming (ADP) method to satisfy the dynamic changing characteristics of a slice load. The study adopts a non-orthogonal multiple access system that enables multiple users to be multiplexed on the same subchannel with different power levels. The authors formulate resource allocation as a constrained MDP problem, in which a resource scheduler (i.e., an agent) has multiple objectives; that is, to jointly optimize power allocation and channel assignment under average delay and outage probability constraints. As illustrated in Fig. 8, the resource scheduler uses an optimal policy to decide on actions for each slice in real time based on the monitored slice rate and queue that indicates the current system state (i.e., allocated power and subcarrier). Actions consist of adjusting the allocated power granularity to reduce slice queue and the allocated subcarrier to reduce interferences that may increase the outage probability of users. In return, the resource scheduler receives the slice weight, which is determined by the service demands of the slices and states of the slice queues, as a reward.

Similar to RL, ADP also automates the decision-making task through agent–environment interaction but uses vocabularies and notational systems from the operations research discipline instead of machine learning. Similar to [21], ADP uses an estimation of the expected reward over the long run (i.e., Q-value) wherein the estimated value

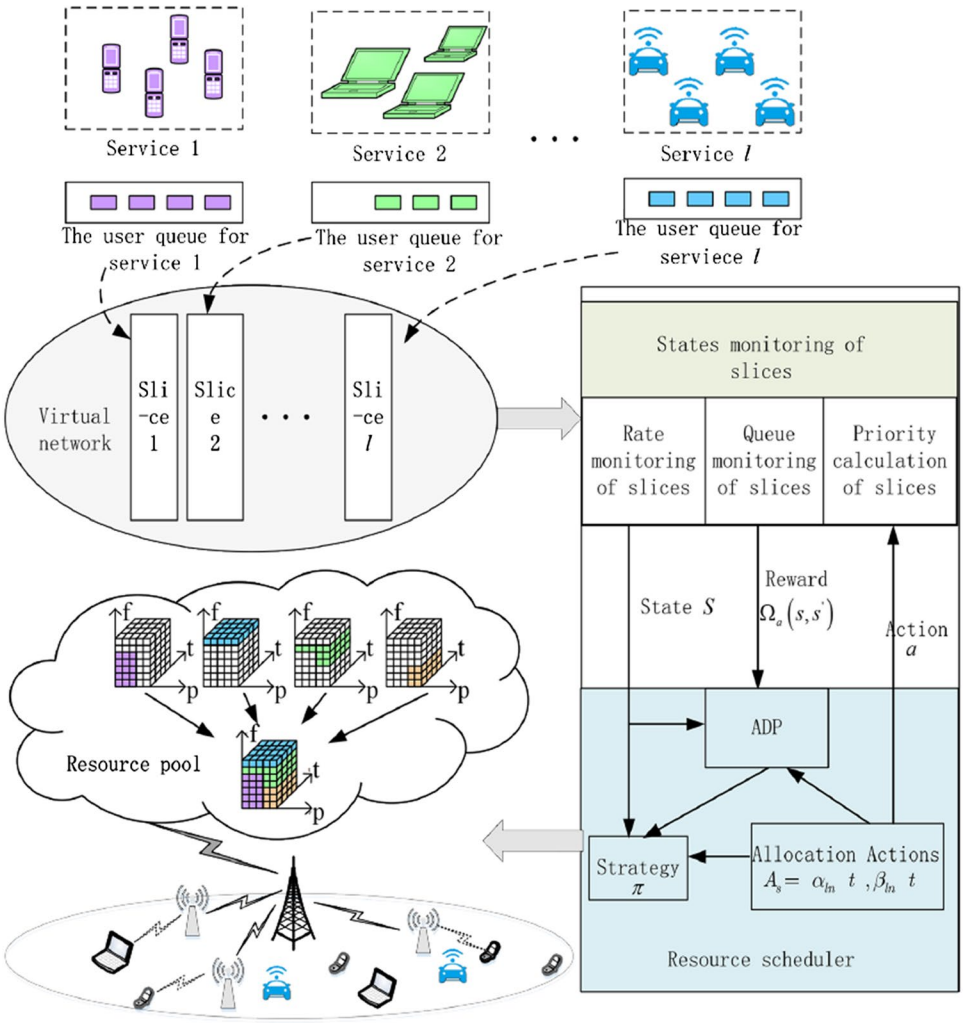


Fig. 8 System scenario [25]

is imported into a function at each time step to obtain an optimal policy, known as the value function. Furthermore, constraints are also imported into the value function using a Lagrange algorithm. In addition, to cater to a large system, ADP does not store the estimated values in a table but approximates them using a function approximator [30]. This function represents a function value via a linear combination of feature vectors that must be updated to obtain the estimated value. A gradient algorithm is used to update the parameter vector, and the objective is to minimize the mean square error between the estimated and approximated values.

The proposed ADP algorithm is evaluated against the Q-learning algorithm proposed in [21] by comparing total rates with increasing number of users. The simulation results show that ADP has a faster convergence rate and the maximum sum rate until a certain number of users, at which point the performance of the two algorithms equalizes. In addition, the comparison of the average queue caching with an increasing number of users shows that ADP has a smaller queue caching that grows rapidly after a certain number of users is reached due to increased outage probability.

4 Discussions

Table 2 presents the comparative analysis of various slice admission control mechanisms in terms of method, algorithm, state, action, reward, characteristics, strengths, and weaknesses. The mechanisms are classified into heuristic-based, value-based, policy-based, and evolutionary RL.

4.1 Objective and system modeling

The objective of all the solutions is to ensure that slice requirements are satisfied while maximizing either MNO's revenue or network utility, such as network throughput or user data rates. Their simulation setup considers two types of slice with different characteristics, such as priority level, elasticity, or resource cost. However, only [24] provides

Table 2 Resource allocation algorithms with type, strengths, and weaknesses

Refs.	Method/algorithm	State (S), action (A), reward (R)	Characteristics	Strengths	Weaknesses
[20]	Heuristic-based/two-tier priority	N/A	Heuristic with two-tier priority level	Better fairness among different slices	Less practical for real scenarios
[21]	Value-based RL/Q-learning	S: current system load, new slice type A: accept, reject R: revenue	Model-free Online Tabular	Guaranteed convergence Nonstationarity robustness Low variance	Limited scalability Biased
[22]	Policy- based RL/Policy gradient	S: current system state, holding time, amount of resources A: accept, reject R: revenue	Model-free Episodic Approximator (NN) Optimizer (gradient descent)	Highly scalable Better convergence Support stochastic policies Unbiased Low variance	No guaranteed convergence Requires training
[23]	Evolutionary RL/GA	S: resource feasibility space, new slice type A: accept, reject R: revenue, throughput, average or reliability	Model-free Episodic	Highly scalable Nonstationarity robustness	No guaranteed convergence
[24]	Value- based RL/DQL	RAN S: total new packets in each slice A: adjust allocated BW to each slice R: weighted sum of SE and QoE CN S: priority and time stamp of the last-arriving flows in each SFC A: adjust the allocated SFC to each flow R: weighted sum of RU and WT	Model-free Episodic Approximator (NN) Optimizer (gradient descent) Experience replay	Guaranteed convergence Nonstationarity robustness Highly scalable	Slow convergence Requires training
[25]	Value-based RL/ADP	S: slice rate, slice queue A: adjust the allocated power granularity and subcarrier to each slice R: slice rate, slice queue	Model-free Online Approximator (linear combination of features) Optimizer (gradient descent)	Guaranteed convergence Nonstationarity robustness Highly scalable	Requires training

solutions for RAN and CN resources. In [20, 21, 25], only RAN resources are included in the network model, whereas [22] involves only transmission and CN resources. In [23], all resource types are covered but a highly abstracted definition for maintaining generality is used, i.e., the heterogeneity of radio resources is disregarded. For the PS use case, we assume that the solution should consider at least RAN and CN resources and aim to maximize slice reliability and minimize delay in addition to throughput.

4.2 Algorithm type

All RL algorithms are model-free, thereby eliminating the need for off-line planning to learn system characteristics, i.e., reward and state transition models. In [21, 25], online algorithms that can immediately change policy at each time step are presented. Meanwhile, the algorithms proposed in [22–24] are episodic, i.e., they have to wait until the completion of a subsequence of agent–environment interaction or an episode to apply a policy change.

4.3 Scalability

A tabular-based method that evaluates the expected rewards of every single action to learn policies is proposed in [21]. This method has to use large tables in memory to store all possible Q-values, thereby increasing its complexity for an environment with an arbitrarily large number of states. A function approximator, i.e., an NN, is used in [22, 24], whereas a linear combination of features is used in [25] to generalize previously encountered states that are similar to unseen ones to reduce the number of states. In [23], function value is not evaluated but rather learned by directly searching the space of possible policies for one with the

highest fitness value. In [22], the space of policies is defined by a collection of numerical parameters and the algorithm estimates the directions in which the parameters should be adjusted to most rapidly to improve a policy's performance. Consequently, the algorithms proposed in [22–25] are highly scalable and applicable to large environments.

4.4 Guaranteed convergence

Although the algorithm presented in [21] is limited in scalability, its convergence to the global optimum is guaranteed. By contrast, the algorithms in [22, 23] typically converge to a local optimum rather than to the global optimum.

5 Challenges

5.1 Heuristic

Although the algorithm established in [20] can provide better fairness among different slices compared with the existing LTE allocation schemes, a heuristic-based solution requires thorough testing and tuning to achieve good performance in practice due to the simplified model used during planning, thereby making it less practical for real scenarios.

5.2 Value-based RL

Tabular-based methods, such as Q-learning, can be combined with a function approximator to improve its scalability to cater to large environments. In theory, any supervised learning method, such as NN, decision tree, or nearest neighbor, can be used. The challenge is to select the most suitable one that can guarantee that Q-learning convergence is maintained. Two approaches are explored in [24, 25]. In [24], DQL uses a function approximator, e.g., an NN, to approximate the Q-values to as close as possible to the target Q-values. In [25], ADP uses a function approximator to represent the function value via a linear combination of feature vectors that must be updated to obtain the estimated Q-values. Both solutions can improve value-based RL scalability while maintaining convergence. However, they both require a training method that is suitable for nonstationary, non-independent, and identically distributed data.

5.3 Policy-based RL

A policy gradient provides faster convergence but typically converges to a local optimum rather than to the global one. To mitigate this issue, a policy gradient can be extended to a value-based RL method; such combination is known as actor–critic RL [19]. Many forms of this combination are possible, namely, Q actor–critic, advantage actor–critic, and natural actor–critic. Each form leads a stochastic gradient ascent algorithm. These methods learn approximations of policy and value function, in which "actor" is a reference to the learned policy and "critic" refers to the learned value function. To address off-line inconvenience, actor–critic methods with a bootstrapping critic, i.e., an online value-based RL, such as temporal difference, can be used.

5.4 Evolutionary RL

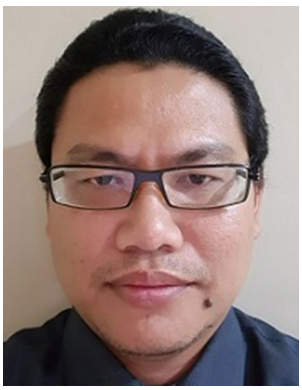
To address the convergence issue of GA, several advanced techniques, such as fitness scaling, diploid evolution, and sequence ordering, can be applied [23]. In addition, the latency impact of slice creation in current studies is not considered in [23]. Therefore, suitable techniques are required to mitigate this issue and consequently improve the slice QoS. Moreover, the abstract definition of resources is used in [23]; that is, only a 1D normalized resource pool is considered. In practice, heterogeneous physical radio resources, such as frequency bands and transmission power, must be distinguished with different orthogonal dimensions. Furthermore, any resource multiplexing over different slices is excluded in [23]. Such multiplexing is not only common in practice but also essential for realizing slice elasticity, particularly for physical resources.

6 Conclusion

In this paper, we present a review of efficient slice resource allocation solutions proposed in the recent literatures from the perspective of the PS use case. The solutions are classified into heuristic-based and RL-based. The manner in which each solution operates, along with their advantages and disadvantages, is also discussed. Furthermore, a comparative analysis is performed. The analysis indicates that all the proposed solutions focus only on maximizing either the MNO's revenue or network utility, such as network throughput, while disregarding other PS essential requirements, such as network reliability and latency, which are crucial for PS users. In addition, the solution should include at least 5G RAN and CN resources in the system model to provide end-to-end QoS guarantee. The flow-based 5G QoS control that enables intra-slice resource allocation should also be included in the solution to complement the inter-slice resource allocation of NS. Furthermore, the challenge in implementing RL-based algorithms is to find the most suitable one for our problem, i.e., an algorithm that is highly scalable, able to solve problems immediately, and exhibits the best convergence properties in terms of speed and the ability to find the global optimum.

References

1. Baldini, G., Karanasios, S., Allen, D., & Vergari, F. (2014). Survey of wireless communication technologies for public safety. *IEEE Communications Surveys & Tutorials, 16*(2), 619–641.
2. Kumbhar, A., Koohifar, F., Güvenç, İ., & Mueller, B. (2017). A survey on legacy and emerging technologies for public safety communications. *IEEE Communications Surveys & Tutorials, 19*(1), 97–124.
3. Avramova, A. P., Ruepp, S., & Dittmann, L. (2015). Towards future broadband public safety systems: Current issues and future directions. In *International conference on information and communication technology convergence (ICTC)* (pp. 74–79).
4. Yu, W., Xu, H., Nguyen, J., Blasch, E., Hematian, A., & Gao, W. (2018). Survey of public safety communications: User-side and network-side solutions and future directions. *IEEE Access, 6,* 70397–70425.
5. Ergul, O., Shah, G. A., Canberk, B., & Akan, O. B. (2016). Adaptive and cognitive communication architecture for next-generation PPDR systems. *IEEE Communications Magazine, 54*(4), 92–100.
6. 3GPP TS 23.203 v. 8.8.0, "Policy and Charging Control Architecture", Dec. 2009.
7. Alasti, M., Neekzad, B., Hui, J., & Vannithamby, R. (2010). Quality of service in WiMAX and LTE networks. *IEEE Communications Magazine, 48*(5), 104–111.
8. Höyhtyä, M., Lähetkangas, K., Suomalainen, J., Hoppari, M., Kujanpää, K., Ngo, K. T., et al. (2018). Critical communications over mobile operators' networks: 5G use cases enabled by licensed spectrum sharing, network slicing and QoS control. *IEEE Access., 6,* 1.
9. 3GPP TR 28.801 V1.2.0 (2017-05), "Study on management and orchestration of network slicing for next generation network (Release 15)", Jan. 2018.
10. Alliance, N. (2016). Description of network slicing concept. NGMN 5G P1 *Requirements & Architecture, Work Stream End-to-End Architecture*, Version 1.0.
11. 5GPPP, "View on 5G Architecture (Version 2.0)", 5G PPP Architecture Working Group, Athens, Jul. 2017.
12. Zhou, X., Li, R., Chen, T., & Zhang, H. (2016). Network slicing as a service: Enabling enterprises' own software-defined cellular networks. *IEEE Communications Magazine, 54*(7), 146–153.
13. Nakao, A., Du, P., Kiriha, Y., Granelli, F., Gebremariam, A. A., Taleb, T., et al. (2017). End-to-end network slicing for 5G mobile networks. *Journal of Information Processing, 25,* 153–163.
14. Ordonez-Lucena, J., Ameigeiras, P., Lopez, D., Ramos-Munoz, J. J., Lorca, J., & Folgueira, J. (2017). Network slicing for 5G with SDN/NFV: Concepts, architectures, and challenges. *IEEE Communications Magazine, 55*(5), 80–87.
15. Rost, P., Mannweiler, C., Michalopoulos, D. S., Sartori, C., Sciancalepore, V., Sastry, N., et al. (2017). Network slicing to enable scalability and flexibility in 5G mobile networks. *IEEE Communications Magazine, 55*(5), 72–79.
16. Han, B., Feng, D., & Schotten, H. D. (2018). A Markov model of slice admission control. *IEEE Networking Letters, 1,* 2–5.
17. ETSI GS NFV-INF 001 V1.1.1, "Network Functions Virtualisation (NFV); Infrastructure Overview", Jan. 2015.
18. ONF TR-521, "SDN Architecture Overview", Issue 1.1, 2016.
19. Sutton, R. S., & Barto, A. G. (2017). *Reinforcement learning: An Introduction. A Bradford book*. Cambridge: The MIT Press.
20. Jiang, M., Condoluci, M., & Mahmoodi, T. (2016). Network slicing management & prioritization in 5G mobile systems. In *22nd European wireless conference VDE 2016* (pp. 1–6).
21. Bega, D., Gramaglia, M., Banchs, A., Sciancalepore, V., Samdanis, K., & Costa-Perez, X. (2017). Optimising 5G infrastructure markets: The business of network slicing. In *IEEE conference on computer communications INFOCOM 2017* (pp. 1–9.).
22. Natalino, C., Raza, M. R., Rostami, A., Öhlen, P., Wosinska, L., & Monti, P. (2018). Machine learning aided orchestration in multi-tenant networks. In *IEEE photonics society summer topical meeting series (SUM)* (pp. 125–126).
23. Han, B., Lianghai, J., & Schotten, H. D. (2018). Slice as an evolutionary service: Genetic optimization for inter-slice resource management in 5G networks. *IEEE Access, 6,* 33137–33147.
24. Li, R., Zhao, Z., Sun, Q., Chi-Lin, I., Yang, C., Chen, X., et al. (2018). Deep reinforcement learning for resource management in network slicing. *IEEE Access, 6,* 1.
25. Tang, L., Tan, Q., Shi, Y., Wang, C., & Chen, Q. (2018). Adaptive Virtual resource allocation in 5G network slicing using constrained Markov decision process. *IEEE Access, 6,* 61184–61195.
26. Watkins, C., & Dayan, P. (1992). Q-learning. *Machine Learning, 8,* 279–292.
27. Mao, H., Alizadeh, M., Menache, I., & Kandula, S. (2016). Resource management with deep reinforcement learning. In *Proceedings of the 15th workshop on ACM HotNets*.
28. Moriarty, D. E., Schultz, A. C., & Grefenstette, J. J. (1999). Evolutionary algorithms for reinforcement learning. *Journal of Artificial Intelligence Research, 11,* 241–276.
29. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., et al. (2015). Human-level control through deep reinforcement learning. *Nature, 518*(7540), 529–533.
30. Powell, W. B. (2008). *What you should know about approximate dynamic programming*. Hoboken: Wiley.



Anuar Othman received the B.Sc. degree from University of Rouen, France, in 2001, and the M.Sc. degree from the University of Strathclyde, United Kingdom, in 2008. He is currently pursuing the Ph.D. degree with the National University of Malaysia, with a focus on Critical Communications Services in 5G. He has over 15 years of experience in telecommunications projects for mobile operators and public safety agencies. In recent years, he has been focusing on the convergence of narrowband and broadband radio communications for business and mission critical market.

Nazrul Anuar Nayan obtained his Bachelor of Engineering in Information & Communication Engineering, from The University of Tokyo in 1998, Master of Engineering in Electrical & Electronics and Doctor of Philosophy in Electronics and Information Systems Engineering from Gifu University, Japan in 2008 and 2011, respectively. In addition, he has also gone for a two-year (2014–2016) post-doctoral research programme at The Institute of Biomedical Engineering, Univ. of Oxford, United Kingdom. He was also a Research Member of Common Room, Kellogg College, Univ. of Oxford in 2015-2016. Nazrul started his career in engineering as a Designer at Hitachi Limited, Tochigi, Japan (1998–2000), then continued as a Test Engineer at Unisem Malaysia Berhad (2001–2003), Senior R&D Engineer at STATSChipPAC Malaysia (2003–2005) and as an Academic Researcher at Mimos Berhad, Kuala Lumpur (2012). His research interests lie in the field of Big Data in Healthcare (Biomedical Signal Processing), Digital Integrated Circuit Design and Computational Thinking. In 2013, he was appointed as a Professional Engineer with Practicing Certificate (Electronics: C115815) by the Board of Engineers, Malaysia.