IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 23, 2019, accepted January 10, 2020, date of publication January 20, 2020, date of current version January 27, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2967626

A Survey on Slice Admission Control Strategies and Optimization Schemes in 5G Network

MOURICE O. OJIJO AND OLABISI E. FALOWO, (Senior Member, IEEE)
Department of Electrical Engineering, University of Cape Town, Rondebosch 7700, South Africa

Corresponding author: Mourice O. Ojijo (ojjmou001@myuct.ac.za)

This work was supported in part by the Telkom South Africa, and in part by the Jasco Group via the Telkom Center of Excellence.

ABSTRACT The Fifth Generation(5G) communication network is envisioned to provide heterogeneous services tailored to specific user demands. These services are diverse and can be generally categorized based on latency, bandwidth, reliability, and connection density requirements. The 5G infrastructure providers are expected to employ network function virtualization, software-defined networking, and network slicing for cost-effective and efficient network resource allocation. In the 5G network, when an infrastructure provider receives a slice request, a slice admission control scheme is applied and an optimization algorithm is used to achieve predefined objectives. To this end, a number of slice admission control objectives, strategies and algorithms have been proposed. However, there is a need to present a coherent review and bridge the gap between many aspects of slice admission control. In this paper, we present the latest developments in this research area. Thus, we begin by introducing slice admission control and discuss background concepts associated with slicing. We then extend our discussion to slice admission objectives followed by the strategies and optimization algorithms. Finally, we conclude with a summary of analysis containing the optimization algorithms.

INDEX TERMS 5G, network slicing, slice admission control, resource allocation.

I. INTRODUCTION

The Fifth Generation (5G) wireless network will provide non-monolithic services by incorporating the concept of advanced resource management. Unlike the 4G network, the 5G infrastructure comprises the following important features: network function virtualization (NFV), software-defined networking (SDN) and network slicing. Furthermore, network slicing is a principal enabler of network resource management and allocation [1]. It allows multiple logical networks to be deployed on a common physical infrastructure in order to provide the quality of service (QoS) required for divergent applications [2], [3]. The logical network provides resource flexibility and isolation which can be easily customized. In the 5G network, efficient resource allocation and reliable connectivity is achieved using slice admission control algorithms which influence slice admission decisions. In this regard, slice requests can be efficiently processed and resources allocated for service launching. The processed of launching network functions is called service orchestration [4]. An orchestrator is responsible for instantiating, supervising and managing virtual network functions(VNFs) [5].

Typically, the underlying objective in the 5G network is to virtualize network functions thereby separating network control and data forwarding planes. This is achieved through network softwarization. SDN and its complementary NVF are the main enablers of 5G network softwarization. Indeed, most of the VNFs can be easily instantiated from off-the-shelf commodity servers, which enable fast and efficient control; thereby leaving data forwarding to the underlay physical substrate and fulfill the test of reliability. For instances, some researchers have proposed the use of aerial base stations embedded on unmanned aerial vehicles (UAV) [6]. This is aimed at providing high-quality connectivity in areas with flash crowd problems; consequently, realizing reliable connectivity through well-defined network slices.

A slice admission control algorithm is essential for efficient management of network resources in the 5G network. The slice tenants (STs) or virtual network operators (VNOs) (who are the third-party service providers and own no physical network); control, manages and sell virtual services and

The associate editor coordinating the review of this manuscript and approving it for publication was Amjad Ali.

the infrastructure network providers (InPs) (who owns and operates the physical network), rely on slice admission control to achieve their predefined objectives [7].

The InP may decide to admit slices deemed to have the best chance of meeting the pre-defined objectives. Slice admission is also dictated by the available resources in the network resource pool as illustrated in Figure 1. Slice requests may be queued while the implemented admission algorithm queries admissibility fitness considering available resources. Slice admission is always succeeded by efficient service chain embedding [8]. Service chain embedding is the mapping of VNFs to the physical network. Efficient embedding promises to reduce operation cost and dynamically create autonomous functions with high mobility and rapid deployability.

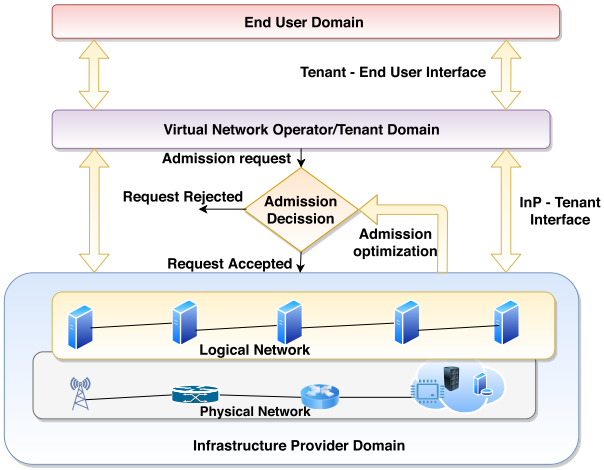

FIGURE 1. Slice admission control illustration.

Network operators generally process a vast amount of data from their customers. This data comprises information regarding the customer network demands [9] which can be translated into the type of slice requested. Considering this, the International Telecommunication Union [ITU] has defined three broad categories of slices namely ultra-reliable low latency communication (uRLLC), massive machine type communication (mMTC) and enhanced mobile broadband (eMBB) slices. Although a slice may belong to any of the above categories, slice requests are yet to be standardized and may not be discretely defined. Besides, the admission control algorithm must intelligently consider the slice request and determine its fitness to meet the predefined objectives. Subsequently, the algorithm correlates a slice request with an objective and then performs admission control. The following subsection presents four main slice admission control objectives in 5G network.

A. SLICE ADMISSION OBJECTIVES

Slice admission control algorithms are deployed to achieve specific objectives. The four main objectives of slice admission control algorithms are: revenue optimization, QoS control, congestion control, and admission fairness. These objectives are discussed as follows.

a. ***Revenue optimization***: The primary goal of an ST and InP is to generate revenue [8] [10]. Thus, any slice provider can decide to apply admission control algorithm to allocate network resources so as to maximize revenue. A slice provider, for instance, may prioritize latency and high bandwidth for high revenue.

b. ***QoS control*** [11]: QoS provisioning is the process of providing and maintaining a level of service based on customer service level agreements (SLAs). In order to maintain the availability of resources for QoS-intensive network slices, the slice provider may decide to deploy slice admission control algorithm to admit requests whose QoSs can be guaranteed [10], [12]. To further this concept, consider an instance of a natural disaster such as massive earth quake, flood or Tsunami, resulting in massive communication breakdown thus limiting service availability, consequently creating a high demand for emergency communication services. The slice service provider may decide to reject slice requests; except for mission-critical services, thus, reserving more resources to ensure intended adjustments are always available for admitted slices, consequently, guaranteeing acceptable QoS and quality of experience (QoE) for these mission-critical services.

c. ***Inter-slice congestion control*** [13]: The 5G network is expected to experience big data explosion. When slice requests are directly channeled to the ochestrator for virtual function instantiation without a slice admission control, congestion may occur. When slice admission control is applied, the slice provider prevents the orchestrator from getting overwhelmed by slice requests which would otherwise be rejected. Such slice requests are deemed to have a high probability of dropage as determined by the admission control algorithm. This probability is defined as the inverse of likeliness to meet the admission control objective. For instance, in priority based admission control, only high priority slices may be admitted to limit congestion. This can significantly reduce the number of admitted slices and subsequently de-congest the resource request queue.

d. ***Slice fairness assurance*** [14]: In many instances, the slice providers may organize slices based on the type: i.e. eMBB, uRLLC and mMTC. These slices may be placed on separate queues respectively. An admission control algorithms is applied for reasonably distribution of slice requests such that no single slice type is repeatedly admitted at the expenses of other slice requests. Generally, a round-robin queue consideration may be adopted to reduce unfairness during selection. Multi-queue admission control works well when slice priority is not the main consideration.

B. SLICE ELASTICITY

Slice admission control algorithm can be designed to deal with resource arbitration and to determine whether a slice is elastic or inelastic. An elastic slice can be adjusted dynamically depending on the demand for resources without affecting the QoS provisioned. An inelastic slice requires rigid resource allocation that can not be adjusted downward at any moment in the duration of resource provisioning; such slices can only be scaled up [15]. For instance, uRLLC slice requires strict resource allocation to guarantee reliability and low latency while eMMB resources may be scaled down once the edge cloud buffering is complete. Some researchers have attempted to propose algorithms that fit many admission considerations. However, the designs are generally too complex and require vast assumptions to implement. Notably, the radio access network (RAN) resource allocation for uRLLC services are mostly considered to be inelastic and more complex to provision. The cloud resources are nonetheless considered to be elastic and are much easier to provision. The combination of RAN, core and cloud logical resource allocation is known as end-to-end slicing. Elastic resource slicing enables providers to achieve flexibility in creating, adding and terminating virtual functions [2]; consequently, minimizing the capital cost while allowing the ST to accommodate more users. Inelastic slices are more complex to manage and maintain because they are subjected to stringent SLAs [16]. The QoS in an inelastic slice may be vastly affected with any attempt to adjust provisioned resources thereby leading to huge penalties proportionate to the degradation of a slice. On the contrary, data loss in elastic slices can always be accommodated as service degradation is not explicitly visible. Slice admission control plays a key role in reducing the effects attributed to slice elasticity.

C. SLICE TENANCY

An ST can acquire heterogeneous virtual resources from the InP as shown in Figure 2 and sell to any of the three user classes i.e. (eMBB, uRLLC, mMTC). In the 5G network, slice tenancy can be classified as single tenant or multiple tenant. In single tenancy only one tenant is contracted by an InP whereas in multi-tenancy, many tenants are contracted. The simplest form of slice admission control is where only one tenant is contracted. In Monti *et al.* [17], a single tenant is assumed to generate multiple slice requests with varying priorities and slices are admitted according to priority (highest priority first and lowest priority last).

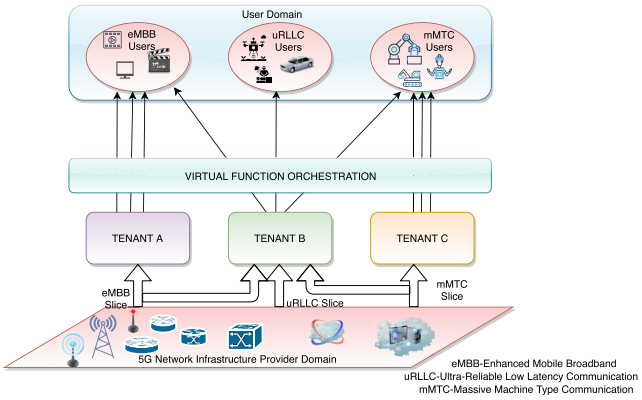

FIGURE 2. An illustration of slice multi-tenancy.

A tenant may send slice requests regularly, randomly or follow some predetermined probability distribution. The price paid by the tenant depends on the priority index of the slice. High priority slices are more expensive than the low priority slices and are much harder to provision and maintain. Each tenant may provide specific slice requirements to be met by the slice provider, this is subject to the a set of predetermined SLA. The following are the major considerations in multi-tenant slice admission control schemes.

a. ***Fairness***: Ensuring fairness in multi-tenant slice admission control schemes is a major challenge for slice service providers. If the choice of admission is biased, tenants may choose to cancel the slice requests when subjected to longer waiting duration [14]. An efficient admission control algorithm, must therefore, ensure some degree of fairness when dealing with multiple tenants such that no request is dismissed without consideration.

b. ***Resource competition***: The 5G network, like other wireless networks, does not have unlimited resources. Competition for resources is thus envisaged to occur among tenants. When more than one tenant request for the same resources, a slice provider must employ a request brokering scheme to deal with such competition [18]. An artificial-intelligent (AI) based slice admission algorithm may be applied in choosing a tenant to serve and which to put on hold. The AI algorithm may be trained to learn which tenant is more reliable and has a high willingness to pay.

c. ***Revenue accumulation***: The VNO associates slice admission with revenue aggregation. An optimum slice admission control strategy may directly translate to an InP accumulating more revenue [19]. Hereof, multiple tenants provide a chance for the slice provider to sell variety of slices. Undoubtedly, when InPs classify tenants according to their willingness to pay, the slice provider may be forced to maintain a greedy policy. This is because greedy algorithms may seem profitable albeit in short term. Research has shown that a combination of greedy and semi-greedy policy performs better in the long-term [20].

A well-designed slice admission control algorithm is can maintain the right balance among the aforementioned considerations while achieving the overall design goals.

D. SLICE ADMISSION DOMAIN: INTER-SLICE VERSUS INTRA-SLICE

Slice admission domain can be classified in two categories namely: *inter-slice admission and intra-slice admission domain*. In inter-slice admission domain, two or more heterogeneous tenants are involved. The tenants send independent

slice requests for isolated and distinct logical resources while in intra-slice different end-users send their request to a tenant so as to be admitted into a common slice with shared resources [12], [21]. Figure 3 is an illustration of this concept. The InP deals directly with tenants and can only admit their slice requests. The tenant in turn admits end-users into the acquired slice for a period of time. The challenges of inter-slice interference must be solved by the InP, while user isolation problems in a common slice must also be addressed by the tenant.

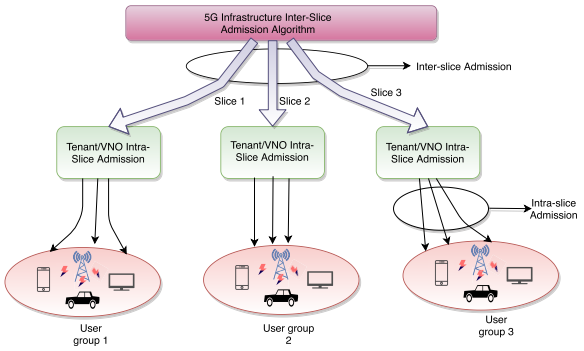

FIGURE 3. An illustration of inter-slice and intra-slice admission.

In intra-slice domain, resource limitation is the main cause of user request rejection. For instance, a virtual gamer may request to be admitted into an eMBB slice, while bandwidth demands may be guaranteed, low latency constraints are difficult to guarantee here, hence, such request may be automatically rejected by an admission control algorithm. On the other hand, in inter-slice admission control, each slice request is evaluated independently and only admitted when the InP can guarantee resource allocation, SLA and QoS [7].

E. SLICING DOMAIN:END-TO-END, RAN, CORE AND CLOUD SLICING

Network slice orchestration can be undertaken at different domains in the 5G network namely: RAN, transport, cloud or end-to-end. In Figure 4, the RAN slicing involves virtual radio resource allocation in the access network and is affected the number of subscribed users [22]. Moreover, grouping users according to slices they are subscribed to, may reduce intra-slice interference [23]. The RAN users are allocated access bandwidth, shared radio frequency (RF) antennas, media access address (MAC), RAN signaling, etc. RAN slices is enabled by radio resource controller (RRC) and radio link controller (RLC). RAN slicing, however, poses challenges such as maintaining fairness, slice isolation, and SLA monitoring [23].

The transport network consist of high bandwidth optical cores [24], and can be physically sliced or virtually sliced. Physical slicing involves allocating each fiber core to a single tenant while virtual slicing is where more than one tenant share the same core. Tenant isolation within the core is achieved through end-to-end encryption.

The core cloud resources are high-end general processing units (GPUs), high capacity storage and random access memory (RAM). Cloud slicing refers to allocating these shared resources among tenants and allowing separate VNFs to be instantiated and run independently.

An end-to-end slicing involves a virtual resource allocation in the RAN, transport and core cloud. The work in [4] is an illustration of this concept where an mMTC slice for

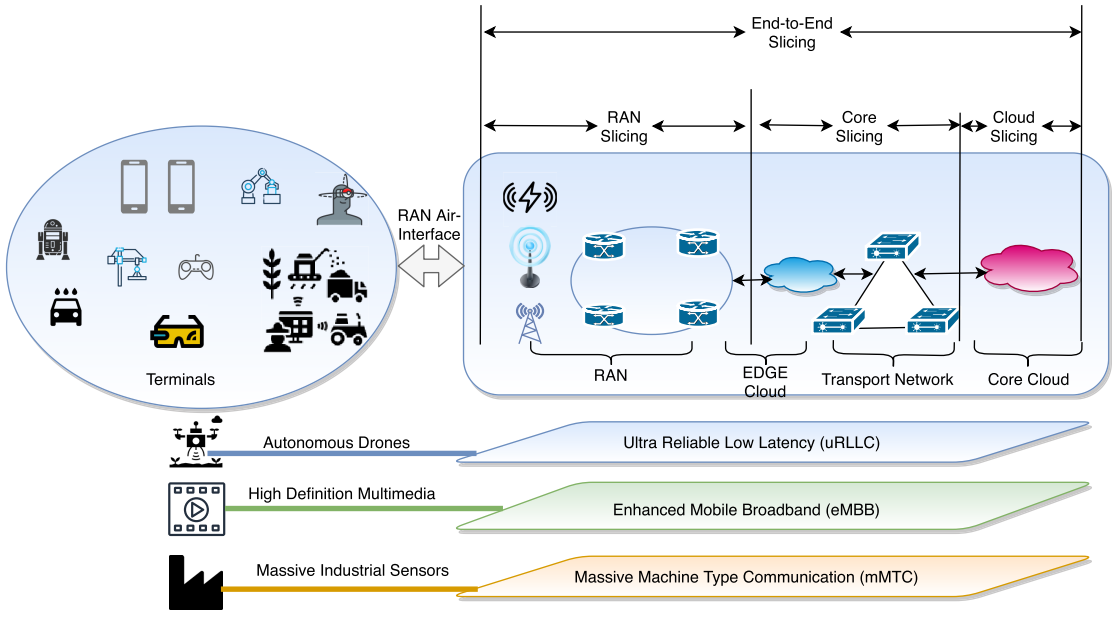

FIGURE 4. End-to-end, RAN, and cloud slicing.

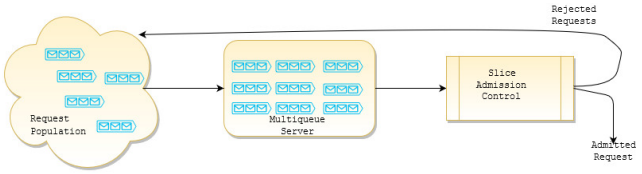

FIGURE 5. Slice requests in multiqueue environment.

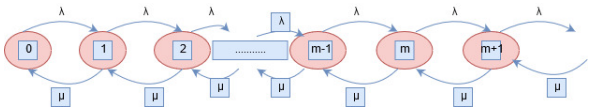

FIGURE 6. Markov chain with a birth and death process or admission requests.

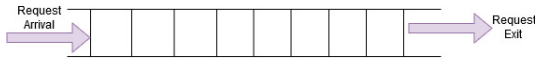

FIGURE 7. Request arrival and exit.

fourth industrial revolution (4IR) is proposed. The slice is expected to provide real-time production information, monitoring, actuation, sensing, and advanced remote control to both industrial production company (IPC) offices and industrial equipment vendor (IEV). In this setup, all the end-to-end resources must be sliced with regard to the user demand. Admitting such a slice by an InP is more challenging because an end-to-end resource availability must be guaranteed every time a request is received. Moreover, the InP has to secure connectivity, bandwidth and required latency throughout the slice duration. Problems associated with multi-domain access must also be addressed. Cloud slicing is however simpler since many of the resources can be discretized and allocated optimaly [25].

F. SLICE REQUEST QUEUING

The InP may receive a vast amount of heterogeneous slice requests simultaneously. A proper queue management is therefore necessary [14]. Many of the slice request arrivals are non-deterministic, where the average waiting time (τ) must be short enough such that no request is canceled before it is attended to.

A generic queue management strategy is shown in Figure 5. A request population is a stochastic distribution of slice requests and have an arrival rate (λ) determined by a known probability distribution (Poissonian). The queue pool is managed and controlled by a queue management algorithm generally influenced by the slice admission control policy. Intuitively, the slice admission control algorithm employed may incorporate any of the existing admission strategies for slice request evaluation. An efficiently managed queue reduces the cost of slice provisioning and improves tenant satisfaction. Several queuing approaches exist and can be employed in slice admission control. We discuss them as follows.

a. ***M/M/1 Model***: This is a Poissonian model with a single queue management server, suggested by Dharmaraja *et al.* [26]. The requests are processed on first in first out (FIFO) basis. The model name is written in Kendall's notation where the first M in the notation represents the arrival rate, the second M represents the service rate and 1 represents the number of queue management servers [25]. Explicitly, in slice admission control, this strategy is the most applicable model because multi-server model introduces further complexity and is difficult to coordinate. If λ is the request arrival rate, and μ is the request service rate, then the traffic intensity or occupancy of the model is an exponential term $\rho = \lambda/\mu$. Moreover, the model can be described as a continuous Markov chain with a birth-death process over a state space diagram shown in Figure 6. Figure 7 demonstrates the arrival and exit of a request according to [27]. Nonetheless, queue is of infinite length where the management server can only process a limited window w for a duration δt. Indeed, the slice request residency within the management server is longer compared to the multi-queue multi-server environment.

b. ***M/M/1/N Model***: This is a finite queue model where N denotes the queue size. This model shares similar characteristics as M/M/1. It is appropriate to define the queue size because the slice admission control algorithm may be hard-coded with a fixed set of requests for processing at a particular time.

c. ***M/M/c Model***: This is a multi-server model applied in [14] for admission control, where c denotes the number of servers in the system [28]. The servers may be identical with similar processing power or heterogeneous with varying speeds and efficiency [26]. While the request arrival rate λ maintains Poisson's distribution, the processing rate μ increases by a factor c [29] such that the traffic intensity $\rho = \lambda/c\mu$. In the event any of the queues are empty, some servers may be rendered idle. However, the request processing efficiency improves while the traffic intensity is reduced by the same factor c. The transition probabilities in the Markov process becomes more complex to obtain for this model [30]. The $M/M/c$ model solves mainly three problems. *1. Balking*: This is when the request leaves as a result of long queues. *2. Renege*: This is where the ST cancels a request after being in the queue for too long and lastly. *3. Jockey*: A slice request may be moved from one queue to another based on any of the reasons mentioned or when a server is idle. Other queuing models do exist albeit not relevant to our discussion, they include $M/M/c/K/K$ where each server is allocated only a limited number of requests to service. $M/D/1$ is a finite time service server while $D/M/1$ is a model with a deterministic request arrival, and $M/E_k/1$ is a model that follows Erlang distribution.

II. SLICE ADMISSION STRATEGIES

The design of any slice admission algorithm is based on a specific strategy. A strategy may be random, greedy, semi-greedy, priority-based or simply first-come-first-served. Whichever is the method used, the ultimate goal is to achieve an objectives defined by the slice operator. Figure 8 is a graphical representation of these classifications.

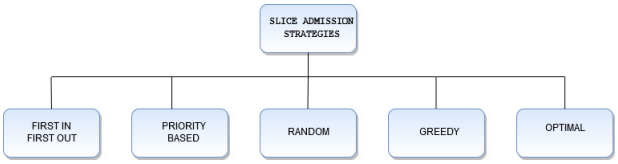

FIGURE 8. Slice admission strategies.

A. FIRST-COME-FIRST-SERVED

The simplest form of slice requests consideration is to admit them as they arrive. This first-come-first-served strategy is suggested by Han *et al.* [14]. Certainly, the scheme does not consider non-trivial constraints such as latency, and bandwidth. Any request arriving when there are no resources will automatically be rejected as such admissions may cause severe network degradation. Moreover, the admitted slice requests must lie within the admissibility region derived by *Bega et al* [15], thereby obeying resource constraints limits. Despite its simplicity, first-in-first-out is not popular because of lack of optimization method applied during slice request admission.

B. PRIORITY BASED SLICE ADMISSION CONTROL

When optimizing slice admission control, the slice providers may choose to give preference to a certain category of slice requests. These requests may have strict latency requirements (uRLLC slices). Such slices are considered to be of high-priority and thus more expensive [31]–[33]. Cumulatively, consistent admission of high priority slices results in higher revenue. For instance, in [17], a priority-based slice admission control is suggested. The author proposes an Reinforcement Learning (RL) method for revenue optimization considering latency requirements in the RAN. Indeed, this approach may improve revenue compared to random and first-come-first-served schemes. STs buying high priority slices enjoy superior QoE and network utilization [12]. Such slice admission strategies use meticulously designed optimization algorithms. While high priority slices are known to contribute higher revenue, short-term contracts may be less profitable. For instance, a slice provider may consider admitting slices for autonomous driving while rejecting other requests. Such contracts may be scarce and limited only to certain regions. On the contrary, slices designed for entertainment are always on high demand and are contracted for a longer duration thereby generating more revenue.

C. GREEDY BASED ADMISSION CONTROL

Priority-based slice admission strategy may improve overall admission objective, but it is highly dependent on the frequency of such contracts and the total slice duration. Its performance is highly affected when such contracts are limited and only last for short duration. A greedy policy proposed by Challa *et al.* [34] deploys a partial adaptive greedy (PAGE) algorithm to maximize revenue while minimizing SLA violation for customers with different willingness to pay indices.

As an illustration, a greedy based algorithm exploits a policy π that always meet a slice admission objective. This policy is determined after several iterations or learning process. It is used to tune a well defined objective function to exploit the evaluated policy without consistent exploration. An example of this strategy is discussed in [19]. The slice provider iteratively adds slice request to a temporary set S^+ if its probability of generating high revenue is considered to be higher. Conversely, another temporary set S^- holds slice requests with a low probability of generating higher revenue. A binary variable $\gamma \in \{0, 1\}$ is considered such that $\gamma = 0, \forall s \in S^-$ when a slice request is rejected and $\gamma = 1, \forall s \in S^+$ when a slice is admitted. Intuitively, a greedy-based strategy may not be optimal always [35]. A semi-greedy method allows the learning agent to explore other requests occasionally with a small probability ϵ, such that $1 - \epsilon = greedy$. An example of a greedy policy is the multi-armed bandit model. The reader is referred to [20] for more insight into the greedy policy and multi-armed bandit model.

D. RANDOM ADMISSION CONTROL

Random slice request admission may be adopted in order to reduce unfairness during admission control [7], [21]. It achieves a fairly normal distribution over a long period. Nonetheless, random admission control is not popular because no optimization is involved. Also, there is a lack of policy. Modern intelligent algorithms may be adopted to deal with unfairness in slice admission control.

E. OPTIMAL ADMISSION CONTROL

An optimal policy is adopted in order to achieve the best slice admission strategy by employing a well-tuned slice admission control algorithm. The idea is to define admission control objective and perform a complete optimization. Machine learning techniques such as in [10], and [36], have been used to obtain optimal slice admission control based on revenue maximization. Congestion control is another objective discussed in [13]; the authors have adopted a Q-learning method to learn an optimal strategy for slice admission control. Although machine learning methods may occupy extended duration during training and occasionally require retraining, good results are still achievable. Successive convex approximation (SCA) and alternating direct method of multipliers (ADMM) have also been applied for optimal slice admission control [7], [16]. Optimal algorithms are more complex to design, and ample tuning is required in order to achieve better results.

Table1 provides a summary of slice admission strategies and their corresponding strengths and weaknesses.

TABLE 1. A summary of slice admission strategies: Strengths and weakness.

Slice Admission Strategy	References	Strength	Weaknesses
Firs-Come-First-Served	*Han et al* [14]	Simplest to implement. Less computational cost.	No optimization adopted. Lacks efficient resource utilization.
Priority Based	*Monti et al* [17] *Kammoun et al* [31], *Caballero et al* [32]	Allows for selective admission. Highly exploitative.	High priority slices may be rarely contracted. Optimization may be sub-optimal, No admission fairness.
Random	*Challa et al* in [34]	Enhances slice admission fairness, Less computational cost.	No optimization involved. Lacks efficient resource utilization.
Greedy	[7], [21],	Highly exploitative.	Optimization process may be sub-optimal.
Optimal	[7], [21],	Enables both exploitation and exploration.	May be computationally intensive.

III. SLICE ADMISSION CONTROL OPTIMIZATION

A number of optimization techniques have been proposed in many literature for enhancing the performance of slice admission control algorithms in the 5G network. In the following subsection, these algorithms are discussed.

A. SUCCESSIVE CONVEX APPROXIMATION (SCA)

Many optimization problems in a 5G slice management are formulated as either integer linear or mixed-integer linear or non-linear integer programming. Such techniques are mathematical models where some constraints are restricted to integers or non-integer. The objective function is formulated as a linear or nonlinear problem integer programming. These formulations tend to be N-P hard and demand considerable relaxation to solve. SCA is one of the most popular methods used in solving such problems. In general, SCA involves iteratively optimizing the objective function $f(x)$, where x is closed form set of vectors, under strict convex constraints [8], [37].

For instance, consider

$$\max_x f(x) \triangleq g(x) + w(x) \tag{1}$$

$$\text{s.t} f(x+1) \triangleq g(x+1) + w(x+1) \leq 1 \quad \forall x \tag{2}$$

where the functions $g(x)$ and $w(x)$ are linear and nonlinear respectively or vice versa. The loss function $x \leftarrow x - \Delta x$ is iteratively reduced until convergence occurs. A simplified SCA algorithm based on (1) and (2) is given in **Algorithm 1**

Algorithm 1 An Illustration of SCA Algorithm

```
find feasible point for x = 0
set the step size Δ ∈ [0, 1]
set the loop counter r = 0
while f(x) ≠ MAX do
    set x^r = solution
    max f(x, x^r)
    check f(x + 1) ≜ g(x + 1) + w(x + 1) ≤ 1∀x
    r = r + 1
    x^{r+1} ← x^r − Δx
end whilef(x) = MAX
```

In slice admission control the functions $g(x)$ and $w(x)$ may be considered as resource blocks in the RAN and core networks respectively, and must not exceed certain limits for different slice requests. The work in [8] employs SCA in order to optimize slice admission control problem with SCA. Interestingly, while SCA strives to realize global optimization the mathematical complexity involved is quite intense. SCA can be combined with techniques such as semidefinite relaxation to solve combinatorial problems as suggested in [19].

1) COMPLEXITY ANALYSIS

The mathematical complexity of integer linear or integer non-linear programming problems depend on the number of variables, the number of constraints in each block, the entries in the objective functions, the entries in the constraints matrix and the constrains limit [38]. Consider the following optimization problem according to [37]

$$\min_x f(x) = \sum_{i=1}^{n} g_i(x_i) + \sum_{i=1}^{n} w_i(x_i)$$
$$s.t\ x_i \in X_i \tag{3}$$

the function $f(x)$ is treated as smooth and convex and assuming that $f(x)\ \forall\ r = 0$ to R satisfy the following conditions

- $f(x)$ is continuous in x
- $f(x)$ is convex in x
- $f(x+1) \triangleq g(x+1) + w(x+1) \leq 1 \forall x$
- Function value consistency: $f(x, x) = f(x), \forall x$
- Gradient consistency: $\Delta f(x)(x) = \Delta f(x), \forall x$
- Upper-bound: $f(x) \geq f(x), \forall x$

the complexity analysis can be stated as: Assuming that $f^*(x)$ is the optimal objective value and $x|f(x) \leq f(x_0)$ is compact. Then $f(x^r) - f(x) = f^*(x) = \mathcal{O}(\frac{1}{r})$ [37] is the rate of convergence. By improving r the rate of convergence can be improved as well.

B. ALTERNATING DIRECTION METHOD OF MULTIPLIERS (ADMM)

Problems that require decomposition and distribution such as dynamic resource allocation can be solved using ADMM [39]. However, ADMM is suited for continuous problems and may not be applicable to discrete problems.

Nonetheless, ADMM is highly adaptive in large scale distribution problems [40]. A variant of ADMM known as alternating direction dual decomposition (ADDD or AD^3) is adopted by the authors in [40] to improve stability and convergence for VNF admission control and function chain embedding (FCE). To illustrate the working of ADMM we adopt a simplified example. Assuming we want to optimize a convex function such that $\min_{x,y} v(x) + h(y)$ subject to $Mx + Ny = b$, we have two sets of functions with separable objectives. Where x and y represents two types of resources considered in minimizing the objective function, M and N are coefficients determining the ratio of usage while b is the maximum usage of x and y. The loss function can be formulated using Lagrangian approximation $L(x, y, z,) = v(x) + h(y) + z^T(Mx + Ny - b)$. We successively update x, y, z such that $x(t+1) \leftarrow \arg\min_x L(x, y, z)$, $y(t+1) \leftarrow \arg\max_y L(x, y, z)$ and $z(t+1) \leftarrow z + Mx + Ny - b$ which is a dual update. In slice admission control, the objective function may represent the objective for admitting a slice, while the closed form constrains indicate the limited resource and the loss function is minimized for optimality. A simplified version of ADMM is given in **Algorithm 2**

Algorithm 2 An Illustration of ADDM Algorithm for Slice Admission

find feasible point for $x(0) \in X_1 \times \ldots \times X_k$;
find feasible point for $y(0) \in Y_1 \times \ldots \times Y_k$;
repeat
  for for $i = 0$ to k **do**
    $x(r+1) \leftarrow \arg\min_x L(x, y, z)$
    $y(r+1) \leftarrow \arg\max_y L(x, y, z)$
  end for
  update $z(t+1) \leftarrow z + \Delta^r(\sum_{i=1}^{k} Mx_i^{r+1} + Ny_i^{r+1} - b)$
  $r = r + 1$
until convergence

1) COMPLEXITY ANALYSIS

When an optimization problem is a dual block, ADMM can be adopted to solve it. Consider the following optimization problem

$$f(x) = \min_{x,y} v(x) + h(y)$$
$$s.t \;\; Mx + Ny = b \;\; \forall \;\; x \in X \;\; and \;\; y \in Y, \qquad (4)$$

the simplified Lagrangian approximation is given by

$$L(x, y, z,) = v(x) + h(y) + z^T(Mx + Ny - b) \qquad (5)$$

A typical ADMM problem with the dual update considered in **Algorithm 2** has a convergence rate given by the gradient accent [41] $z(t+1) \leftarrow z + \Delta^r(\sum_{i=1}^{k} Mx_i^{r+1} + Ny_i^{r+1} - b) = \mathcal{O}(\frac{1}{r^2})$

C. HEURISTIC APPROACH

A heuristic approach is a broad range of computational techniques that generally aim to solve problems faster while sacrificing optimality and accuracy. In slice admission control, a heuristic approach may be adopted in cases where a strict optimal solution is relaxed but still maintain some good degree of sufficiency. Such solutions are not mathematically intensive. As an illustration, assume a slice admission based on a priority index which defines the QoE as depicted in [12], and [22]. Let us denote a slice $s_n \in S \;\; \forall \;\; n = (1, 2, 3, ..S)$ at time t and a priority index p_s normalized by the Boltzmann distribution such that $\sum_{s \in S} p_s = 1$. To derive p_s, a set random slice request parameters have been passed though dimensionality reduction. The aim is to determine a single parameter for sorting. Let us assume a priority threshold λ such that any slice request with a priority larger than the threshold will be admitted subject to resource availability. A simplified heuristic algorithm may be written as shown in **Algorithm 3**

Algorithm 3 An Illustration of a Heuristic Algorithm for Slice Admission

for $n = 1$ to N **do**
  calculate $p_{s_n} = \frac{e^{sn}}{\sum_{n=1}^{N} e^{sn}}$
  output p_{s_n} for $s_n \in S, \forall n = (1, 2, 3 \ldots N)$
end for
γ = minimum threshold
for $n = 1$ to N **do**
  check resource availability
  if $p_{s_n} \geq \gamma$ **then**
    ADMIT
  else
    REJECT
  end if
end for

From Algorithm 3 we can see that any slice request with priority index greater or equal to γ is admitted. Although this guarantees any slice request falling in the category an admission, the consensus is that the algorithm may miss optimality as low priority resources may consume more expensive resources over time as opposed to short-time high priority slices.

1) COMPLEXITY ANALYSIS

Algorithm 3 shows a simplified heuristic method for slice admission control. The complexity of such algorithm highly depend on the number of single loop or nested loops used and the quantity of iterations needed for one episodic search. In the above algorithm there are two loops of N iterations each, therefore the total number of iterations required for one episode is $2N$. The computational cost required grows as the value of N grows. Delgado *et al.* [42] proposes a joint application admission control and network slicing in virtual sensor networks. The proposed method employs heuristic

method for the admission control, Algorithm 3 in the literature employs two nested loops of length $P_d \times I_g$ and $P_f \times I_g$. The total duration required to execute the entire algorithm is therefore given by $t = \tau(P_d \times I_g + P_f \times I_g + \sigma)$ where τ is the unit time required to execute each code and σ is a time variance. The computational cost is therefore given by $cost = \beta t$ where β is the unit cost per episode.

D. MACHINE LEARNING APPROACH

The use of machine learning for resource management in 5G has been gaining momentum, more so, in slice admission control. Intuitively, RL has been applied in slice admission control, as evident in the following literature [10], [16], [17], [23], [43]–[48]. RL generally reduces complexity which comes with problem models that require strict constraining as such constraints can be formulated as rewards or penalties. Moreover, in RL (Figure 9), a learning agent reads the the environment normally the admission request parameters and receives the reward value based on the action taken, the reward determines how good an action is. An action in slice admission control is a binary value indicating either a slice admission or rejection. The reward for admitting or rejecting a slice is evaluated by the agent to improve the next action. In multistage decision process (MDP) RL problems, the solution can be arrived in the following ways: value iteration, policy iteration, and Q-learning. In value iteration approach, the objective is to evaluate the state-action pair with optimum values, while in policy iteration, the aim is to resolve the best policy which leads to the objective while minimizing cost or maximizing the overall reward. In Q-learning approach a lookup Q-table consisting of state-action pairs and possible rewards is created. The RL algorithm is therefore applied to learn the state-action pairs. In slice admission control such a table may be too large hence becoming memory-intensive in what is known as the *curse of dimensionality*. Artificial neural networks used for value approximation have been adopted to solve such problems. This is referred to as deep RL. An actor-critic [61] model may be adopted where actor learns the best policy of state-action choice while the critic evaluates the actor's choice and generates the reward signal. The reward signal is used to improve the choice of the next action and the critic is trained to provide minimized criticism.

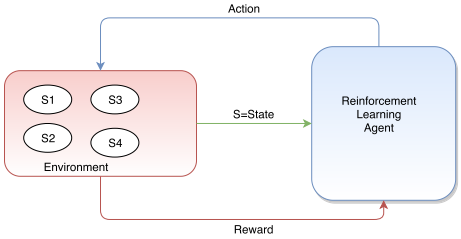

FIGURE 9. An illustration of reinforcement learning.

The following steps are applicable in RL algorithm:

i. Formulate the problem as an Multistage Decision Process (MDP) by defining the initial state-space, action-space, next-state give by $(s \cup S, a \cup A, s' \cup S) = (s, a, s')$ and the transition probability p. In slice admission, a set of parameters that define a slice request corresponds to a union of the states that form the state space(s), while the actions $a \in \{0, 1\}$ is binary variable and it defines a slice rejection or admission.
ii. The learning agent selects a state from the state-space and performs an action. In this case, an action is either slice request admission or rejection. For any action taken a reward is calculated.
iii. The next step is to move to the next state. The state is influenced by the current state and action such that $s' = f(s, a)$, this is known as state a transition. In RL state transitions are based on a MDP model. The learning agent maximizes the total reward or minimizes total cost based on the predefined admission objective. In slice admission control, reward accumulation is ordinarily the objective of the agent which may be translated to revenue accumulation [10]. The reward function $r = f(s, a, s')$ is a function of the current state, current action and next state.
iv. The process of maximum reward accumulation is used to create a policy π, the policy is improved in every step so that the value of the current policy is less than or equal to the value of the next policy that is $V(\pi) \leq V(\pi')$ until an optimum policy π^* is obtained.
v. The obtained policy is applied in any subsequent slice admission control.

A simplified algorithm based on dynamic programming (DP) is given in **Algorithm 4**. The algorithm according to [20] performs state value iteration.

Algorithm 4 An Illustration of DP Algorithm for Slice Admission

initialize $V(s) = 0 \ \ \forall s \in S$
repeat
  $\Delta \leftarrow 0$
  for each $s \in S$ **do**
    $v \leftarrow V(s)$
    perform action
    determine $p(s'r|s, a)$
    determine r
    $V(s) \leftarrow \max_a \sum_{s',r} p(s'r|s, a)[r + \gamma V(s)]$
    $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
  end for
until $\Delta \leq \theta$ (a very small value)

RL solution to slice admission control is quite appealing as mandatory system modeling is not necessary. However, not every resource management problem can be formulated as an MDP problem. Similarly creating and MDP may be too complex hindering the application of RL in resource management. The reader may refer to [20] detailed discussion on RL. The machine learning algorithms in 5G networks may

TABLE 2. A summary of optimization algorithms: Strengths and weakness.

Optimization Technique	References	Strength	Weaknesses
Successive Convex Approximation (SCA)	[8], [19], [53], [37]	Strives to reach global optimum. Works well in problems with stringent constraints	Mathematically intensive Requires massive relaxation Requires mandatory system modeling Not easily scalable
Alternating Direction Method of Multipliers (ADMM)	[1], [19], [40], [45], [53], [54]	Achieves optimality Best suited for long-term objective	Mathematically intensive Not suitable for discrete variables Requires mandatory system model Not easily scalable
Heuristic Approach	[2], [12], [55]–[58]	Not mathematically intensive. Based on search, Has high flexibility	May not guarantee optimality
Reinforcement Learning approach	[5], [16], [20], [43]–[46], [60], [62]–[66]	Strives to reach global optimum after training Highly scalable and flexible Not strictly dependent on the system model.	Training may be complex. Time-consuming and processing power intensive
Genetic Algorithm	[7]	Less mathematical intensive Suitable for nonbinary decisions	May not guarantee optimality

be applied to easily classify slice requests thereby simplifying slice admission control. Moreover, the complexity of designing conventional algorithms and dealing with big data analytics is largely solved by adopting machine learning [49].

1) COMPLEXITY ANALYSIS

Machine learning algorithms are known to be highly data intensive and the quantity of data determines the computational complexity and processing time. Specifically, in all RL algorithms, states and actions spaces must be defined. The size of state-action space is directly proportional to the computational cost and time involved. The complexity may increase further if states are continuous and must all be explored. If for each state an action has to be taken followed by a reward calculation then the time duration for training becomes even longer. Also, if a state-action-reward table has to be updated then memory requirement may increase exponentially. However, this problem may only be mandatory in Q-learning and not all RL solutions. For instance, an environment involving $N \times M$ state matrix, the action space may be in $\mathcal{O}(NM)$. Indeed, if the state becomes too large the action space my grow exponentially [50] and the computational cost involved in solving the MDP becomes too intensive. One solution to this problem is to perform value approximation with ANN or simply state-action aggregation.

E. GENETIC ALGORITHM

Genetic Algorithm (GA) is an optimization technique based on natural selection of evolution. Generally, in GA, a random selection is performed on the set known as the initial population which in slice admission control may denote slice admission requests. Each sample in the population is evaluated to determine its fitness. In its simple form, GA follows three main steps [7].

i. Reproduction stage: In this stage, every policy is copied into a new set and arranged according to the fitness. The best performing policy is arranged on top while the worst performing policy is arranged at the bottom. If revenue maximization is an objective, policies that generate more revenue are the fittest set in the population while those that are contrary becomes unfit policies and are discarded.
ii. Crossover Stage: In this stage, the reproduced policies are randomly paired with the aim of swapping the progression. The subsequent policy (child) inherits both parents ''genes''. The child is considered to be a superior policy than the parents.
iii. Mutation stage: This operation is performed over several iterations of bit inversions. The aim is to produce genetic diversity from the initial policy. The probability of a mutation should be set low otherwise a high probability may just create a simple random search. In slice admission control, the mutation process is the creation of a superior policy for best request selection.

Algorithm 5 An Illustration of GA Algorithm

```
start
generate initial population
evaluate fitness
repeat
    perform selection
    perform crossover
    perform mutation
    evaluate fitness
until convrgence
end
```

A simplified pseudo-code for GA is provided in **Algorithm 5**. Consider $x(n) = x(1), n(2), n(3) \ldots x(N)$ random population. The probability of selecting any candidate of binary string b of length l is given by $p(b)$, where each individual binary set b_i is a possible solution to a problem defined by an objective function [51]. The selection process depends the fitness of

TABLE 3. Summary of analysis of slice admission control from research papers.

Authors Category	Admission Objective	Slice Elasticity	Tenancy	Slicing Domain	Admission Strategy	Optimization Algorithm	Admission Domain	Queuing Strategy
Nejad et al [8]	Revenue optimization and QoS Control	Elastic and Inelastic	Multi-tenant	End to end	Optimal	SCA	Inter-slice	Unspecified
Bega et al [10]	Revenue Optimization and QoS Control	Elastic and Inelastic	Multi-tenant	RAN	Optimal	Deep Reinforcement Learning	Inter and Intra-slice	Unspecified
Han et al [13]	Inter-slice Congestion Control	Elastic and Inelastic	Multi-Tenant	End-to-End	Greedy	Q-learning	Inter-slice	Unspecified
Han et al [14]	Multi-queue control and Fairness	Unspecified	Multi-tenant	Unspecified	FIFO	Heuristic	Unspecified	M/M/c
Monti et al [17]	Revenue Optimization	Inelastic	Single Tenant	RAN	Priority	Reinforcement Learning	Intra-slice	Unspecified
Sun et al [16]	Revenue Optimization	Inelastic	Single-tenant	RAN	Optimal	Deep Reinforcement learning	Intra-slice	Unspecified
Tang et al [19]	Revenue Optimization	Inelastic	Multi-tenant	RAN	Optimal	SCA	Intra-slice	Unspecified
Jiang et al [12]	QoS Control	Unspecified	Multi-tenant	Unspecified	Priority	Heuristic	Intra-slice	Unspecified
Han et al [7]	Revenue Optimization	Unspecified	Multitenant	Unspecified	Optimal	Genetic Algorithm	Inter-slice	Unspecified
Vassilaras et al [18]	Resource Arbitration	Elastic and Inelastic	Multitenant	End-to-end	Unspecified	Heuristic	Inter-slice	Unspecified
Han et al [7]	Fairness Control	Unspecified	Multitenant	Unspecified	Randoms	Heuristic	Inter-slice	M/M/1
Natalino et al [5]	Revenue Optimization	Elastic and Inelastic	Multitenant	RAN	Priority	Reinforcement Learning	Inter-slice	Unspecified
Tong et al [43]	Revenue Optimization	Elastic and Inelastic	Unspecified	Unspecified	Optimal	Reinforcement Learning	Unspecified	Unspecified
Parsaeefard et al [67]	QoS Control	Elastic and Inelastic	Multi-tenant	RAN	Optimal	SCA	Intra-slice	Unspecified
Yi et al [35]	Revenue Optimization QoS Control	Elastic	Unspecified	RAN	Greedy	Heuristic	Unspecified	Unspecified
Perveen et al [21]	QoS Control	Elastic and Inelastic	Multi-tenant	End-to-end	Random	Heuristic	Inter-slice	Unspecified
Kammoun et al [31]	QoS Control	Unspecified	Single Tenant	End-to-end	Priority	Heuristic	Inter-slice	Unspecified
Caballero et al [32]	QoS Control	Elastic and Inelastic	Multi-tenant	Unspecified	Priority	Heuristic	Inter-slice	Unspecified
Soliman et al [33]	QoS Control	Unspecified	Multi-tenant	RAN	Priority	Heuristic	Inter-slice	Unspecified
Sun et al [11]	QoS Control	Unspecified	Multi-tenant	vRAN	Priority	DRL and Heuristic	Inter-slice	Unspecified

each candidate which can be obtained from Boltzmann distribution $z(n) = \frac{e^{x(n_i)}}{\sum_{n=1}^{N} e^{x(n)}}$. The crossover operation can be applied based on some probability $p_o \in [0, 1]$. The mutation operation of a candidate n may be obtained by flipping bit k occurring with a probability p_q. For instance if $b = 00100$ and $\bar{b} = 110111$ then by mutating $p_q \cdot p_q \cdot (1-p_q) \cdot p_q \cdot p_q \cdot p_q = p^j(1-p_q)^{l-j}$. The probability that b similar to $\bar{b}$ is therefore given by;

$$Pb \rightarrow \bar{b} = p_q^{H(b,|b)}(1-p_q)^{l-H(b,\bar{b})} \tag{6}$$

where $H(b, \bar{b})$ is the Hamming distance between b and $\bar{b}$ [51]

GA nonetheless, has several limitations such as i) repeated policy evaluation. ii) scalability problem iii) sub-optimal convergence, and iv) not performing well on binary decisions.

1) COMPUTATIONAL ANALYSIS

The computational complexity of solving slice admission control with GA is dictated by the candidate population and the number of binary sequence in the code book representing each candidate. Assume S slices where each slice request has N features to be considered. If each feature in a request can be mapped to a binary condition representing the possibility of meeting the demand then all the possible constructions can be

in $\mathcal{O}(2^{S\times N})$ [7], [51], [52]. This value can be further increased if each reproduction stage involves for instance, M copies. If the number of iterations involved during crossover and selection is β then the time duration before convergence is in $\mathcal{O}(2^{S\times N} \times \beta \times M)$.

In Table 2, we provide a summary of the optimization algorithms applied in slice admission control.

IV. SUMMARY OF ANALYSIS OF SLICE ADMISSION CONTROLS

Slice admission control has attracted attention from many researchers with variant objectives, ranging from revenue optimization, QoS control, congestion control and fairness of admission. Each of the objectives requires an admission strategy and optimization technique. An admitted slice can occupy an end-to-end domain, be owned by a single tenant or more, be elastic, inelastic or a combination of both. An end-to-end slicing domain requires more resources that may be owned by different InPs. For a slice request to be properly admitted and provisioned, a proper coordination among the InPs is required in order to guarantee seamless service delivery without violating SLA. In Table 3, we present a summary of slice admission control objectives from different authors and their associated features. It is clear that majority of the researchers have considered revenue optimization as the main objective of slice admission control and adopted both machine learning and variable heuristic techniques in optimization. Other optimization algorithms such as SCA and GA have been used as well.

While a slice may be considered as elastic or inelastic, the concept has not been elaborately explored by many researches. We have reviewed existing literature and classified the considered slice dimensions as either elastic, inelastic or, both.

Multi-tenancy in the 5G slicing has gained extensive attraction from many researchers, and its consideration has attained more publications compared to single tenancy. This is as a result of the diversity and heterogeneity in the 5G network.

The 5G infrastructure offers flexibility in resource allocation, which enables InPs to provision resources based on the requested domain i.e RAN, cloud and end-to-end. It is noted that, not much work has been done on end-to-end and specific cloud slicing admission control. Similarly many researchers have not distinctly specified the slicing domain considered. Therefore it reasonable to say that these areas still require more insight.

The trade-off in achieving optimal slice admission strategy and the practical application of an ideal optimization algorithm has made researchers to weigh between reducing complexity and achieving the best results. Many of the considerations so far have been based on priority admission. Subsequently, on the choice of an optimization algorithm, we have considered SCA and machine learning techniques as optimal because they strive to settle at the global optimum. Other techniques such as heuristic approach are known to be best-effort and mostly do not necessarily give optimal results. Finally, more researchers have explored machine learning techniques in the 5G network slice admission control compared to heuristic and GA approaches.

V. CONCLUSION

Network slicing is a key component of the 5G wireless network. An efficient admission control algorithm is needed to make slice admission decisions in the 5G network. In this paper, we have reviewed recent slice admission control algorithms. We have discussed admission objectives applied by different authors and also elaborated on diverse aspects of network slices such as slice elasticity, slice tenancy, slice admission domain and queuing. Different slicing strategies have been presented, and these strategies plays important roles in determining how a slice admission control objective is achieved. In order to enhance slice admission objectives, optimization algorithm are used. We have discussed five main optimization algorithms applied in the literature for optimizing slice admission control. We have also presented the merits and demerits of the algorithms followed by simplified examples of the algorithms for easy comprehension. A summary of these optimization schemes are given in compacted form in Table 3. Finally, it is observed that not much work has been done in slice admission control, particularly in the following areas, 1) dealing with inter and intra-slice isolation, ii) extensive end-to-end slice admission, iii) exploration of deep learning approach in slice admission control. Thus, there is need for further research in these areas.

REFERENCES

[1] G. Wang, G. Feng, S. Qin, R. Wen, and S. Sun, "Optimizing network slice dimensioning via resource pricing," *IEEE Access*, vol. 7, pp. 30331–30343, 2019.
[2] X. Li, J. Rao, H. Zhang, and A. Callard, "Network slicing with elastic SFC," in *Proc. IEEE 86th Veh. Technol. Conf. (VTC-Fall)*, Sep. 2017, pp. 1–5.
[3] Y. L. Lee, J. Loo, T. C. Chuah, and L.-C. Wang, "Dynamic network slicing for multitenant heterogeneous cloud radio access networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 4, pp. 2146–2161, Apr. 2018.
[4] T. Taleb, I. Afolabi, and M. Bagaa, "Orchestrating 5G network slices to support industrial Internet and to shape next-generation smart factories," *IEEE Netw.*, vol. 33, no. 4, pp. 146–154, Jul. 2019.
[5] C. Natalino, M. R. Raza, A. Rostami, P. Ohlen, L. Wosinska, and P. Monti, "Machine learning aided orchestration in multi-tenant networks," in *Proc. IEEE Photon. Soc. Summer Top. Meeting Ser. (SUM)*, Jul. 2018, pp. 125–126.
[6] M. Aicardi, R. Bruschi, F. Davoli, P. Lago, and J. F. Pajo, "Decentralized scalable dynamic load balancing among virtual network slice instantiations," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2018, pp. 1–7.
[7] B. Han, D. Feng, and H. D. Schotten, "A Markov model of slice admission control," *IEEE Netw. Lett.*, vol. 1, no. 1, pp. 2–5, Mar. 2018. [Online]. Available: http://arxiv.org/abs/1804.01861
[8] M. A. T. Nejad, S. Parsaeefard, M. A. Maddah-Ali, T. Mahmoodi, and B. H. Khalaj, "VSPACE: VNF simultaneous placement, admission control and embedding," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 542–557, Mar. 2018.
[9] M. G. Kibria, K. Nguyen, G. P. Villardi, O. Zhao, K. Ishizu, and F. Kojima, "Big data analytics, machine learning, and artificial intelligence in next-generation wireless networks," *IEEE Access*, vol. 6, pp. 32328–32338, 2018.
[10] D. Bega, M. Gramaglia, A. Banchs, V. Sciancalepore, and X. Costa-Perez, "A machine learning approach to 5G infrastructure market optimization," *IEEE Trans. Mobile Comput.*, to be published.

[11] G. Sun, K. Xiong, G. O. Boateng, D. Ayepah-Mensah, G. Liu, and W. Jiang, "Autonomous resource provisioning and resource customization for mixed traffics in virtualized radio access network," *IEEE Syst. J.*, vol. 13, no. 3, pp. 2454–2465, Sep. 2019.

[12] M. Jiang, M. Condoluci, and T. Mahmoodi, "Network slicing management & prioritization in 5G mobile systems," in *Proc. Eur. Wireless 22th Eur. Wireless Conf.*, pp. 197–202, 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7499297/

[13] B. Han, A. Dedomenico, G. Dandachi, A. Drosou, D. Tzovaras, R. Querio, F. Moggio, O. Bulakci, and H. D. Schotten, "Admission and congestion control for 5G network slicing," in *Proc. IEEE Conf. Standards Commun. Netw. (CSCN)*, Oct. 2018.

[14] B. Han, V. Sciancalepore, D. Feng, X. Costa-Perez, and H. D. Schotten, "A utility-driven multi-queue admission control solution for network slicing," May 2019, *arXiv:1901.06399*. [Online]. Available: http://arxiv.org/abs/1901.06399

[15] D. Bega, M. Gramaglia, A. Banchs, V. Sciancalepore, K. Samdanis, and X. Costa-Perez, "Optimising 5G infrastructure markets: The business of network slicing," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, May 2017.

[16] R. Li, Z. Zhao, Q. Sun, C.-L. I, C. Yang, X. Chen, M. Zhao, and H. Zhang, "Deep reinforcement learning for resource management in network slicing," *IEEE Access*, vol. 6, pp. 74429–74441, 2018.

[17] P. Monti, C. Natalino, M. R. Raza, P. Ohlen, and L. Wosinska, "A slice admission policy based on reinforcement learning for a 5G flexible RAN," in *Proc. Eur. Conf. Opt. Commun. (ECOC)*, no. 1, 2018, pp. 1–3.

[18] S. Vassilaras, L. Gkatzikis, N. Liakopoulos, I. N. Stiakogiannakis, M. Qi, L. Shi, L. Liu, M. Debbah, and G. S. Paschos, "The algorithmic aspects of network slicing," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 112–119, Aug. 2017.

[19] J. Tang, B. Shim, and T. Q. S. Quek, "Service multiplexing and revenue maximization in sliced C-RAN incorporated with URLLC and multicast eMBB," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 4, pp. 881–895, Apr. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8638932/

[20] R. S. Sutton and A. G. Barto, *Reinforcement Learning Complete Draft*. Cambridge, MA, USA: MIT Press, 2017.

[21] A. Perveen, M. Patwary, and A. Aneiba, "Dynamically reconfigurable slice allocation and admission control within 5G wireless networks," in *Proc. IEEE 89th Veh. Technol. Conf. (VTC-Spring)*, Apr. 2019, pp. 1–7.

[22] T. Guo and A. Suarez, "Enabling 5G RAN slicing with EDF slice scheduling," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 2865–2877, Mar. 2019.

[23] S. E. Elayoubi, S. B. Jemaa, Z. Altman, and A. Galindo-Serrano, "5G RAN slicing for verticals: Enablers and challenges," *IEEE Commun. Mag.*, vol. 57, no. 1, pp. 28–34, Jan. 2019.

[24] C. Song, M. Zhang, X. Huang, Y. Zhan, D. Wang, M. Liu, and Y. Rong, "Machine learning enabling traffic-aware dynamic slicing for 5G optical transport networks," in *Proc. Conf. Lasers Electro-Optics*, 2018, Paper JTu2A.44.

[25] M. A. C. Almeida and F. R. B. Cruz, "A note on Bayesian estimation of traffic intensity in single-server Markovian queues," *Commun. Statist.-Simul. Comput.*, vol. 47, no. 9, pp. 2577–2586, Oct. 2018.

[26] S. Dharmaraja and R. Kumar, "Transient solution of a Markovian queuing model with heterogeneous servers and catastrophes," *Opsearch*, vol. 52, no. 4, pp. 810–826, Dec. 2015.

[27] X. Liu, S. Li, and W. Tong, "A queuing model considering resources sharing for cloud service performance," *J. Supercomput.*, vol. 71, no. 11, pp. 4042–4055, Nov. 2015.

[28] J.-C. Ke, F.-M. Chang, and T.-H. Liu, "M/M/c balking retrial queue with vacation," *Qual. Technol. Quant. Manage.*, vol. 16, no. 1, pp. 54–66, Jan. 2019, doi: 10.1080/16843703.2017.1365280.

[29] Z. Liu and S. Yu, "The M/M/c queueing system in a random environment," *J. Math. Anal. Appl.*, vol. 436, no. 1, pp. 556–567, Apr. 2016.

[30] L. Zhang and J. Li, "The M/M/c queue with mass exodus and mass arrivals when empty," *J. Appl. Probab.*, vol. 52, no. 4, pp. 990–1002, Dec. 2015.

[31] A. Kammoun, N. Tabbane, G. Diaz, and N. Achir, "Admission control algorithm for network slicing management in SDN-NFV environment," in *Proc. 6th Int. Conf. Multimedia Comput. Syst. (ICMCS)*, May 2018, pp. 1–6.

[32] P. Caballero, A. Banchs, G. De Veciana, X. Costa-Perez, and A. Azcorra, "Network slicing for guaranteed rate services: Admission control and resource allocation games," *IEEE Trans. Wireless Commun.*, vol. 17, no. 10, pp. 6419–6432, Oct. 2018.

[33] H. M. Soliman and A. Leon-Garcia, "QoS-aware frequency-space network slicing and admission control for virtual wireless networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2016, pp. 1–6.

[34] R. Challa, V. V. Zalyubovskiy, S. M. Raza, H. Choo, and A. De, "Network slice admission model: Tradeoff between monetization and rejections," *IEEE Syst. J.*, to be published.

[35] B. Yi, X. Wang, and M. Huang, "Optimised approach for VNF embedding in NFV," *IET Commun.*, vol. 12, no. 20, pp. 2630–2638, Dec. 2018.

[36] D. Bega, M. Gramaglia, M. Fiore, A. Banchs, and X. Costa-Perez, "DeepCog: Cognitive network management in sliced 5G networks with deep learning," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr./May 2019.

[37] M. Razaviyayn, "Successive convex approximation: Analysis and applications," Ph.D. dissertation, Faculty Graduate School, Univ. Minnesota, Minneapolis, MN, USA, 2014, pp. 8–48.

[38] S. S. Rao, *Engineering Optimization*. Hoboken, NJ, USA: Wiley, 2009.

[39] Z. Allybokus, K. Avrachenkov, J. Leguay, and L. Maggi, "Lower bounds for the fair resource allocation problem," Feb. 2018, *arXiv:1802.02932*. [Online]. Available: http://arxiv.org/abs/1802.02932

[40] P. T. A. Quang, A. Bradai, K. D. Singh, G. Picard, and R. Riggio, "Single and multi-domain adaptive allocation algorithms for VNF forwarding graph embedding," *IEEE Trans. Netw. Serv. Manage.*, vol. 16, no. 1, pp. 98–112, Mar. 2019.

[41] W. Tian and X. Yuan, "An alternating direction method of multipliers with a worst-case $O(1/n^2)$ convergence rate," *Math. Comput.*, vol. 88, no. 318, pp. 1685–1713, 2019.

[42] C. Delgado, M. Canales, J. Ortin, J. R. Gallego, A. Redondi, S. Bousnina, and M. Cesana, "Joint application admission control and network slicing in virtual sensor networks," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 28–43, Feb. 2018.

[43] H. Tong and T. Brown, "Adaptive call admission control under quality of service constraints: A reinforcement learning solution," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 2, pp. 209–221, Feb. 2000.

[44] H. Mao, M. Alizadeh, I. Menache, and S. Kandula, "Resource management with deep reinforcement learning," in *Proc. 15th ACM Workshop Hot Topics Netw. (HotNets)*, 2016, pp. 50–56.

[45] G. Sun, Z. T. Gebrekidan, G. O. Boateng, D. Ayepah-Mensah, and W. Jiang, "Dynamic reservation and deep reinforcement learning based autonomous resource slicing for virtualized radio access networks," *IEEE Access*, vol. 7, pp. 45758–45772, 2019.

[46] T. Maksymyuk, J. Gazda, O. Yaremko, and D. Nevinskiy, "Deep learning based massive MIMO beamforming for 5G mobile network," in *Proc. IEEE 4th Int. Symp. Wireless Syst. Int. Conf. Intell. Data Acquisition Adv. Comput. Syst. (IDAACS-SWS)*, Sep. 2018, pp. 241–244. [Online]. Available: https://ieeexplore.ieee.org/document/8525802/

[47] L.-V. Le, B.-S.-P. Lin, L.-P. Tung, and D. Sinh, "SDN/NFV, machine learning, and big data driven network slicing for 5G," in *Proc. IEEE 5G World Forum (GWF)*, Jul. 2018, pp. 20–25.

[48] R. Mijumbi, J.-L. Gorricho, J. Serrat, M. Claeys, J. Famaey, and F. De Turck, "Neural network-based autonomous allocation of resources in virtual networks," in *Proc. Eur. Conf. Netw. Commun. (EuCNC)*, Jun. 2014.

[49] M. Eugenio, M. Cayamcela, and W. Lim, "Artificial intelligence in 5G technology: A survey," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, 2018, pp. 860–865.

[50] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Process. Mag.*, vol. 34, no. 6, pp. 26–38, Nov. 2017.

[51] G. Rudolph, "Convergence analysis of canonical genetic algorithms," *IEEE Trans. Neural Netw.*, vol. 5, no. 1, pp. 96–101, Jan. 1994.

[52] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, 1st ed. Boston, MA, USA: Addison-Wesley, 1989.

[53] A. Yadav, O. A. Dobre, and N. Ansari, "Distributed energy and resource management for full-duplex dense small cells for 5G," in *Proc. 13th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jun. 2017, pp. 133–139.

[54] R. Sawant and S. Nema, "Outage probability analysis in hybrid cooperative cognitive radio networks," in *Proc. Int. Conf. Big Data, IoT Data Sci. (BID)*, Dec. 2017, pp. 77–81.

[55] G. Durisi, T. Koch, and P. Popovski, "Toward massive, ultrareliable, and low-latency wireless communication with short packets," *Proc. IEEE*, vol. 104, no. 9, pp. 1711–1726, Sep. 2016.

[56] V. S. Reddy, A. Baumgartner, and T. Bauschert, "Robust embedding of VNF/service chains with delay bounds," in *Proc. IEEE Conf. Netw. Function Virtualization Softw. Defined Netw. (NFV-SDN)*, Nov. 2016, pp. 93–99.
[57] T. Truong-Huu, P. M. Mohan, and M. Gurusamy, "Service chain embedding for diversified 5G slices with virtual network function sharing," *IEEE Commun. Lett.*, vol. 23, no. 5, pp. 826–829, May 2019. [Online]. Available: https://www.researchgate.net/publication/331165922
[58] N. Zhang, Y.-F. Liu, H. Farmanbar, T.-H. Chang, M. Hong, and Z.-Q. Luo, "Network slicing for service-oriented networks under resource constraints," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2512–2521, Nov. 2017.
[59] V. P. Kafle, Y. Fukushima, P. Martinez-Julia, and T. Miyazawa, "Consideration on automation of 5G network slicing with machine learning," in *Proc. ITU Kaleidoscope, Mach. Learn. 5G Future (ITU K)*, Nov. 2018, pp. 1–8.
[60] C. Song *et al.*, "Machine learning enabling traffic-aware dynamic slicing for 5G optical transport networks," in *Proc. Conf. Lasers Electro-Opt. (CLEO)*, San Jose, CA, USA, 2018, pp. 1–2. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8427129&isnumber=8426208
[61] I. Grondman, L. Busoniu, G. A. D. Lopes, and R. Babuska, "A survey of actor-critic reinforcement learning: Standard and natural policy gradients," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 6, pp. 1291–1307, Nov. 2012.
[62] Y. Yang, Y. Li, K. Li, S. Zhao, R. Chen, J. Wang, and S. Ci, "DECCO: Deep-learning enabled coverage and capacity optimization for massive MIMO systems," *IEEE Access*, vol. 6, pp. 23361–23371, 2018.
[63] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2224–2287, Mar. 2019. [Online]. Available: http://arxiv.org/abs/1803.04311
[64] L. L. Vy, L.-P. Tung, and B.-S.-P. Lin, "Big data and machine learning driven handover management and forecasting," in *Proc. IEEE Conf. Standards Commun. Netw. (CSCN)*, Sep. 2017, pp. 214–219.
[65] S. Parsaeefard, V. Jumba, M. Derakhshani, and T. Le-Ngoc, "Joint resource provisioning and admission control in wireless virtualized networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Mar. 2015, pp. 2020–2025.
[66] P. Caballero, A. Banchs, G. De Veciana, and X. Costa-Perez, "eEnabling customization in multi-tenant networks," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, May 2017, pp. 1–9.

MOURICE O. OJIJO received the B.Sc. degree from the University of Mysore, Mysore, India, in 2005, the master's degree in electronics from the University of Mysore, in 2007, and the Diploma degree in telecommunication engineering from The Mombasa Polytechnic (currently Technical University of Mombasa), in 2001. He is currently pursuing the Ph.D. degree with the University of Cape Town, South Africa. He is also a Lecturer with Kabarak University. His research interests are 5G networks and machine learning.

OLABISI E. FALOWO (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from the University of Cape Town, in 2008. He is currently an Associate Professor with the University of Cape Town. He has published over 80 technical articles in peer-reviewed conference proceedings and journals, including the IEEE COMMUNICATION LETTERS, *IET Communications*, the *Elsevier Computer Communications*, the *Elsevier Computer Networks*, the *EURASIP Journal on Wireless Communications and Networking*, the *Wiley Wireless Communications and Mobile Computing*, and *Telecommunication Systems*. His primary research interest is in radio resource management in heterogeneous wireless networks.