Availability-Aware VNF Placement for uRLLC Applications in MEC-Enabled 5G Networks

Samaresh Bera, *Senior Member, IEEE*
Department of Computer Science and Engineering
Indian Institute of Technology Jammu, 181221, India
Email: s.bera.1989@ieee.org

***Abstract*—In this paper, we study the VNF placement problem in MEC-enabled 5G networks to meet the stringent reliability and latency requirements of uRLLC applications. We pose it as a constrained optimization problem, which is NP-hard, to maximize the total reward obtained by a network service provider by serving uRLLC service requests. We propose an approximated randomized rounding approach to solve the NP-hard optimization problem in polynomial time. We prove that the proposed randomized approach achieves performance guarantees while violating the resource constraints in a bounded way. Furthermore, we present a greedy-heuristic approach to tackle the violations in the resource constraints. Simulation results show the proposed approach yields close-to-optimal performance. Specifically, the total reward is within 5% and 10% of the optimal solution using the proposed randomized rounding and greedy approaches, respectively.**

***Index Terms*—Mobile edge computing, Resource allocation, Optimization, Ultra-reliable and low-latency communications, 5G**

I. Introduction

The emergence of the mobile edge computing (MEC) framework facilitates network service providers to meet the stringent latency requirements of many applications, such as autonomous driving and remote surgery, by placing computing functionalities near the users. Furthermore, the MEC-based networking has proliferated with the introduction of 5G (and beyond) networks. With MEC, a service can be hosted either at the edge cloud or at the central cloud, depending on its requirements. Recent studies show that MEC is helpful in meeting the stringent latency requirements of the above-mentioned applications while leveraging the benefits of software-defined networking (SDN) and network function virtualization (NFV) [1]–[3].

The applications supported by 5G (and beyond) are broadly categorized as enhanced mobile broadband (eMBB), ultra-reliable and low-latency communications (uRLLC), and massive machine-type communications (mMTC) [4]. While there has been significant progress in addressing the high-bandwidth and low-latency requirements of 5G applications [5], ensuring high reliability is still challenging for uRLLC applications. This is due to the coupling of high-reliability and low-latency requirements of uRLLC applications that makes the network modeling very challenging [6].

In this paper, we study the VNF placement problem in an MEC-enabled 5G network while considering the stringent availability[1] requirements of uRLLC applications. We pose this as a constrained optimization problem, which captures multi-dimensional MEC networking resources, such as CPU and RAM, and availability requirements of uRLLC service requests, to maximize the total reward obtained by the service provider. Furthermore, we propose an approximation algorithm to solve the NP-hard optimization problem with performance bounds. The key contributions in this paper are as follows:

- We mathematically model the VNF placement as a constrained optimization problem to maximize the total reward to a service provider by serving incoming requests. The problem considers multi-dimensional networking resources and availability requirements to meet the stringent reliability of uRLLC applications in 5G.
- We propose an approximation algorithm based on randomized rounding techniques [7] to solve the NP-hard optimization problem in polynomial time. Furthermore, we show that it provably achieves performance guarantees while violating the resource constraints in a bounded way.
- We propose a greedy-heuristic approach based on the randomized rounding solution to tackle the violations in resource constraints, if any. The simulation results show that the proposed approximation algorithm performs close-to-optimal while violating the constraints in a bounded way. Furthermore, the proposed greedy approach also yields competitive performance to the optimal solution.

The rest of the paper is organized as follows. Section II highlights the state-of-the-art solution approaches for the VNF placement problem in 5G networks. Section III presents the detailed network model and the optimization problem. Section IV presents the proposed approximation algorithm while analyzing the bounds on the performance, and the proposed greedy approach. The efficacy of the proposed scheme is studied in Section V. Finally, Section VI concludes the paper with future research directions.

II. Related Work

This section discusses the existing works on VNF placement and resource allocation in 5G networks while highlighting the key differences between them and the proposed scheme.

[1]In this work, the terms 'reliability' and 'availability' are used interchangeably to denote the same thing.

Yala et al. [1] studied availability and latency-aware VNF placement problem at MECs and the central cloud. The authors modeled it as a trade-off between the service latency and the availability of VNFs. For latency-critical services, the placement of VNFs is preferred at the MECs over the central cloud. In contrast, the VNFs related to availability-critical services are placed in the central cloud. However, as discussed in [8], the uRLLC use cases in 5G have stringent latency and availability requirements, which must be ensured. Furthermore, the authors propose a genetic algorithm-based approach for which the solution may not converge even after multiple iterations.

Poularakis et al. [3] studied service placement and request routing problem in an MEC-enabled network, where base-stations are enabled with storage and compute resources and act as edge clouds. The authors framed the optimization problem as the minimization of request routing to the central cloud while adhering to the associated constraints. Similarly, Yang et al. [9] framed the VNF placement and routing problem as the minimization of service delay while considering the networking resources and request-specific requirements. Both [3] and [9] proposed approximation algorithms to solve the NP-hard optimization problems in polynomial time. Behravesh et al. [10] studied the joint user association and VNF placement problem in the network consisting of MECs and the central cloud. The authors formulated it as a mixed integer linear program (MILP) and solved it using an optimization problem solver. However, it is unsuitable for large-scale deployment due to the NP-hardness of the optimization problem.

Synthesis: While [3], [9] are the closest ones to our work, they did not consider the availability of VNFs, which makes the problem challenging due to the limited networking resources at the MECs. Furthermore, while the other existing schemes tried to address the issues and challenges in supporting the stringent QoS requirements, very few are scalable and yield competitive performance to the optimal solution.

III. System Model

Figure 1 presents an overview of the MEC-assisted VNF placement in 5G networks. The VNFs associated with a request are placed at the MECs and/or the central cloud.

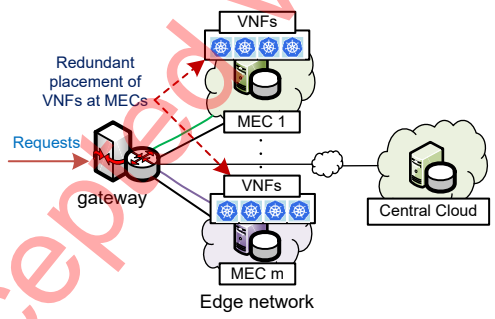


Fig. 1: VNFs placement in MEC-enabled 5G networks

In this work, we consider the following:

- VNFs in 5G network are categorized as control-plane functions (CPFs) and user-plane functions (UPFs). The CPFs are pre-placed either at the MEC, central cloud, or both.
- The VNFs are placed in the form of a virtual machine (VM) or a container. Furthermore, we focus on the UPF placement at the MECs in the network. The requests that cannot be served by MEC are either dropped or served by central cloud while considering their requirements.
- Considering issues with latency in serialized HTTP/JSON connection [11], all UPFs associated with a service are placed at a single physical machine (PM).
- A request is replicated and assigned to multiple MECs for providing services based on the availability requirement.
- The first error-free service-response returns to the gateway is considered. Hence, end-to-end latency and availability are improved.

A. Network Model

Let there be a set of MECs denoted as $\mathcal{M} = \{1, 2, \cdots, M\}$. Each MEC $m \in \mathcal{M}$ has a certain amount of CPU and RAM resources to host VNFs, denoted by C_m and D_m, respectively. The service provider receives service requests denoted by a set $\mathcal{R} = \{1, 2, \cdots, R\}$. Each request $r \in \mathcal{R}$ has a certain amount of CPU and RAM resource requirements according to the UPFs associated with the service, denoted by c_r and d_r, respectively. Furthermore, each request is associated with a threshold on service failure ϵ_r to meet its availability requirement and a reward ζ_r obtained by the service provider if the request is served.

B. Latency Model

As shown in Figure 1, a request can be served either at the MECs or at the central cloud. We also consider that the service execution rate at a VNF is independent of its placement (whether at MEC or the central cloud) when the same amount of resources are allocated to the VNF. Therefore, in this work, we consider that the placement of VNFs at the MECs helps in reducing the end-to-end latency similar to the existing works [1], [9], [12].

Considering the low-latency requirements of uRLLC applications, we aim to serve the requests by placing VNFs at the MECs as much as possible to reduce the service delay.

C. Availability Model

A service may not be available due to the following reasons:

- At least one VNF associated with the service fails due to software failure of the VNF itself [13]. Let the probability of a VNF failure be ϵ_{v}.
- A VNF fails due to the failure of the host physical machine [13]. Let the probability of a physical machine failure be ϵ_{p}.

As all the VNFs associated with a request are placed at the same MEC, the total probability of failure of a service is calculated as:

$$\begin{aligned}\epsilon_{\text{m}} &= \Pr\left[\text{VNF fails}|\text{PM is OK}\right] + \Pr\left[\text{VNF fails}|\text{PM fails}\right] \\ &= \left(\epsilon_{\text{v}} + \epsilon_{\text{p}}\right).\end{aligned}$$

Now, for a given request $r \in \mathcal{R}$ with a threshold failure requirement ϵ_r, the following condition needs to be satisfied to meet the availability requirement:

$$[\epsilon_{\mathrm{m}}]^{\Psi_r} \leq \epsilon_r, \tag{1}$$

where Ψ_r denotes the number of redundant placement of VNFs at MECs for request r. To get the number of redundant placement Ψ_r, we rewrite (1) as follows:

$$\Psi_r = \lceil \log_{\epsilon_{\mathrm{m}}} (\epsilon_r) \rceil. \tag{2}$$

D. Optimization Problem

Given the MECs with resources and the requests with requirements, the objective of the service provider is to maximize the total reward by serving requests at the MECs. Mathematically,

$$\text{Maximize } P_{\text{IP}} = \sum_{r \in \mathcal{R}} \zeta_r y_r, \tag{3}$$

subject to

$$x_{r,m} \text{ and } y_r \in \{0,1\}, \forall r \in \mathcal{R}, \forall m \in \mathcal{M}, \tag{4a}$$

$$\sum_{m \in \mathcal{M}} x_{r,m} \geq \Psi_r y_r, \forall r \in \mathcal{R}, \tag{4b}$$

$$y_r \leq 1, \forall r \in \mathcal{R}, \tag{4c}$$

$$\sum_{r \in \mathcal{R}} c_r x_{r,m} \leq C_m, \forall m \in \mathcal{M}, \tag{4d}$$

$$\sum_{r \in \mathcal{R}} d_r x_{r,m} \leq D_m, \forall m \in \mathcal{M}. \tag{4e}$$

Equation (3) denotes the objective, which is to maximize the total reward by serving requests at the MECs, where $y_r = 1$ if the request r is served by the MEC, else 0, as denoted in (4a). The service requests that cannot be served at the MEC are either dropped to forwarded to the central cloud. Equation (4a) also represents binary decision variables on VNF placement at MECs, where $x_{r,m} = 1$ if VNFs associated with request r are placed at MEC m, else 0. Equation (4b) ensures that the availability requirement is satisfied. Furthermore, it also ensures that if a request is served, VNFs associated with it must be placed in the network. A request can be admitted at most once, which is ensured in (4c). Finally, (4d) and (4e) present that the CPU and RAM utilizations at the MECs are within the total CPU and RAM capacities, respectively. The optimization problem is a variation of the multi-constraint knapsack problem, which is NP-hard in general [14]. In the subsequent section, we propose a polynomial time approximation algorithm to solve this problem.

IV. Solution Approach: Randomized Rounding

A. Approximated Solution

We first relax the binary variables to continuous ones to solve the problem (3) in polynomial time. Mathematically, the optimization problem is represented as:

$$\text{Maximize } P_{\text{LR}} = \sum_{r \in \mathcal{R}} \zeta_r y_r, \tag{5}$$

subject to

$$\text{(4b), (4c), (4d), and (4e)},$$
$$x_{r,m} \text{ and } y_r \in [0,1]. \tag{6a}$$

The problem in (5) can be solved using standard LP-solvers. We use the `IBM CPLEX` [15] to get the solution. Let the solution be $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{y}}$. Now, we round the solution of the relaxed problem using the randomized rounding approach [7], as presented in Algorithm 1.

Algorithm 1 Randomized rounding algorithm

Inputs: Set of MECs: $\mathcal{M}$, each with C_m and D_m, $\forall m \in \mathcal{M}$;
Set of requests: $\mathcal{R}$, each with c_r, d_r, ζ_r, and ϵ_r, $\forall r \in \mathcal{R}$;
Output: Binary solution: $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$

1: Calculate Ψ_r using (2)
2: Solve the optimization problem in (5) to obtain $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$
3: **for** $r \in \mathcal{R}$ **do**
4: **for** $m \in \mathcal{M}$ **do**
5: Set $\hat{x}_{r,m} = 1$ with probability $\tilde{x}_{r,m}$ and $\hat{x}_{r,m} = 0$ with probability $(1 - \tilde{x}_{r,m})$
6: **if** $\sum_{m \in \mathcal{M}} \hat{x}_{r,m} \geq \Psi_r$ **then**
7: Set $\hat{y}_r = 1$ with probability $\tilde{y}_r$ and $\hat{y}_r = 0$ with probability $(1 - \tilde{y}_r)$
8: **return** $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$

From the construction of Algorithm 1, a request is either served at the MECs or dropped (or can be served by the central cloud). Therefore, the constraint (4c) is always satisfied. Furthermore, the `Step 6` (in Algorithm 1) satisfies the redundancy constraint (4b) to meet the availability requirements. And, the values of $x_{r,m}$ and y_r are always either 0 or 1, which satisfies (4a). Now, we check whether the remaining constraints (4d) and (4e) are satisfied.

Lemma 1. *The solution returned by Algorithm 1 satisfies the CPU and RAM capacity constraints in expectation.*

Proof. **CPU capacity constraint:** The expected CPU utilization of an MEC $m \in \mathcal{M}$ is given by

$$\mathbb{E}\left[\sum_{r \in \mathcal{R}} \hat{x}_{r,m} c_r\right] = \sum_{r \in \mathcal{R}} \Pr\left[\hat{x}_{r,m} = 1\right] c_r = \sum_{r \in \mathcal{R}} \tilde{x}_{r,m} c_r \leq C_m, \tag{7}$$

where the last equality holds as $\{\hat{x}_{r,m}\} = 1$ with success probabilities $\{\tilde{x}_{r,m}\}$ (refer to `Step 5` in Algorithm 1). Furthermore, the inequality holds due to the constraint (4d).

RAM capacity constraint: The expected RAM utilization of an MEC $m \in \mathcal{M}$ is given by

$$\mathbb{E}\left[\sum_{r \in \mathcal{R}} \hat{x}_{r,m} d_r\right] = \sum_{r \in \mathcal{R}} \Pr\left[\hat{x}_{r,m} = 1\right] d_r = \sum_{r \in \mathcal{R}} \tilde{x}_{r,m} d_r \leq D_m. \tag{8}$$

Similar to (7), the last equality and the inequality hold due to the success probability and constraint (4e), respectively. □

Lemma 2. *The total reward returned by Algorithm 1 is in expectation equal to that of the optimal fractional solution.*

Proof. The expected reward obtained by the service provider by serving requests at the MECs is given by

$$\mathbb{E}\left[\sum_{r\in\mathcal{R}}\zeta_r\hat{y}_r\right]=\sum_{r\in\mathcal{R}}\Pr\left[\hat{y}_r=1\right]\zeta_r=\sum_{r\in\mathcal{R}}\tilde{y}_r\zeta_r, \tag{9}$$

where the last equality holds as $\{\hat{y}_r = 1\}$ with success probabilities $\{\tilde{y}_r\}$. □

Though the constraints (4d) and (4e) are satisfied in expectation, they can be violated in practice. Therefore, we give the theoretical bounds on the violation of the constraints below.

Lemma 3. *The CPU load on an MEC $m \in \mathcal{M}$ returned by the Algorithm 1 will not exceed its capacity by more than a factor of $(1+\delta_c) = \frac{3\ln(R)}{\mu_c} + 4$ with high probability.*

Proof. We are interested in finding the probability that the constraint is violated. Mathematically,

$$\Pr\left[\sum_{r\in\mathcal{R}}\hat{x}_{r,m}c_r \geq (1+\delta_c)\sum_{r\in\mathcal{R}}\tilde{x}_{r,m}c_r\right]. \tag{10}$$

We will apply the Chernoff Bound [7] to get a theoretical bound on the above probability. Before that, we normalize the expression in (10), which is given by:

$$\Pr\left[\sum_{r\in\mathcal{R}}\frac{\hat{x}_{r,m}c_r}{\alpha_c} \geq (1+\delta_c)\sum_{r\in\mathcal{R}}\frac{\tilde{x}_{r,m}c_r}{\alpha_c}\right], \tag{11}$$

$$\text{where } \alpha_c = \max\{c_r, \forall r\in\mathcal{R}\},$$

$$\Rightarrow \Pr\left[\sum_{r\in\mathcal{R}} z^c_{r,m} \geq (1+\delta_c)\,\mu_c\right], \tag{12}$$

$$\text{where } \mu_c = \sum_{r\in\mathcal{R}}\frac{\tilde{x}_{r,m}c_r}{\alpha_c}.$$

Now, for a given MEC $m \in \mathcal{M}$, $z^c_{r,m} \in [0,1]$, $\forall r \in \mathcal{R}$, are independent random variables (RVs) with expected total value $\mathbb{E}\left[\sum_{r\in\mathcal{R}} z^c_{r,m}\right] = \mu_c$. By following the Chernoff bound (upper tail) [7], we get

$$\Pr\left[\sum_{r\in\mathcal{R}} z^c_{r,m} \geq (1+\delta_c)\,\mu_c\right] \leq \exp^{\frac{-\delta_c^2\mu_c}{2+\delta_c}}, \text{ which implies to}$$

$$\Pr\left[\sum_{r\in\mathcal{R}}\frac{\hat{x}_{r,m}c_r}{\alpha_c} \geq (1+\delta_c)\sum_{r\in\mathcal{R}}\frac{\tilde{x}_{r,m}c_r}{\alpha_c}\right] \leq \exp^{\frac{-\delta_c^2\mu_c}{2+\delta_c}},$$

which is equivalent to

$$\Pr\left[\sum_{r\in\mathcal{R}}\hat{x}_{r,m}c_r \geq (1+\delta_c)\sum_{r\in\mathcal{R}}\tilde{x}_{r,m}c_r\right] \leq \exp^{\frac{-\delta_c^2\mu_c}{2+\delta_c}}. \tag{13}$$

Next, we need to find a value of δ_c for which the probability value quickly goes to zero as the number of requests increases. To do that, we set

$$\exp^{\frac{-\delta_c^2\mu_c}{2+\delta_c}} \leq \frac{1}{R^3}, \tag{14}$$

which means δ_c must satisfy the below inequality:

$$\delta_c \geq \frac{3}{2}\frac{\ln(R)}{\mu_c} + \sqrt{\frac{9}{4}\left(\frac{\ln(R)}{\mu_c}\right)^2 + \frac{6\ln(R)}{\mu_c}}. \tag{15}$$

To hold the above condition true, δ_c must be as follows:

$$\delta_c = \frac{3\ln(R)}{\mu_c} + 3. \tag{16}$$

Finally, we upper bound the probability that any of the MECs CPU capacity is violated using Union bound [16] as:

$$\begin{aligned}
&\Pr\left[\bigcup_{m\in\mathcal{M}}\sum_{r\in\mathcal{R}} z^c_{r,m} \geq (1+\delta_c)\mu_c\right] \\
&\leq \sum_{m\in\mathcal{M}}\Pr\left[\sum_{r\in\mathcal{R}} z^c_{r,m} \geq (1+\delta_c)\mu_c\right] \\
&\leq \sum_{m\in\mathcal{M}}\Pr\left[\sum_{r\in\mathcal{R}}\frac{\hat{x}_{r,m}c_r}{\alpha_c} \geq (1+\delta_c)\sum_{r\in\mathcal{R}}\frac{\tilde{x}_{r,m}c_r}{\alpha_c}\right] \\
&\leq M\frac{1}{R^3} \leq \frac{1}{R^2}, \text{ where } R=|\mathcal{R}| \text{ and } M=|\mathcal{M}|.
\end{aligned} \tag{17}$$

The last inequality in (17) holds as the number of MECs M is much lesser than the number of requests R in practice ($M < R$). Therefore, the CPU capacity of any MEC $m \in \mathcal{M}$ will not exceed by more than a factor of $(1+\delta_c) = \frac{3\ln(R)}{\mu_c} + 4$ with high probability. □

Lemma 4. *The RAM load on an MEC $m \in \mathcal{M}$ returned by the Algorithm 1 will not exceed its capacity by more than a factor of $(1+\delta_d) = \frac{3\ln(R)}{\mu_d} + 4$ with high probability, where $\mu_d = \sum_{r\in\mathcal{R}}\frac{\tilde{x}_{r,m}d_r}{\alpha_d}$, and $\alpha_d = \max\{d_r, \forall r\in\mathcal{R}\}$.*

Proof. The proof for Lemma 4 follows the proof of the upper bound of the CPU load in Lemma 3. □

Now, we study the theoretical bound on the objective value.

Lemma 5. *The objective value returned by Algorithm 1 is atmost $(1-\sqrt{\frac{4\ln(R)}{\mu_{opt}}})$ times worse than the optimal solution of the relaxed problem, where $\mu_{opt} = \sum_{r\in\mathcal{R}}\frac{\zeta_r\tilde{y}_r}{\alpha_{opt}}$ and $\alpha_{opt} = \max\{\zeta_r, \forall r\in\mathcal{R}\}$.*

Proof. Let $\delta_{\text{opt}} = \sqrt{\frac{4\ln(R)}{\mu_{\text{opt}}}}$. Mathematically, we are interested in finding the probability that the above lemma does not hold, i.e., the following condition is true:

$$\Pr\left[\sum_{r\in\mathcal{R}}\zeta_r\hat{y}_r \leq (1-\delta_{\text{opt}})\sum_{r\in\mathcal{R}}\zeta_r\tilde{y}_r\right]. \tag{18}$$

We will apply the Chernoff bound (lower tail) [7] to find the bound. Before that, we normalize both sides of (18) as

follows:

$$\Pr\left[\sum_{r\in\mathcal{R}}\frac{\zeta_r\hat{y}_r}{\alpha_{\text{opt}}} \le (1-\delta_{\text{opt}})\sum_{r\in\mathcal{R}}\frac{\zeta_r\tilde{y}_r}{\alpha_{\text{opt}}}\right], \quad (19)$$

$$\text{where } \alpha_{\text{opt}} = \max\{\zeta_r, \forall r \in \mathcal{R}\},$$

$$\Rightarrow \Pr\left[\sum_{r\in\mathcal{R}} z_r^{\text{opt}} \le (1-\delta_{\text{opt}})\mu_{\text{opt}}\right]. \quad (20)$$

The values of $z_r^{\text{opt}} \in [0,1]$, $\forall r \in \mathcal{R}$, are independent RVs, and $\mathbb{E}\left[\sum_{r\in\mathcal{R}} z_r^{\text{opt}}\right] = \mu_{\text{opt}}$. Now, using the Chernoff bound (lower-tail) [7], we get

$$\Pr\left[\sum_{r\in\mathcal{R}} z_r^{\text{opt}} \le (1-\delta_{\text{opt}})\mu_{\text{opt}}\right] \le \exp^{\frac{-\delta_{\text{opt}}^2\mu_{\text{opt}}}{2}}, \text{ which implies}$$

$$\Pr\left[\sum_{r\in\mathcal{R}}\frac{\zeta_r\hat{y}_r}{\alpha_{\text{opt}}} \le (1-\delta_{\text{opt}})\sum_{r\in\mathcal{R}}\frac{\zeta_r\tilde{y}_r}{\alpha_{\text{opt}}}\right] \le \exp^{\frac{-\delta_{\text{opt}}^2\mu_{\text{opt}}}{2}},$$

which is equivalent to

$$\Pr\left[\sum_{r\in\mathcal{R}}\zeta_r\hat{y}_r \le (1-\delta_{\text{opt}})\sum_{r\in\mathcal{R}}\zeta_r\tilde{y}_r\right] \le \exp^{\frac{-\delta_{\text{opt}}^2\mu_{\text{opt}}}{2}}. \quad (21)$$

Now, we upper bound the right hand side of the inequality in (21) by $\frac{1}{R^2}$, and we get

$$\exp^{\frac{-\delta_{\text{opt}}^2\mu_{\text{opt}}}{2}} \le \frac{1}{R^2} \Rightarrow \delta_{\text{opt}} \ge \sqrt{\frac{4\ln(R)}{\mu_{\text{opt}}}}. \quad (22)$$

Therefore, the lowest value of δ_{opt} is $\sqrt{\frac{4\ln(R)}{\mu_{\text{opt}}}}$ for which the above condition holds. Thus, with high probability, the objective value returned by the randomized algorithm is at most $\left(1-\sqrt{\frac{4\ln(R)}{\mu_{\text{opt}}}}\right)$ times worse than the the optimal solution of the relaxed problem. □

B. Greedy Solution

The solution returned by the randomized algorithm (refer to Algorithm 1) may not be feasible if there is a violation in at least one of the capacity constraints – CPU and RAM, as mentioned in Section IV-A. Therefore, we propose a feasible solution for the given approximated solution. Algorithm 2 presents a greedy approach to obtain a feasible solution. We check the over-utilized MECs and remove requests one-by-one according to their reward[2] until the approximated solution is feasible (refer to `Step 9` in Algorithm 2). Therefore, the total objective value obtained by the greedy approach may be lower than the approximated solution. We note that any other greedy approach can be applied to obtain a feasible solution.

V. Performance Evaluation

In this section, we conduct the experiment to show the effectiveness of the proposed scheme. Table I presents the parameters and their values used for the experiment considered based on the literature [1], [3], [8], [17], [18].

[2]Request with the smallest reward is removed first.

Algorithm 2 Feasible solution: Greedy algorithm

1: Get the solution $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ from Algorithm 1
2: **for** $m \in \mathcal{M}$ **do**
3: **for** $r \in \mathcal{R}$ **do**
4: `Calculate` CPU utilization C_m^{ut}
5: `Calculate` RAM utilization D_m^{ut}
6: **while** $C_m^{\text{ut}} > C_m$ **or** $D_m^{\text{ut}} > D_m$ **do**
7: $\tilde{\mathcal{R}} \leftarrow$ `Get` requests with $\hat{x}_{r,m} = 1$
8: $\tilde{r} \leftarrow$ `Get` request from $\tilde{\mathcal{R}}$ with the lowest ζ_r
9: `Set` $\hat{x}_{r,m} = 0, \forall m \in \mathcal{M}$, and $\hat{y}_r = 0$
10: $C_m^{\text{ut}} \leftarrow C_m^{\text{ut}} - c_r$ and $D_m^{\text{ut}} \leftarrow D_m^{\text{ut}} - d_r$
11: **return** `Reward` $\leftarrow \sum_{r\in\mathcal{R}}\zeta_r\hat{y}_r$

We deploy a network with 10 MECs, in which a service request can be served from any of the MECs by placing the VNFs associated with the request. We note that the VNFs associated with two service-types are completely isolated and independent. The CPU and RAM capacities of an MEC are chosen at uniform random from the range specified in the Table I. Furthermore, the failure probabilities of MECs and VNFs are considered from [1]. We generate requests considering the VNFs required to support different uRLLC use-case scenarios [17].

TABLE I: Simulation settings

Parameter	Value
Number of MECs	10
CPU at each MEC	[32, 56] core
RAM at each CPU	[32, 80] GB
Failure of a VNF (ϵ_{v}) [1]	0.001
Failure of a PM (ϵ_{p}) [1]	0.004
Number of requests	{30, 35, 40, 50, 60}
Avail. requirements ($1-\epsilon_r$) [8]	{0.99, 0.999, 0.9999}
Reward (ζ_r)	[6, 8] $\times$ $(1-\epsilon_r)$
CPU and RAM requirements	Based on [17]

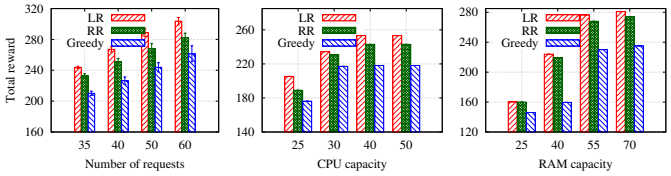
A. Results and Discussion

We present two variations of the proposed scheme — randomized rounding (Algorithm 1) and greedy algorithm (Algorithm 2). We compare the proposed scheme with the optimal solution to the relaxed problem. Henceforth, we refer `LR`, `RR`, and `Greedy` to present the optimal solution, randomized rounding solution, and greedy solution, respectively.

We take the average of 50 runs of the experiment and present the results with 95% confidence interval. We consider the following performance metrics — total reward, and percentages of resource utilization and requests served. The total reward is presented with varying number of requests and resources at the MEC. The resource utilization includes the percentage of CPU and RAM utilization at the MEC.

1) Total Reward: The service provider's objective is to maximize the total reward by serving service requests, as mentioned in Section III. To understand the impact of different number of requests and networking resources at the MECs, we present the total reward with varying number of requests

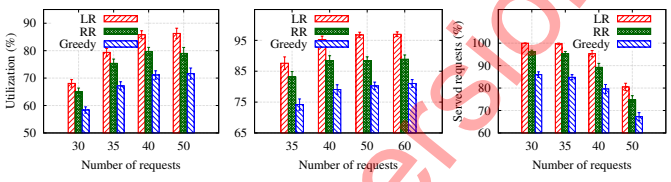
and networking resources in Figure 2 using LR, RR, and Greedy. Figures 2(a), 2(b) and 2(c) present the total reward with varying number of requests, CPU, and RAM resources, respectively, while the other resources are fixed. We see that RR and Greedy yield competitive performance to LR. Furthermore, it is observed that the total reward does not increase even after increasing the resources after a certain point. This is because more requests cannot be served due to the limitation in the other networking resources.



(a) Varying requests (b) CPU ($D_m = 48$) (c) RAM ($C_m = 40$)

Fig. 2: Total reward

2) Resource Utilization and Requests Served: Figures 3(a) and 3(b) show the percentages of CPU and RAM resource utilizations at the MECs, respectively, with varying numbers of requests. In all cases, we see that resource utilization increases with an increase in the number of requests for all schemes. This is because more requests are served by the MECs to maximize the reward. However, the utilization at the MECs gets saturated after a certain number of requests. Furthermore, we note that resource utilization for RR and Greedy is lower by 8% and 18%, respectively, than LR. This is because some requests are not served in the rounding procedure and further in the Greedy approach. Figure 3(c) shows the percentage of requests served by the service provider. It also provides competitive performance to the LR.



(a) CPU utilization (b) RAM utilization (c) Requests served

Fig. 3: Percentage of CPU and RAM utilization, and served requests

In summary, the proposed approaches, RR and Greedy, yield competitive performance to the optimal solution to the relaxed optimization problem, LR, in terms of the total reward in polynomial time.

VI. Conclusion

In this paper, we studied the VNF placement problem in MEC-enabled 5G networks while considering the reliability and latency requirements of uRLLC applications. We proposed an approximation algorithm based on the randomized rounding techniques to solve the NP-hard optimization problem in polynomial time. Furthermore, we proved the theoretical bounds of the proposed solution with respect to the optimal solution to the relaxed optimization problem. Finally, we presented simulation results to show the efficacy of the proposed approach. In this work, we considered that VNFs are independent and isolated from one service-type to another. As a future research direction, we are interested in studying the impact on the performance when VNFs are shared among multiple service-types.

Acknowledgment

The work has been supported by the project (Grant Number: SGT-100084) at IIT Jammu, India.

References

[1] L. Yala, P. A. Frangoudis, and A. Ksentini, "Latency and availability driven VNF placement in a MEC-NFV environment," in *IEEE GLOBECOM*, 2018, pp. 1–7.

[2] D. Harutyunyan, R. Behravesh, and N. Slamnik-Kriještorac, "Cost-efficient placement and scaling of 5G core network and MEC-enabled application VNFs," in *IFIP/IEEE IM*, 2021, pp. 241–249.

[3] K. Poularakis, J. Llorca, A. M. Tulino, I. Taylor, and L. Tassiulas, "Joint service placement and request routing in multi-cell mobile edge computing networks," in *IEEE INFOCOM*, 2019, pp. 10–18.

[4] "5G programmable infrastructure converging disaggregated network and compute resources," 5GPPP, Tech. Rep. D2.2, Jan. 2018.

[5] J. Navarro-Ortiz, P. Romero-Diaz, S. Sendra, P. Ameigeiras, J. J. Ramos-Munoz, and J. M. Lopez-Soler, "A survey on 5G usage scenarios and traffic models," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 905–929, 2020.

[6] P. Popovski, Č. Stefanović, J. J. Nielsen, E. de Carvalho, M. Angjelichinoski, K. F. Trillingsgaard, and A.-S. Bana, "Wireless access in ultra-reliable low-latency communication (uRLLC)," *IEEE Trans. Commun.*, vol. 67, no. 8, pp. 5783–5801, 2019.

[7] M. Mitzenmacher and E. Upfal, *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge, UK: Cambridge University Press, 2005.

[8] "Verticals uRLLC use cases and requirements," NGMN Alliance, Tech. Rep. V 2.5.4, Feb. 2020.

[9] S. Yang, F. Li, S. Trajanovski, X. Chen, Y. Wang, and X. Fu, "Delay-aware virtual network function placement and routing in edge clouds," *IEEE Trans. Mobile Comput.*, vol. 20, no. 2, pp. 445–459, 2021.

[10] R. Behravesh, E. Coronado, D. Harutyunyan, and R. Riggio, "Joint user association and VNF placement for latency sensitive applications in 5G networks," in *IEEE CloudNet*, 2019, pp. 1–7.

[11] V. Jain, H.-T. Chu, S. Qi, C.-A. Lee, H.-C. Chang, C.-Y. Hsieh, K. K. Ramakrishnan, and J.-C. Chen, "L25GC: A low latency 5G core network based on high-performance NFV platforms," in *ACM SIGCOMM*, 2022, pp. 143–157.

[12] D. Harris and D. Raz, "Dynamic VNF placement in 5G edge nodes," in *IEEE NetSoft*, 2022, pp. 216–224.

[13] R. Birke, I. Giurgiu, L. Y. Chen, D. Wiesmann, and T. Engbersen, "Failure analysis of virtual and physical machines: Patterns, causes and characteristics," in *IEEE/IFIP DSN*, 2014, pp. 1–12.

[14] S. Martello and P. Toth, *Knapsack Problems: Algorithms and Computer Implementations*. John Wiley & Sons, 1990.

[15] IBM, "CPLEX CP Optimizer," Mar. 2020.

[16] L. Comtet, *Advanced Combinatorics*. Dordrecht: Springer Netherlands, 1974.

[17] J.-J. Pedreno-Manresa, P. S. Khodashenas, M. S. Siddiqui, and P. Pavon-Marino, "On the need of joint bandwidth and NFV resource orchestration: A realistic 5G access network use case," *IEEE Commun. Lett.*, vol. 22, no. 1, pp. 145–148, 2018.

[18] Q. Zhang, F. Liu, and C. Zeng, "Adaptive interference-aware VNF placement for service-customized 5G network slices," in *IEEE INFOCOM*, 2019, pp. 2449–2457.