An Efficient Model for Mobile Network Slice Embedding under Resource Uncertainty

Andrea Fendt [1] [2], Christian Mannweiler[1], Lars Christoph Schmelz[1], Bernhard Bauer[2]
[1] Nokia Bell Labs, Network Management and Automation, Munich, Germany
{andrea.fendt, christian.mannweiler, christoph.schmelz}@nokia-bell-labs.com
[2] University of Augsburg, Department of Computer Science, Augsburg, Germany
bernhard.bauer@informatik.uni-augsburg.de

***Abstract*—The fifth generation (5G) of mobile networks will support several new use cases, like the Internet of Things (IoT), massive Machine Type Communication (mMTC) and Ultra-Reliable and Low Latency Communication (URLLC) as well as significant improvements of the conventional Mobile Broadband (MBB) use case. End-to-end network slicing is a key-feature of 5G since it allows to share and at the same time isolate resources between several different use cases as well as between tenants by providing logical network. The virtual separation of the network slices on a common end-to-end mobile network infrastructure enables an efficient usage of the underlying network resources and provides means for security and safety related isolation of the defined logical networks. A much-discussed challenge is the reuse or overbooking of resources guaranteed by contract. However, there is a consensus that over-provisioning of mobile communication bands is economically infeasible and a certain risk of network overload is acceptable for the majority of the 5G use cases. In this paper, an efficient model for mobile network slice embedding is presented which enables an informed decision on network slice admission. This is based on the guaranteed end-to-end mobile network resources that have to be provided on the one hand and the capacities and capabilities of the underlying network infrastructure on the other hand. The network slice embedding problem is solved in form of a Mixed Integer Linear Program with an uncertainty-aware objective function. Subsequently, the confidence in the availability of each resource is analyzed.**

***Index Terms*—5G, Network Slice, Virtual Network Embedding, Linear Programming, Optimization under Uncertainty**

I. Introduction and Related Work

Mobile communication channels are based on very limited frequency ranges. However, capacity demands are increasing massively. In addition to that, in most networks the peak resource consumption is unlikely to be requested by all Network Slice (NSL) simultaneously. Hence, an overprovisioning of mobile throughput resources is infeasible, non-beneficial and contradicts with user satisfactions. Careful resource overbooking on the air interface is acceptable for most 5G mobile network use cases and it is unavoidable for scalable, efficient and fair resource utilization in future 5G mobile networks. However, spectrum efficient Radio Access Network (RAN) subnet slice isolation is challenging since mobile data traffic as well as channel capacities are fluctuating. In addition, user mobility is hard to predict. [1] In this paper, a two-step approach is presented. In the first step, the best Network Slice Embedding (NSLE) regarding robustness of resource provisioning is determined. Secondly, the degree of robustness, or the risk of Service Level Agreement (SLA) violation for an embedded NSL (i.e., the probability of failure to provide the guaranteed resources) is derived. Therefor, the already deployed NSLs as well as new Network Slice Requests (NSLRs) are modeled as undirected graphs defining communication requirements as edges between mobile network subscribers and servers. Subscribers and servers are represented as nodes requiring node resources and capabilities. If the decision maker considers the robustness as good enough, i.e., if the probability that the actual required resources will be available to the NSL when requested is acceptable, then the NSLs can be deployed according to the embedding determined in the first step. This might imply an overbooking of the physical network resources.

Several promising approaches, as explained below, on assigning and overbooking virtual network resources can be found in literature. However, to the best of the authors' knowledge, none of these concepts considers the provisioning of the confidence in resource availability and the risk of violation of fixed SLAs in end-to-end mobile networks comprising the RAN, fixed networks and cloud server resources. In [2], Marotta and Kassler present a robust virtual network function placement algorithm which is based on Γ-robust optimization to protect the solution against data uncertainty. In [3], they propose an even faster three-step heuristic taking also latency constraints into account. Blanco et al. [4] present a robust Virtual Network Function (VNF) placement optimization model for optimizing the power consumption in 5G mobile networks which mitigates service demand uncertainty, while Altın et al. [5] provide a robust Virtual Network Embedding (VNE) algorithm considering communication traffic patterns. Chochlidakis et al. [6] focus on an adjustable tradeoff between robustness and resource utilization of the embedding, taking user mobility into account. Reddy, Baumgartner and Bauschert propose a similar approach in [7] using Γ-robustness optimization to handle unpredictable short-term mobile data traffic increase. Coniglio et al. [8] developed a chance-constraint formalization of the general VNE problem without latency constraints. The approaches above share the assumption that the probability distributions of the uncertain parameters are unknown. Therefore, they need to protect the solution against any possible variation within a predefined uncertainty budget.

However, this probably leads to a less beneficial solution since the objective of energy efficiency has to be balanced with the protection against resource uncertainty. In contrast to that, this work assumes that the data history on mobile network resource availability as well as resource utilization of the deployed and running NSLs is available. The data can be used to determine an estimated probability distribution for the resource availability and resource utilization. For new NSLs, that have not been deployed before, an estimation of the resource requirements can be made based on the SLA requirements, the type of the NSL and the resource utilization data of similar, already deployed NSLs.

The work of Trinh et al. [9] proposes an overbooking mechanism for virtual networks. Their work is based on soft-guaranteed service levels providing only a percentage of time with full bandwidth or service availability and a reduction factor for the limited availability. The main focus of this work is to calculate which price reduction can be offered per user to tenants that are willing to accept a predefined limitation of service quality for a single resource, like bandwidth. This paper pursuits the opposite approach: For given NSL requirements with specific resource and capability demands the best NSLE is calculated and the risk of violating the SLAs of the NSLs is determined for this possible NSLE. Several publications, for instance, Ball et al. [10] and Liu et al. [11] propose an optimal communication link overbooking ratio calculation for telecommunication networks maintaining a predefine QoS level. Sadreddini et al. [12] present a Framework for Cognitive Radio networks to find the optimal compensation rate for network overbooking using Particle Swarm Optimization. In contrast to that, this paper provides the confidence in resource availability, for incoming end-to-end mobile NSLRs with fixed SLAs. The proposed admission control algorithm takes the required resources as well as numerous further technical feasibility constraints into account. To the best of the authors' knowledge, this has not been addressed in literature yet.

II. Robust Network Slice Embedding Model

This model is based on the nearly optimal NSLE model previously published in [13] and advanced in [14]. This previous model assumes full and exact knowledge about resource availability. Its formalization is related to [15]. However, end-to-end mobile network resources, like the throughput of the communication links as well as the computation power and memory requirements on the cloud servers, underlie fluctuations and cannot be predicted accurately. For robust NSLE the expected resources are used with a buffer in the Mixed Integer Linear Program (MILP). This might lead to an overbooking of scarce network resources. However, the uncertainty-aware objective function guarantees a beneficial NSLE while avoiding unnecessary uncertainties.

The needed graph-theoretical notations are based on [16] and have been used similarly in [13] and [14]. Undirected graphs G, defined as an ordered pair $G = (\mathcal{V}, \mathcal{E})$, are used to model the physical network as well as the NSLs. A graph G is defined as a set of $n \in \mathbb{N}$ vertices $\mathcal{V} = \{v_1, v_2, \dots, v_n\}$ that are interconnected by a set of $m \in \mathbb{N}$ edges. Every edge e_{ij} has exactly two ends, one so-called start-node v_i and one so-called end-node v_j, for $i, j = 1, \dots, n$ and $i \neq j$. e_{ij} can be written as $e_{ij} := \{v_i, v_j\}$ or shorter as $e_{ij} := v_i v_j$. Since the graphs are undirected, we have $e_{ij} = e_{ji}$. Based on that, we define $N = (\mathcal{U}, \mathcal{C}, \mathcal{E})$ as a network graph, which is an undirected graph with a set of vertices $\mathcal{V} := \mathcal{U} \cup \mathcal{C}$, consisting of the User Equipments (UEs) $\mathcal{U} := \{u_1, \dots, u_n\}$ with $n \in \mathbb{N}$ and the cloud server nodes $\mathcal{C} := \{c_1, \dots, c_m\}$ with $m \in \mathbb{N}$. Without loss of generality, we assume that the edges start either in an UE node or in a cloud node, but always end in a cloud node: $\mathcal{E} \subseteq \{u_i c_j, c_k c_l\}$ for all $i = 1, \dots, n$ as well as for all $j, k, l = 1, \dots, m$ with $k \neq l$. A path $P = v_1 v_2 v_3 \dots v_n$ of length $n \in \mathbb{N}$ shall be defined as an undirected graph $P = (\mathcal{V}, \mathcal{E})$ with successively connected, pairwise different vertices $\mathcal{V} = \{v_1, v_2, \dots, v_n\}$ that are connected by the set of edges $\mathcal{E} = \{v_1 v_2, v_2 v_3, \dots v_{n-1} v_n\}$. The so-called start vertex of P is v_1, while the so-called end-vertex is v_n. The set of paths sharing the same start-vertex v_i and the same end-vertex v_j, with $i \neq j$, shall be denoted as $\mathcal{P}_{ij}$. Paths in network graphs $P_r \in \mathcal{P}_{vw}$ can start either in an UE or a cloud node $d_v \in \mathcal{U} \cup \mathcal{C}$, but must end in a cloud node $c_w \in \mathcal{C}$. The NSLE model defines a network graph $N = (\mathcal{U}, \mathcal{C}, \mathcal{E})$ for the physical network infrastructure, also referred to as the substrate, with the UEs $u_v \in \mathcal{U}$, the cloud servers $c_w \in \mathcal{C}$ and the wired and wireless communication links $e_j \in \mathcal{E}$, also referred to as edges in the following. $n \in \mathbb{N}$ virtual networks, in this case NSLs, shall be embedded into N. Whereas each NSL is modeled as an undirected graph $N_k = (\mathcal{U}_k, \mathcal{A}_k, \mathcal{L}_k)$ for $k = 1, .., n$. The set of UEs associated with an NSL $\mathcal{U}_k$ is always a subset of the UEs in the physical network: $\mathcal{U}_k \subseteq \mathcal{U}$. Each NSL has its own distinct set of applications $a_m^k \in \mathcal{A}_k$ and virtual communication links $l_i^k \in \mathcal{L}_k$. Since NSLs are isolated, they do not share applications and links. NSLE is aiming at the optimal mapping of virtual applications a_m^k to physical cloud nodes c_w and virtual links l_i^k to physical paths, with a fixed, i.e., already embedded set of UEs. This mapping is subject to numerous quality of service constraints, for instance, the throughput and reliability of the communication links and the computation power and memory of the cloud nodes. The expected available throughput of an edge e_j in the substrate is represented by a normal distribution with a mean $\mu_{T_j^s}$ and a standard deviation $\sigma_{T_j^s}$. For simplicity uplink and downlink data traffic are combined to one throughput parameter in this model. The probability distribution accounts for fluctuations in the signal quality, which results in varying available throughput. For example, the Received Signal to Noise Ratio (RSNR) and therefore the channel quality as well as the actual throughput in the RAN highly depend on, e.g., the distance and obstacles between the UE and the antenna as well as weather conditions and interferences. The link latency L_j^s of e_j is assumed to be constant. However, in practice the link latency only remains constant as long as the link throughput capacity is not exceeded and a congestion in data traffic causes an additional delay. Furthermore, the cloud servers $c_w \in \mathcal{C}$ in the substrate have a constant computation

power P_w^s and memory capacity M_w^s. In addition to that, the edges e_j as well as the node c_w have a specific availability A_j^s and B_w^s as well as reliability R_j^s and S_w^s. The NSLs require a specific maximum Latency L_i^k for each communication link l_i^k. The required throughput, however, is uncertain and therefore modeled as a normal distribution $\mathcal{N}(\mu_{T_i^k}, \sigma_{T_i^k})$ for each link $l_i^k \in \mathcal{L}_k$. Note that a standard deviation of zero represents the special case of resource certainty. Also, the required computation power and the memory capacity for the applications are defined as normal distributions: $\mathcal{N}(\mu_{P_m^k}, \sigma_{P_m^k})$ and $\mathcal{N}(\mu_{M_m^k}, \sigma_{M_m^k})$. Additionally, the mapping of the NSLs must respect a predefined link availability A_i^k, link reliability R_i^k as well as node availability B_m^k and node reliability S_m^k. The following binary and continuous embedding variables are defined for the NSLE optimization problem:

$$y_k := \begin{cases} 1 & \text{if } N_k \text{ is embedded into } N_s \\ 0 & \text{otherwise} \end{cases}$$

$$a2c_{mw}^k := \begin{cases} 1 & \text{if } a_m^k \text{ is mapped on } c_w \\ 0 & \text{otherwise} \end{cases}$$

$$l2p_{ir}^k \in [0,1] \text{ percentage of data transfer of } l_i^k \text{ mapped on } P_r \in \mathcal{P}_{vw}$$

$$p2e_{rj} := \begin{cases} 1 & \text{if } e_j \text{ is used in } P_r \\ 0 & \text{otherwise} \end{cases}$$

$$l2e_{ij}^k := \sum_r (l2p_{ir}^k \cdot p2e_{rj})$$

For a given substrate the $p2e_{rj}$ mapping is known and not subject to optimization. The $l2e_{ij}^k$ mapping results from the $l2p_{ir}^k$ mapping combined with the $p2e_{rj}$ mapping. In order to maintain a Linear Program that can be solved efficiently the uncertainty in the resource availability and utilization is addressed in the objective function only. The objective function ensures that as many NSLs are embedded as possible. The most beneficial ones are selected, if there are not enough resources and the allocation minimizes uncertainty. It comprises four subfunctions f_{rev} for the embedding revenue, f_{thr} for the throughput uncertainty, f_{cpu} for the CPU uncertainty and f_{mem} for the memory uncertainty. $f_{rev}\left((y_k)\right) := \sum_k \frac{\omega_k}{\beta_1} \cdot y_k$ $f_{rev}\left((y_k)\right)$ sums up the weights of the embedded NSLs, which shall be maximized. (y_k) refers to the vector of the embedding variables $(y_k) := (y_1, y_2, \ldots, y_n)$. The weights originally given to the NSLs are normalized with a normalization factor β_1. $f'_{thr}\left(\left(l2e_{ij}^k\right)\right) := -\sum_j \frac{\max\left(\mu_{T_j^s} - \sigma_{T_j^s}\alpha_1, \epsilon\right) - \left(\sum_{k,i} l2e_{ij}^k (\mu_{T_i^k} + \sigma_{T_i^k}\alpha_1)\right)}{\max\left(\mu_{T_j^s} - \sigma_{T_j^s}\alpha_1, \epsilon\right)}$

$f_{thr}\left(\left(l2e_{ij}^k\right)\right) := -\sum_{k,i,j} l2e_{ij}^k \cdot \frac{\mu_{T_i^k} + \sigma_{T_i^k}\alpha_1}{\max\left(\mu_{T_j^s} - \sigma_{T_j^s}\alpha_1, \epsilon\right)\cdot\beta_2}$

The second part of the objective function, denoted as $f_{thr}\left(\left(l2e_{ij}^k\right)\right)$, is derived from the function $f'_{thr}\left(\left(l2e_{ij}^k\right)\right)$. $\left(l2e_{ij}^k\right)$ stands for the hyper-matrix of all link to edge mapping variables. $f'_{thr}\left(\left(l2e_{ij}^k\right)\right)$ minimizes the sum of proportions of used throughput plus the standard deviation of the throughput for that link in the NSL multiplied by a factor α_1. If the mean throughput of a physical link is smaller than or equal to its standard deviation, then a small ϵ is used to prevent dividing by zero or negative values. $\epsilon > 0$ shall be defined as a very small positive double value. $\epsilon = 1 \cdot 10^{-10}$ has been used in the evaluation below. The term $f'_{thr}\left(\left(l2e_{ij}^k\right)\right)$ is simplified to the convex combination $f_{thr}\left(\left(l2e_{ij}^k\right)\right)$ by removing unnecessary constants and dividing by a suitable normalization factor β_2.

$f'_{cpu}\left(\left(a2c_{mw}^k\right)\right) := -\sum_w \frac{P_w^s - \left(\sum_{k,m} a2c_{mw}^k \cdot \left(\mu_{P_m^k} + \sigma_{P_m^k}\alpha_2\right)\right)}{P_w^s}$

$f_{cpu}\left(\left(a2c_{mw}^k\right)\right) := -\sum_{k,m,w} a2c_{mw}^k \cdot \frac{\mu_{P_m^k} + \sigma_{P_m^k}\alpha_2}{P_w^s \cdot \beta_3}$

Similarly, the functions $f_{cpu}\left(\left(a2c_{mw}^k\right)\right)$ and $f_{mem}\left(\left(a2c_{mw}^k\right)\right)$ are responsible for minimizing the total sum of all used computation power and memory shares plus the factorized standard deviations. $f'_{mem}\left(\left(a2c_{mw}^k\right)\right) := -\sum_w \frac{M_w^s - \left(\sum_{k,m} a2c_{mw}^k \cdot \left(\mu_{M_m^k} + \sigma_{M_m^k}\alpha_3\right)\right)}{M_w^s}$

$f_{mem}\left(\left(a2c_{mw}^k\right)\right) := -\sum_{k,m,w} a2c_{mw}^k \cdot \frac{\mu_{M_m^k} + \sigma_{M_m^k}\alpha_3}{M_w^s \cdot \beta_4}$

The robust NSLE objective function is defined as follows:

$$\begin{aligned} \max\ & \rho_1 \cdot f_{rev}\left((y_k)\right) + \rho_2 \cdot f_{thr}\left(\left(l2e_{ij}^k\right)\right) \\ & + \rho_3 \cdot f_{cpu}\left(\left(a2c_{mw}^k\right)\right) + \rho_4 \cdot f_{mem}\left(\left(a2c_{mw}^k\right)\right) \end{aligned} \tag{1}$$

The weights ρ_1, ρ_2, ρ_3 and $\rho_4 \in [0,1]$ associated to the four subfunctions sum up to one. For the evaluation below the so-called embedding importance ρ_1 is set to 0.85. The throughput, cpu and memory importance represented by ρ_2, ρ_3 and ρ_4 are set to 0.05. Moreover, the factors α_1, α_2 and α_3 are set to 1. The above objective function is subject to the following constraints:

$$\sum_w a2c_{mw}^k = y_k,\ \forall k, m \tag{2}$$

$$\sum_{P_r \in \mathcal{P}_{vw}} l2p_{ir}^k = y_k,\ \forall k, i \text{ with } l_i^k = \{u_v, a_m^k\} \tag{3}$$

$$\sum_{P_r \in \mathcal{P}_{vw}} l2p_{ir}^k = a2c_{bv}^k,\ \forall k, i \text{ with } l_i^k = \{a_b^k, a_m^k\} \tag{4}$$

$$\sum_{P_r \in \mathcal{P}_{vw}} l2p_{ir}^k = a2c_{mw}^k,\ \forall k, i \text{ with } l_i^k = \{f_v^k, a_m^k\} \tag{5}$$

$$\sum_k \sum_i l2e_{ij}^k \left(\mu_{T_i^k} + \gamma \cdot \sigma_{T_i^k}\right) \leq \mu_{T_j^s} - \gamma \cdot \sigma_{T_j^s},\ \forall j \tag{6}$$

$$\sum_k \sum_m a2c_{mw}^k \left(\mu_{P_m^k} + \gamma \cdot \sigma_{P_m^k}\right) \leq P_w^s,\ \forall w \tag{7}$$

$$\sum_k \sum_m a2c_{mw}^k \left(\mu_{M_m^k} + \gamma \cdot \sigma_{M_m^k}\right) \leq M_w^s,\ \forall w \tag{8}$$

$$\sum_j l2e_{ij}^k \cdot L_j^s \leq L_i^k \cdot l2p_{ir}^k,\ \forall k, i, P_r \in \mathcal{P} \tag{9}$$

$$a2c_{mw}^k \cdot B_m^k \leq B_w^s, \quad \forall k, m, w \tag{10}$$

$$a2c_{mw}^k \cdot S_m^k \leq S_w^s, \quad \forall k, m, w \tag{11}$$

$$l2e_{ij}^k \cdot A_i^k \leq l2e_{ij}^k \cdot A_j^s, \quad \forall k, i, j \tag{12}$$

$$l2e_{ij}^k \cdot R_i^k \leq l2e_{ij}^k \cdot R_j^s, \quad \forall k, i, j \tag{13}$$

Eq. 2 specifies the map-once constraints, stating that every application must be mapped exactly once, if the corresponding

NSL has been embedded. The graph constraints in eq. 3 to eq. 5 make sure that the physical paths and cloud nodes the virtual links and applications are mapped to are connected accordingly. The ineq. 6 to ineq. 8 model the resource availability constraints using the expected mean available resources reduced by a safety discount and the resource demand increased by a safety buffer. The safety discount/buffer is defined as a factor γ multiplied with the standard deviation of the respective resource availability or demand probability distribution. A higher γ can improve the robustness of the embedding, whereas a lower γ might result in a higher benefit for the network operator, but requires a higher risk tolerance. For the following evaluation γ is set to 1.5. Ineq. 9 models the latency constraints and ineq. 10 to 13 define the network quality constraints. A more detailed explanation of these linear constraints can be found in [14]. The model as described above is used to determine a nearly optimal NSLE using the most stable network resources. To provide a beneficial solution, the expected resource demand and provisioning are used with a safety buffer instead of the worst case demand and availability. This may leads to a resource overbooking and resource availability violations can occur. The probability of meeting the resource constraints and the according risk of SLA violation can be evaluated as follows: The provisioning of an uncertain resource R is assumed to be Gaussian distributed with $\mathcal{N}(\mu_R, \sigma_R)$. That means, it is expected that the actually provided amount of resource is probably close to the expected value. However, an arbitrary distribution functions could be used with this approach. The resource is used by several NSLs with uncertain demands, also modeled with Gaussian distributions. $\mathcal{N}(\mu_{D_1}, \sigma_{D_1}), \mathcal{N}(\mu_{D_2}, \sigma_{D_2}), \ldots, \mathcal{N}(\mu_{D_n}, \sigma_{D_n})$. Then the overall demand for R is Gaussian distributed with $\mathcal{N}(\sum_i^n \mu_{D_i}, \sum_i^n \sigma_{D_i})$ and the residual resources of R are also Gaussian distributed with $\mathcal{N}(\mu_R - \sum_i^n \mu_{D_i}, \sigma_R + \sum_i^n \sigma_{D_i})$. Thus, the Probability of Feasibility (PoF) for meeting the constraint requirements for R for the embedded NSLs is the probability that the residual resources are greater or equal to zero: $PoF_R := \int_0^\infty \mathcal{N}(\mu_R - \sum_i^n \mu_{D_i}, \sigma_R + \sum_i^n \sigma_{D_i})$. The PoF of an NSL resource constraint is calculated for each resource as well as for each communication link or node of the requested NSL. For instance, the required throughput of an NSL link l_i^k has been assumed to be normal distributed with a mean $\mu_{T_i^k}$ and a standard deviation $\sigma_{T_i^k}$. The mapping algorithm determines an $l2e_{ij}^k \in [0, 1]$, if path-splitting is enabled, for each edge e_j in the substrate network. The expected (mean) throughput utilization for l_i^k is scaled with the proportion of usage $l2e_{ij}^k$ before considered in calculating the residual throughput resource availability:

$$PoF_{T_i^k} := \int_0^\infty \mathcal{N}\left(\mu_{T_j^s} - \sum_{k,i} l2e_{ij}^k \cdot \mu_{T_i^k}, \sigma_{T_j^s} + \sum_{k,i} \sigma_{T_i^k}\right)$$

Since the CPU and the memory provided by the cloud serves are certain, but the resource demands can deviate from the expectations, the PoF for those resource is defined as follows:

$$PoF_{P_m^k} := \int_0^\infty \mathcal{N}\left(P_w^s - \sum_{k,m} a2c_{mw}^k \cdot \mu_{P_m^k}, \sum_{k,m} \sigma_{P_m^k}\right)$$

$$PoF_{M_m^k} := \int_0^\infty \mathcal{N}\left(M_w^s - \sum_{k,m} a2c_{mw}^k \cdot \mu_{M_m^k}, \sum_{k,m} \sigma_{M_m^k}\right)$$

The PoF_T has to be calculated for each virtual link l_i^k in every NSL. The PoF_P and PoF_M are determined for every application node a_m^k in every NSL. In order to evaluate the confidence in meeting the requirements of an NSL, a box-plot for the PoFs can be created for each resource within an NSL. In addition to that, when assuming stochastic independence between the resource provisioning/demand of the network elements, the PoFs can simply be multiplied to determine the overall PoF for each resource and a whole NSL.

III. Implementation and Evaluation

For this evaluation, a dedicated java program and the Solving Constraint Integer Programs (SCIP) Solver [17] is used. All evaluation results have been collected on an ordinary MacBook Pro with a 3.1 GHz Intel Core i7 and a 16 GB 1867 MHz DDR3. Solving medium sized problem instances with the SCIP solver takes between about 2.1 and 7.8 minutes, while the actual solving time only consumes 3.5 seconds in average (see table I). For the evaluation below 3 medium sized, randomly generated NSLE problems have been used. The substrates consist of 30 UEs and 31 cloud server nodes, connected by 60 communication links. 10 NSLs shall be embedded into these substrates, containing about 5 UEs, 12 application nodes and 14 links per NSL in average. This results in a large number of variables and constraints to be prepared, when setting up the MILP. For this evaluation randomly generated, relative values are used for the parameters of the model. For instance, the probability distributions for the available throughput of the edges in the substrate network have a random mean from the interval $(100, 150)$ and a random standard deviation between 10 and 15. The probability distributions of the required throughput on the links in the NSLs are generated similarly from the interval $(20, 30)$ for the mean and $(2, 3)$ for the standard deviation. Table I shows the preparation and solving time of the three examples in seconds, as well as the overall confidences of the embedded NSLs. The preparation of the MILP consumes a great share of the overall computation time. The full constraint matrix cannot be stored at once in the RAM. Due to the Java Interface of the SCIP solver, the constraints must be fed one by one into the solver, which drastically slows down the preparation. The uncertainty-sensitive objective function UNCERTAIN (UNC.), as proposed in this paper, is compared to a simple objective function SIMPLE (SIM.), as presented in [13]. γ is set to 1.5 for the UNC. and 0 for the SIM. case. The SIM. objective function solely consists of the f_{rev} subfunction of the UNC. objective function in eq. 1. The results in table I also support the expectation that the solving time for the MILP is usually

TABLE I
RUNTIME AND RESOURCE CONFIDENCE ANALYSIS

Nb.	Obj.	Runtime in sec. Prep.-t.	Sol.-t.	PoF per Network Slice 0	1	2	3	4	5	6	7	8	9
1	UNC.	262.5	3.1	-	0.9967649	0.9955543	0.9999924	0.9930735	-	-	0.9961224	-	0.9941126
	SIM.	277.2	2.2	-	0.1600463	0.2584679	0.1957743	0.1182899	-	0.0730512	0.0911598	0.0116235	0.0189479
2	UNC.	461.7	8.8	-	-	0.9821750	0.98340937	0.9841528	0.9547410	-	-	0.9655072	0.9911498
	SIM.	402.8	4.5	-	0.0007363	0.2265701	0.2437432	0.0032591	0.0019295	-	-	0.0312014	0.9644017
3	UNC.	168.5	1.0	0.9993309	-	-	0.9983087	-	-	-	-	0.9994359	0.9999998
	SIM.	123.1	1.1	0.0104692	-	-	0.0918219	-	0.0677698	0.1032131	-	0.2650311	0.1690731

higher when using the uncertainty-aware objective function UNC. instead of its SIM. version. In addition to that, table I shows that the PoFs are strongly improved by using the robust MILP with the UNC. objective function, as presented in section II. For instance, the overall confidence of slice 1 when using the SIM. objective function is only 0.1600. When using the UNC. objective function it is increased to 0.9967649. That means, under the assumption of stochastic independence, the probability that all resources are available simultaneously is above 99%. This is derived from the product of all throughput confidence values $PoF_{T_i^k}$ of about 0.9967649126 for the used communication links as well as the product of all memory and CPU confidence values for the allocated cloud server nodes $PoF_{M_m^k} = 0.9999999999$ and $PoF_{P_m^k} = 1.0$. Some of the NSLs could not be embedded due to a lack of resources or due to violating other constraints like latency. Consequently, there are no confidence values provided for these NSLs in table I.

IV. CONCLUSION AND OUTLOOK

In this paper an efficient model for mobile NSLE under resource uncertainty, also taking strict latency and real-time communication constraints into account, has been presented. The evaluation shows that the proposed approach is a reliable mechanism for mobile NSLE in 5G. The suggested resource allocation is robust to uncertainty in expected resource and demand fluctuations derived from network performance data analysis and predictions on future behavior. It helps finding an NSLE with the best balance between resource efficiency and robustness. In future work, further evaluation on the scalability of the MILP-based NSLE has to be performed. Heuristics should to be considered for large NSLE problem instances. For instance, splitting large problem instances into smaller local subproblem, which can be solved in a small fraction of the time.

REFERENCES

[1] M. Richart, J. Baliosian, J. Serrat, and J.-L. Gorricho, "Resource slicing in virtual wireless networks: A survey," *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 462–476, Sep. 2016.

[2] A. Marotta and A. Kassler, "A power efficient and robust virtual network functions placement problem," in *2016 28th International Teletraffic Congress (ITC 28)*, Würzburg, Germany: IEEE, Sep. 2016, pp. 331–339.

[3] A. Marotta, E. Zola, F. D'Andreagiovanni, and A. Kassler, "A fast robust optimization-based heuristic for the deployment of green virtual network functions," *Journal of Network and Computer Applications*, vol. 95, pp. 42–53, Oct. 2017, ISSN: 10848045.

[4] B. Blanco, I. Taboada, J. O. Fajardo, and F. Liberal, "A robust optimization based energy-aware virtual network function placement proposal for small cell 5g networks with mobile edge computing capabilities," *Mobile Information Systems*, vol. 2017, pp. 1–14, 2017.

[5] A. Altın, E. Amaldi, P. Belotti, and M. Ç. Pınar, "Provisioning virtual private networks under traffic uncertainty," *Networks*, vol. 49, no. 1, pp. 100–115, Jan. 1, 2007.

[6] G. Chochlidakis and V. Friderikos, "Robust virtual network embedding for mobile networks," in *2015 IEEE 26th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Aug. 2015, pp. 1867–1871.

[7] A. Baumgartner, T. Bauschert, A. A. Blzarour, and V. S. Reddy, "Network slice embedding under traffic uncertainties — a light robust approach," in *2017 13th International Conference on Network and Service Management (CNSM)*, Tokyo: IEEE, Nov. 2017, pp. 1–5.

[8] S. Coniglio, A. Koster, and M. Tieves, "Data uncertainty in virtual network embedding: Robust optimization and protection levels," *Journal of Network and Systems Management*, vol. 24, no. 3, pp. 681–710, Jul. 2016.

[9] H. E. T. Trinh and C. Aswakul, "Quality of service using careful overbooking for optimal virtual network resource allocation," in *The 8th Electrical Engineering/ Electronics, Computer, Telecommunications and Information Technology (ECTI) Association of Thailand - Conference 2011*, 2011, pp. 296–299.

[10] R. Ball, M. Clement, F. Huang, Q. Snell, and C. Deccio, "Aggressive telecommunications overbooking ratios," in *IEEE International Conference on Performance, Computing, and Communications, 2004*, Apr. 2004, pp. 31–38.

[11] J. Liu, X. Jiang, and S. Horiguchi, "Opportunistic link overbooking for resource efficiency under per-flow service guarantee," *IEEE Transactions on Communications*, vol. 58, no. 6, pp. 1769–1781, Jun. 2010.

[12] Z. Sadreddini, E. Güler, and T. Çavdar, "PSO-optimized instant overbooking framework for cognitive radio networks," in *2015 38th International Conference on Telecommunications and Signal Processing (TSP)*, Jul. 2015, pp. 49–53.

[13] A. Fendt, S. Lohmuller, L. C. Schmelz, and B. Bauer, "A network slice resource allocation and optimization model for end-to-end mobile networks," in *2018 IEEE 5G World Forum (5GWF)*, Jul. 2018, pp. 262–267.

[14] A. Fendt, C. Mannweiler, L. C. Schmelz, and B. Bauer, "A formal optimization model for 5g mobile network slice resource allocation," in *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, Nov. 2018, pp. 101–106.

[15] Z. Despotovic, A. Hecker, A. N. Malik, R. Guerzoni, I. Vaishnavi, R. Trivisonno, and S. A. Beker, "VNetMapper: A fast and scalable approach to virtual networks embedding," in *2014 23rd International Conference on Computer Communication and Networks (ICCCN)*, China: IEEE, Aug. 2014, pp. 1–6.

[16] R. Diestel, *Graphentheorie*, 3., neu bearb. und erw. Aufl. Berlin: Springer, 2006.

[17] A. Gleixner et al., "The SCIP Optimization Suite 6.0," Optimization Online, Technical Report, Jul. 2018. [Online]. Available: http://www.optimization-online.org/DB_HTML/2018/07/6692.html.