A Matheuristic for Green and Robust 5G Virtual Network Function Placement

Thomas Bauschert[1], Fabio D'Andreagiovanni[2,3(✉)], Andreas Kassler[4], and Chenghao Wang[3]

[1] Chair of Communication Networks, Technische Universität Chemnitz, 09126 Chemnitz, Germany
thomas.bauschert@etit.tu-chemnitz.de
[2] French National Center for Scientific Research (CNRS), Paris, France
[3] Sorbonne Universités, Université de Technologie de Compiègne, CNRS, Heudiasyc UMR 7253, CS 60319, 60203 Compiègne, France
{d.andreagiovanni,chenghao.wang}@hds.utc.fr
[4] Karlstad University, Universitetsgatan 2, 65188 Karlstad, Sweden
andreas.kassler@kau.se

Abstract. We investigate the problem of optimally placing virtual network functions in 5G-based virtualized infrastructures according to a green paradigm that pursues energy-efficiency. This optimization problem can be modelled as an articulated 0-1 Linear Program based on a flow model. Since the problem can prove hard to be solved by a state-of-the-art optimization software, even for instances of moderate size, we propose a new fast matheuristic for its solution. Preliminary computational tests on a set of realistic instances return encouraging results, showing that our algorithm can find better solutions in considerably less time than a state-of-the-art solver.

Keywords: 5G · Virtual Network Function · Traffic uncertainty · Robust Optimization · Matheuristic

1 Introduction

The Fifth Generation of wireless telecommunications systems, widely known as 5G, has attracted a lot of attention in recent times, since it is largely considered a crucial element for a full realization of a digital society and a critical technology to support the deployment of smart cities [1]. 5G is going to offer enhanced service performances unknown to previous wireless technologies, such as data rates of at least 40 Mbps for tens of thousands of users, data rates of 100 Mbps for metropolitan areas, enhanced spectral efficiency and a dramatic reduction

This work has been partially carried out in the framework of the Labex MS2T program. Labex MS2T is supported by the French Government, through the program "Investments for the future", managed by the French National Agency for Research (Reference ANR-11-IDEX-0004-02).

P. Kaufmann and P. A. Castillo (Eds.): EvoApplications 2019, LNCS 11454, pp. 430–438, 2019.
https://doi.org/10.1007/978-3-030-16692-2_29

of latency (see e.g., [1]). In particular, 5G will be strongly based on Network Function Virtualization, according to which network functions run on a set of virtual machines (VMs) that are hosted in cheap commodity hardware servers [2]. This will considerably reduce the cost of network infrastructures, decreasing the need for expensive dedicated hardware. The problem of optimally designing virtual networks, allocating Virtual Network Functions Components (VNFCs) to physical servers and managing the data flows between servers has received great attention in recent times, in particular focusing on adopting a green networking perspective aimed at minimizing the overall power consumption (see e.g., [3–8]). However, while in the available literature purely heuristic solution approaches for virtual network design have been quite widely investigated, the development of hybrid exact-heuristic algorithms exploiting the potentialities of mathematical programming (so-called matheuristic - see [9]) has received very limited attention. By this work, we aim to start to fill this gap by proposing a new matheuristic for the green placement of virtual network function in 5G, while taking into account the uncertainty of function requests (see e.g., [4,6]).

The remainder of this short paper is organized as follows: in Sect. 2, we describe a Binary Linear Programming model for modelling the green and robust placement of VNCFs; in Sect. 3, we present a new matheuristic to fast solve the placement problem: finally, in Sect. 4, we report preliminary computational results and derive some conclusions.

2 A Binary Linear Program for VNFC Placement

From a modelling point of view, we can essentially describe the topology of the 5G network that we consider through a graph $G(N, L)$, where N is the node set and L is the link set. Each link $\ell \in L$ corresponds to a pair (i, j), where $i, j \in N$ are the nodes it connects. Each link is associated with a bandwidth b_ℓ. The network interconnects a set of servers S and the node to which a server s is connected is denoted by $n(s) \in N$. Each server offers an amount of computational resources (e.g., CPU and RAM): denoting by R the set of resource types, the amount of resources available for each type $r \in R$ at a server $s \in S$ is denoted by a_{sr}. The set of VNFCs is denoted by V and the set of service chains offered in the network is denoted by $\mathcal{C}$. When executed, a VNFC $v \in V$ requires an amount a_{vr} of each resource type $r \in R$. Each chain $C \in \mathcal{C}$ corresponds to a subset of pairs (v_1, v_2) belonging to $V \times V$. The exchange of data between v_1 and v_2 in a pair (v_1, v_2) requires an amount of bandwidth $b_{v_1 v_2}$ in each traversed network link. Concerning power consumption, every node $n \in N$ and link $\ell \in L$ consumes P_n and P_ℓ when used, respectively. Each server $s \in S$ has a consumption that is a linear function in the range $[P_s^{\min}, P_s^{\max}]$.

The optimization problem related to VNFC placement that we consider can be resumed as follows: given a 5G network interconnecting a set of servers, we want to decide how to establish a set of virtual chains in the network,

respecting the available resource budget of the servers, while minimizing the overall power consumption. The decisions taken are modelled by the following decision variables:

(1) variables $y_s \in \{0,1\}$, $\forall s \in S$ representing the activation of a server ($y_s = 1$ if s is turned on and 0 otherwise);
(2) variables $x_{vs} \in \{0,1\}$, $\forall v \in V, s \in S$ representing the allocation of a VNFC v to server s ($x_{vs} = 1$ if v is allocated to s and 0 otherwise);
(3) variables $z_n \in \{0,1\}$, $\forall n \in N$ representing the activation of a node n ($z_n = 1$ if n is turned on and 0 otherwise);
(4) variables $w_{ij} \in \{0,1\}$, $\forall (i,j) \in L$ representing the activation of a link $\ell = (i,j)$ ($w_{ij} = 1$ if $\ell = (i,j)$ is turned on and 0 otherwise);
(5) variables $f_{ij}^{(v_1,v_2)} \in \{0,1\}$, $\forall (i,j) \in L, (v_1,v_2) \in \bigcup_{C \in \mathcal{C}}$ representing that link (i,j) is used for data exchange between v_1 and v_2 belonging to some $C \in \mathcal{C}$.

These variables are employed in the following Binary Linear Program, denoted by BLP-VP, modelling the VNFC optimal placement problem:

$$\max \sum_{s \in S} \left[P_s^{\min} \cdot y_s + (P_s^{\max} - P_s^{\min}) \cdot \frac{1}{a_{rs}} \cdot \sum_{v \in V} a_{vr} \cdot x_{vs} \right] + \sum_{n \in N} P_n \cdot z_n + \sum_{(i,j) \in L} P_{ij} \cdot w_{ij} \quad (\text{with } r = \text{CPU}) \qquad \text{(BLP-VP)} \qquad (1)$$

$$\sum_{s \in S} x_{vs} = 1 \qquad v \in V \qquad (2)$$

$$y_s \leq \sum_{v \in V} x_{vs} \qquad s \in S \qquad (3)$$

$$x_{vs} \leq y_s \qquad s \in S, v \in V \qquad (4)$$

$$\sum_{v \in V} a_{vr} \cdot x_{vs} \leq a_{rs} \cdot y_s \qquad s \in S, r \in R \qquad (5)$$

$$\sum_{(n,i) \in L} b^{v_1,v_2} \cdot f_{ni}^{v_1,v_2} - \sum_{(i,n) \in L} b^{v_1,v_2} \cdot f_{in}^{v_1,v_2} = \sum_{s \in S : n(s) = n} b^{v_1,v_2} \cdot (x_{v_1 s} - x_{v_2 s}) \qquad n \in N, (v_1,v_2) \in \bigcup_{C \in \{\mathcal{C}\}} C \qquad (6)$$

$$\sum_{(v_1,v_2) \in \bigcup_{C \in \{\mathcal{C}\}} C} b^{v_1,v_2} f_{ij}^{v_1,v_2} \leq B_{ij} w_{ij} \qquad (i,j) \in L \qquad (7)$$

$$w_{ij} \leq z_i \text{ and } w_{ij} \leq z_j \qquad (i,j) \in L \qquad (8)$$

$$f_{ij}^{v_1,v_2} \leq z_i \text{ and } f_{ij}^{v_1,v_2} \leq z_j \qquad (i,j) \in L \qquad (9)$$

$$y_s, x_{vs}, z_n, w_{i,j}, f_{i,j}^{v_1,v_2} \in \{0,1\} \qquad s \in S, v \in V, n \in N, \\ (i,j) \in L, (v_1,v_2) \in \bigcup_{C \in \{\mathcal{C}\}} C.$$

The previous model is based on the model proposed in [6], to which we refer the reader for a detailed description of all its elements. The objective function (1) pursues the minimization of the total power consumption, expressed as the sum of the power consumed by servers, nodes and links. The constraints (2) impose that each VNFC must be allocated on exactly one server. The constraints (3) and (4) logically link the values of the server activation and VNFC allocation decision variables. The constraints (5) model the resource capacity for each server and resource type, imposing that the overall resource usage of all the VNFCs cannot exceed the capacity of each server. The constraints (6) express the usage of bandwidth on links of the network, under the form of flow conservation constraints with a flow balance in the right hand side that takes into account the variable allocation of VNFCs to servers. The bandwidth capacity of links is modelled by the constraints (7). Finally, the constraints (8) and (9) link the link and node activation decision variables, imposing that links are activated if and only if the corresponding nodes are activated.

Protecting Against Resource Uncertainty. As in [6], we make the resource capacity constraints (5) robust against fluctuations in the resource requests a_{vr}. Indeed, the resource need for virtual chains requests is typically not exactly known in advance. Taking into account such data uncertainty in the model is very important, since by neglecting the possibility of variations in the input data we risk to obtain design solutions that are of bad quality and even infeasible in practice. For a discussion about the effects of the presence of data uncertainty in mathematical optimization and (telecommunications) network design, we refer the reader to the works [10,11]. In order to protect against resource uncertainty, we adopt a Robust Optimization (RO) paradigm (see e.g., [10]). RO, which has been highly appreciated for its high computational efficiency with respect to more traditional paradigms like Stochastic Programming, essentially takes into account data uncertainty by including additional hard constraints in the optimization problem. These constraints have the task of excluding solutions that are vulnerable to input data deviations, maintaining only *robust solutions*. The data deviations that are relevant to the decision maker and against which protection is needed are specified through a so-called *uncertainty set*. The specific RO model that we consider is Γ-Robustness [12], which belongs to the family of *cardinality-constrained* uncertainty sets and, adapting to our case, assumes that each value a_{vr} may vary in a range $[\bar{a}_{vr} - \Delta a_{vr}, \bar{a}_{vr} + \Delta a_{vr}]$ centered on a reference value $\bar{a}_{vr}$ that may deviate up to $\Delta a_{vr} > 0$. Furthermore, the uncertainty model assumes that at most Γ coefficients in every (5) may vary.

Under these modelling assumptions, the robust version of the constraints (5) can be obtained according to the procedure detailed in [6]. Specifically, each constraint (5) must be replaced by the following set of constraints and additional

decision variables:

$$\sum_{v \in V} \bar{a}_{vr} \cdot x_{vs} + \left(\Gamma \cdot v_{rs} + \sum_{v \in V} w_{rsv} \right) \leq a_{rs} \cdot y_s$$

$$\begin{aligned} & v_{rs} + w_{rsv} \geq \Delta a_{vr} \cdot x_{vs} && v \in V \\ & v_{rs} \geq 0 \\ & w_{rsv} \geq 0 && v \in V. \end{aligned} \tag{10}$$

The robust model that we solve in what follows is thus (BLP-VP) with (5) replaced with (10). We denote such robust model by (ROB-BLP-VP).

3 A Matheuristic for ROB-BLP-VP

We present here a new matheuristic for optimal VNFC placement that is based on the integration of a Genetic Algorithm (GA) with an *exact* large neighborhood search, namely a search formulated as an optimization problem solved by a state-of-the-art solver such as CPLEX [13]. The solver is also used for completing partial solutions of (ROB-BLP-VP) in an optimal way: for a fixed value configuration of a subset of decision variables, we employ the solver to find a feasible valorization of all the remaining variables while optimizing the objective function. At the basis of this matheuristic there is the consideration that, while a state-of-the-art solver may find difficulties in identifying good quality solutions for ROB-BLP-VP, it is instead able to efficiently identify good solutions for appropriate subproblems of ROB-BLP-VP, derived by fixing the value of a consistent subset of variables.

GAs are widely known metaheuristics that draw inspiration from the evolution of a population (see [14] for an exhaustive introduction to the topic). The individuals of the population represent solutions of the optimization problem and the *chromosome* of an individual corresponds to a valorization of decision variables of a solution. The quality of an individual/solution is assessed through a *fitness function*. A GA begins with the definition of an initial population that then changes through evolutionary mechanisms like crossover and mutation of individuals, until some stopping criterion is met.

3.1 Initialization of the Population

Solution Representation. The first step consists of establishing what the individuals of the population represent. We decided that the chromosome of an individual corresponds with a valorization of the decision variables (y, x) (of ROB-BLP-VP), which represent the server activation and the VNFC allocation. Indeed, such variables are particularly critical for the problem: once their values have been fixed, we obtain a subproblem of (ROB-BLP-VP) that reduces to a

kind of robust network flow problem and is easier to be solved by a state-of-the-art optimization solver, returning an optimal solution for (ROB-BLP-VP) using the valorization of (y, x) as basis.

Fitness Function. To assess the quality of an individual, we adopt a fitness function that corresponds to the objective function (1) of (ROB-BLP-VP).

Initial Population. The strategy that we explored to generate the initial group of individuals relies on the following principles: to generate an individual, we randomly activate a number $\sigma < |S|$ of servers and then we randomly assign each VNFC in V to one single activated server, checking that the resource constraints (5) are not violated. In this way, we obtain a valorization $(\bar{y}, \bar{x})$ of the server and allocation variables that we can then complete by solving the remaining subproblem of (ROB-BLP-VP) through a state-of-the-art solver. By this strategy, we can obtain the optimal solution of (ROB-BLP-VP) for a fixed $(\bar{y}, \bar{x})$. We denote the set of individuals constituting the population at a generic iteration of the algorithm by POP.

3.2 Evolution of the Population

Selection. The individuals chosen for being combined and generating the new individuals are chosen according to a *tournament selection* principle: we first create a number β of (small cardinality) groups of individuals by randomly selecting them from POP. Then the γ individuals in each group associated with the best fitness value are combined through crossover.

Crossover. We form the couples that generate the offsprings according to the following procedure. From the tournament selection, we obtain $\beta \cdot \gamma$ individuals that are randomly paired in couples, each generating one offspring. Assuming that the two parents are associated with chromosomes/partial solutions (y^1, x^1) and (y^2, x^2), the chromosome of the offspring $(y^{\text{off}}, x^{\text{off}})$ is defined according to two rules: (1) if the parents have the same binary value in a position j, then the offspring inherits such value in its position j (i.e., if $(y^1, x^1)_j = (y^2, x^2)_j$ then $(y^{\text{off}}, x^{\text{off}})_j = (y^1, x^1)_j$); (2) if the parents have distinct binary values in a position j, then the offspring inherits a null value (i.e., if $(y^1, x^1)_j \neq (y^2, x^2)_j$ then $(y^{\text{off}}, x^{\text{off}})_j = 0$). Possible violations in the constraints (2) and (5) associated with $(y^{\text{off}}, x^{\text{off}})$ are then repaired. The main rationale at the basis of this procedure is assuming that two solutions having the same valorization of a variable is a good indication that such valorization should be maintained also in the offspring.

3.3 Exact Improvement Search

We attempt at improving the best solution found by the GA through an *exact* large neighborhood search, namely a search that is formulated as a suitable Binary Linear Programming problem solved by a state-of-the-art optimization

solver [9]. The search is based on using the effective heuristic RINS (*Relaxation Induced Neighborhood Search* - we refer the reader to [15] for an exhaustive description of it). Specifically, given a partial solution $(\bar{y}, \bar{x})$ of (ROB-BLP-VP) and (y^{LR}, x^{LR}) an optimal solution of a Linear Relaxation (i.e., a solution obtained by removing the integrality requirements on the binary variables), we solve a subproblem of (ROB-BLP-VP) where the value of the j-th component of the vectors (y, x) is fixed according to the following two rules:

$$IF\ (\bar{y}, \bar{x})_j = 0 \wedge (y^{LR}, x^{LR}) \leq \epsilon\ THEN\ (y, x)_j = 0$$

$$IF\ (\bar{y}, \bar{x})_j = 1 \wedge (y^{LR}, x^{LR}) \geq 1 - \epsilon\ THEN\ (y, x)_j = 1$$

The subproblem of (ROB-BLP-VP) subject to such variable fixing is then solved by the state-of-the-art solver, running with a time limit.

4 Preliminary Computational Results

We preliminary assessed the performance of the proposed matheuristic by considering 10 instances that refer to a network made up of 10 nodes to which 50 servers are connected and that are defined for different VNFC features, defined referring to the works [6,8]. To execute the tests, we employed a Windows machine with 2.70 GHz processor and 8 GB of RAM. As optimization solver, we relied on IBM ILOG CPLEX 12.5, which is interfaced through Concert Technology with a C/C++ code. The global time limit imposed to CPLEX to solve (ROB-BLP-VP) is set to 3600 s. The same time limit is set for the matheuristic (denoted here by $MatHeu$), assigning 3000 s to the GA phase and 600 to the improvement phase based on RINS (in which we set $\epsilon = 0.1$). The initial population includes 100 individuals/solutions and, at each iteration, we consider $\beta = 10$ groups from each of which $\gamma = 2$ individuals are chosen.

The results of the computational tests are presented in Table 1, where: *ID* identifies the instance; T^* *(CPLEX)* and T^* *(MatHeu)* are the time (in seconds) that CPLEX and $MatHeu$ needs to find the best solution within the time limit, respectively, whereas $\Delta T^*\%$ is the percentage reduction in time that $MatHeu$ grants to find a solution that is at least as good as the best solution found by CPLEX. Finally, $\Delta P^*\%$ is the reduction in power consumption that the best solution found by $MatHeu$ grants with respect to the best solution found by CPLEX within the time limit.

As highlighted in several works, such as [5,6] even simplified deterministic versions of (ROB-BLP-VP) may prove difficult to solve for state-of-the-art optimization solvers also in the case of instances. We confirm such behaviour in the case of our instances, which highlights the need for fast (heuristic) solution algorithms. On the basis of the results, we can say that $MatHeu$, for all the instances, is able to return a solution that is at least as good as the best solution found by CPLEX within the time limit in 20% less time, on average. Concerning the reduction in consumed power, we can instead notice that $MatHeu$ allows to find better quality solution than CPLEX within the time limit, with a reduction in power consumption that can reach 10% and on average is equal to 7.3%.

Table 1. Preliminary computational results

ID	T^* (CPLEX)	T^* (MatHeu)	$\Delta T^*\%$	$\Delta P^*\%$
I1	3322	2580	22.3	5.4
I2	3194	2742	14.1	6.8
I3	3157	2335	26.0	6.2
I4	3552	2905	18.2	10.2
I5	3513	2536	27.8	6.9
I6	3402	2892	14.9	5.8
I7	3475	2642	23.9	8.6
I8	3362	3041	9.5	9.3
I9	3595	2587	28.0	7.6
I10	3488	2769	20.6	5.5

We consider such results remarkable: as future work, they encourage to refine the solution construction mechanism, better exploiting the specific features of (ROB-BLP-VP) to define the rules adopted to generate the initial population and the offspring solutions by crossover. Furthermore, we intend to investigate also the integration of the GA construction phase with other ad-hoc exact large neighborhood search procedures besides RINS.

References

1. Larsson, C.: 5G Networks - Planning, Design and Optimization. Academic Press, Cambridge (2018)
2. Abdelwahab, S., Hamdaoui, B., Guizani, M., Znati, T.: Network function virtualization in 5G. IEEE Commun. Mag. **54**(4), 84–91 (2016)
3. Herrera, J., Botero, J.: Resource allocation in NFV: a comprehensive survey. IEEE Trans. Netw. Serv. Manage. **13**(3), 518–532 (2016)
4. Baumgartner, A., Bauschert, T., D'Andreagiovanni, F., Reddy, V.S.: Towards robust network slice design under correlated demand uncertainties. In: IEEE International Conference on Communications (ICC), pp. 1–7 (2018)
5. Luizelli, M.C., Bays, L.R., Buriol, L.S., Barcellos, M.P., Gaspary, L.P.: Piecing together the NFV provisioning puzzle: efficient placement and chaining of virtual network functions. In: IFIP/IEEE International Symposium on Integrated Network Management (IM), pp. 98–106 (2015)
6. Marotta, A., D'Andreagiovanni, F., Kassler, A., Zola, E.: On the energy cost of robustness for green virtual network function placement in 5G virtualized infrastructures. Comput. Netw. **125**, 64–75 (2017)
7. Mechtri, M., Ghribi, C., Zeghlache, D.: A scalable algorithm for the placement of service function chains. IEEE Trans. Netw. Serv. Manage. **13**(3), 533–546 (2016)
8. Marotta, A., Zola, E., D'Andreagiovanni, F., Kassler, A.: A fast robust approach for green virtual network functions deployment. J. Netw. Comput. Appl. **95**, 42–53 (2017)

9. Blum, C., Puchinger, J., Raidl, G., Roli, A.: Hybrid metaheuristics in combinatorial optimization: a survey. Appl. Soft. Comput. **11**, 4135–4151 (2011)
10. Ben-Tal, A., El Ghaoui, L., Nemirovski, A.: Robust Optimization. Princeton University Press, Princeton (2009)
11. Bauschert, T., Büsing, C., D'Andreagiovanni, F., Koster, A.M.C.A., Kutschka, M., Steglich, U.: Network planning under demand uncertainty with robust optimization. IEEE Commun. Mag. **52**, 178–185 (2014)
12. Bertsimas, D., Sim, M.: The price of robustness. Oper. Res. **52**(1), 35–53 (2004)
13. IBM ILOG CPLEX. http://www-01.ibm.com/software
14. Goldberg, D.: Genetic Algorithms in Search, Optimization & Machine Learning. Addison-Wesley, Reading (1988)
15. Danna, E., Rothberg, E., Le Pape, C.: Exploring relaxation induced neighborhoods to improve MIP solutions. Math. Program. **102**, 71–90 (2005)