Routing and Resource Assignment Problems in Future 5G Radio Access Networks

Amal Benhamiche
Orange Labs, Châtillon
France
amal.benhamiche@orange.com

Wesley da Silva Coelho
Orange Labs, Châtillon
France
wesley.dasilvacoelho@orange.com

Nancy Perrot
Orange Labs, Châtillon
France
nancy.perrot@orange.com

ABSTRACT

Given a *mobile network* composed of a set of devices, a set of antennas (Base Stations) and a discrete set of radio resources, we define a *domain* as a subset of devices/antennas that communicate via radio transmission links in order to exchange data for a specific service. In this context, we are interested in the Domain Creation (DC) problem. It consists in finding an allocation of radio resources to the transmission links of the network so that different domains, each one related to a specific service (gaming, video streaming, content sharing, etc.), can be implemented simultaneously. Every domain has specific requirements in terms of quality of the transmission links (SINR) and hardware resources dedicated to carrying out the corresponding service. We give an integer linear programming formulation for the problem and propose two classes of valid inequalities to strengthen its linear relaxation. The resulting formulation is used within a branch-and-cut algorithm for the problem. We further propose an efficient heuristic obtained from solving the routing and resource assignment sub-problems separately. We assess the efficiency of our approaches through some experiments on instances of varying size and realistic input data from Orange mobile network.

1 INTRODUCTION

In future 5G networks, mobile User Equipment (UEs) will be able to host functions that give them new abilities such as sharing connectivity, capacity and CPU resources with other UEs, regardless of the ongoing traditional communications. The 5G wireless technology, along with the evolution of mobile users behavior and needs, will make the current scheme of communication (UE to Base Station) no longer optimal in terms of radio resource utilization. The Device-to-Device (D2D) communication mode is one of the new approaches presented as a promising alternative to traditional communication in cellular networks. A D2D communication is defined as a direct communication between two mobile or fix user devices, without traversing the Base Station (BS) [3]. This technology allows to reuse radio resources and to decrease end-to-end latency of local communications. Then, D2D would allow a set of UEs geographically close to each other to establish direct D2D communications, or span multiple links (multi-hop D2D communications), to access a given service (e.g. video streaming or gaming) while ensuring the required service quality.

A *domain* is defined as the set of UEs and BSs that are used to establish mobile communications (D2D or cellular) related to a specific service. The communication is either direct or uses multiple links (D2D or via the BS). Two UEs can then communicate through cellular links, using the BS or D2D links, and both technologies can coexist within the same mobile network. In any case, radio resources should be allocated to every active link involved in a communication, and the SINR (Signal-to-Interference-plus-Noise Ratio) level required by the service should be ensured.

In this context, we consider a mobile network composed of a set of devices (UEs), a set of antennas (BS) and a set of services eligible to D2D communications, with their associated traffic matrices. These traffic matrices are in the form of data volume to be exchanged between pairs of devices. The involved devices can communicate through one or several links, either using D2D or cellular communication (via the BS). A non-negative weight, corresponding to the SINR, is associated with each link. It is a measure of the quality of the communication that could be established using this link. Every service requires a minimal quality threshold in terms of SINR and available resources (hardware capacity for the devices, radio resources for the links) to be successfully established. The *Domain Creation* (DC) problem consists in finding a minimum cost allocation of the radio resources to the network links so that (*i*) every pair (link, resource) belongs to a unique domain, that is, it is used for a single service; (*ii*) the SINR of each pair (link, resource) assigned to a domain is above the quality threshold required by the corresponding service; (*iii*) every demand is routed from its origin to its destination within a domain and; (*iv*) all the required types of hardware capacities (CPU, RAM, battery) in the devices are satisfied.

State-of-the-art

The problem of assigning radio resources to transmission links is studied in [1] and [2] under the denomination of Frequency Assignment Problem (FAP). In [1], the authors give an ILP formulation for the problem and propose a branch-and-cut algorithm to solve it. The work in [2] presents several variants of FAP and discusses existing and original optimization (including exact and heuristic) approaches to solve them. The routing and resource allocation aspects are combined in the so-called *Routing and Wavelength Assignment* (RWA) problem (see [6] and [11] for instance) which arises in Optical Networks. The DC problem differs from the above cited problems in that traffic demands are *unsplittable* (each demand has to be sent along a unique path using D2D or cellular links), several types of capacities on the devices are considered, and radio resources re-use is submitted to distance constraints so as to avoid radio interference. [13] has focused more specifically on the optimization of mobile network resources using D2D technology. As the related optimization problem is relatively difficult to be solved in competitive time, a greedy heuristic has been proposed as an alternative to solve such a problem. This heuristic is divided into two phases, each one responsible for the allocation of uplinks (from UEs to BSs) and downlinks (from BSs to UEs) in order to improve the total throughput and to significantly reduce the interference between classic and D2D communications.

ISBN 978-3-89318-079-0 on OpenProceedings, ISSN 2510-7437

DOI: 10.5441/002/inoc.2019.17

Our contribution

In this work, we formally define the Domain Creation Problem and we propose a node-arc (compact) ILP formulation to model it. We present some strategies to enhance the linear relaxation of the formulation along with two classes of valid inequalities. Our results are embedded within a branch-and-cut algorithm to solve the problem. We further propose a two-phase heuristic, obtained by decomposing the problem into a routing and a radio resource allocation subproblems. To solve the routing subproblem, we propose two methods : first, a LP-based heuristic from the linear relaxation of the compact formulation, second, a non-compact formulation obtained by generating a subset of relevant paths. Then, the allocation subproblem is transformed into a vertex coloring problem that is solved heuristically by an improved greedy algorithm. This greedy algorithm is compared to a dual bound given by the exact solution of the associated Max-Clique Problem. Numerical experiments are made on instances generated thanks to realistic parameters of Orange mobile networks.

Our paper is organized as follows. We introduce the DC problem in Section 2 and give an ILP formulation along with two classes of valid inequalities. An overview of our branch-and-cut algorithm is given in Section 3 with a brief description of the separation routines used to generate the cuts and some numerical results for small instances. The outline of our two-phase heuristic is detailed in Section 4. Finally, we give some concluding remarks in Section 5 and discuss future works on extensions of the problem.

2 THE DOMAIN CREATION PROBLEM

2.1 Problem definition

The network is represented by a directed graph $G = (V \cup U, A)$, where V is the set of nodes associated with devices, U the set of Base Station nodes, and A the set of arcs. We denote by $\delta^+(u)$ (resp. $\delta^-(u)$) the subset of arcs going from (resp. to) node u. Every node $u \in V$ has an associated weight vector $c^u = \{c_1^u, \dots, c_b^u\}$, where $c_i^u \geqslant 0$ is the capacity available at node u for the physical resource $i \in C^d$. Every arc $e \in A$ has a weight denoted $SINR_e$ that expresses a measure on the quality of the transmission link represented by e. For every pair of arcs $e, f \in A$ we denote by $d(e, f)$ the *distance* between e and f. This value corresponds to the minimum distance between the opposite ends of the given pair of links, that is, the origin of one and the destination of the other. Namely, two arcs e and f are said to be *close* if $d(e, f) \leqslant D$, where D is a minimum acceptable distance. Let $R = \{1, \dots, r\}$ be the set of available radio resources. An arc e of A is said to be *active* if a radio resource $r \in R$ is assigned to it. A resource $r \in R$ can be assigned to two different arcs $e, f \in A$ only if $d(e, f) > D$. Indeed, due to interference constraints, two communications cannot be established on e and f simultaneously using the same resource $r \in R$ if e is close to f. We denote by K the set of traffic demands to be routed and S the set of service types. Every demand $k \in K$ is defined by an origin node $o^k \in V$, a destination node $d^k \in V$ and a requested service s_k. Moreover, every $k \in K$ has an associated cost $\mu_{s_k}^e$ to use arc $e \in A$ and a traffic vector $a^{s_k} = (a_1^{s_k}, \dots, a_b^{s_k})$ where the element $a_m^{s_k} \geqslant 0$ denotes the quantity of physical resource type m from the set C^d of all resources types (e.g. CPU, RAM, storage) needed to process the service s_k requested by k. Finally, we denote by β_k the quality threshold needed by k to access the required service s_k.

The Domain Creation (DC) Problem consists in finding a minimum cost allocation of the radio resources in R to the active arcs of G so as to provide a feasible routing path for each demand. In particular, a routing $p^k = \{(o^k, u), \dots, (v, d^k)\}$ for a demand k is said to be feasible if

- all the arcs of p^k have a SINR value above the quality threshold β_k required by the demand k and,
- all the nodes in p^k have enough capacity to satisfy the resource requirements of k.

An instance of the problem, with fives devices and one BS, is illustrated in Figure 1. For sake of clarity, each pair of arcs between two nodes is represented by an edge. The edges representing D2D links are shown in solid lines while edges representing device to BS links are in dashed lines. Two different services, namely *gaming* and *video streaming*, and three demands per service are to be delivered:{(u_1,u_3), (u_2,u_5), (u_4, u_2)} and {(u_4, u_3), (u_3, u_1), (u_5, u_2)}, assuming that twelve radio resources $\{r_1, \dots, r_{12}\}$, are available. A feasible solution is represented in Figure 2. The figure on the left side represents the *Gaming* domain, where all three demands are satisfied through D2D links. The figure on the right side is the *Video streaming* domain, where just one demand uses the BS. Note that, using the legacy approach, all demands must pass through the BS, using one uplink (from UE origin to BS) and one downlink (from the BS to UE destination) for each demand. Since all active links share at least one arc extremity (BS), the solution would require all the available resources (a different one for each active link) to avoid interference. Thus, using D2D communications allow here to save 50% of radio resources compared to a "fully cellular" solution.

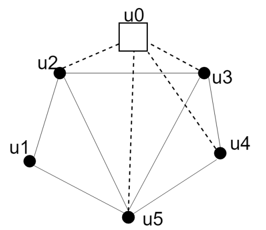

Figure 1: Example of a network with 6 nodes and 6 demands to be delivered for 2 different services

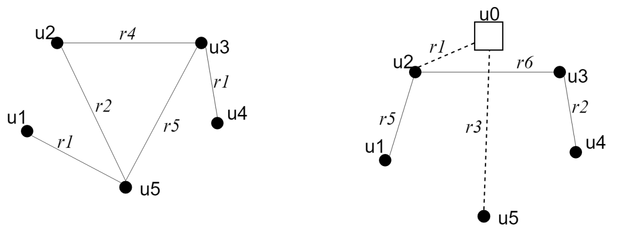

Figure 2: Feasible solution: active links and associated radio resource for *Gaming* domain (left) and *Video streaming* domain (right)

2.2 ILP formulation

In this section, we propose a compact ILP formulation for the DC problem followed by some valid inequalities to be used in a branch-and-cut algorithm.

2.2.1 Notations and formulation. The three types of binary variables are:

• x_{er}^k, $e \in A, k \in K, r \in R$ that takes the value 1 if the arc e is used

by the demand k and assigned with the resource r, 0 otherwise.
• y_i^k, $i \in V \cup U$, $k \in K$, that takes the value 1 if the node i is used by the demand k, 0 otherwise.
• z^r, $r \in R$ that takes the value 1 if the resource $r \in R$ is assigned with at least one arc, 0 otherwise.

Then, the DC problem can be formulated as:

$$\min \sum_{k \in K} \sum_{e \in A} \sum_{r \in R} \mu_e^{s_k} x_{er}^k + \sum_{r \in R} \psi z^r \tag{1}$$

s.t.

$$\sum_{e \in \delta^-(u)} \sum_{r \in R} x_{er}^k - \sum_{e \in \delta^+(u)} \sum_{r \in R} x_{er}^k = \begin{cases} 1, & \text{if } u = d_k, \\ -1, & \text{if } u = o_k, \\ 0, & \text{otherwise,} \end{cases}$$

$$\forall k \in K, \forall u \in U \cup V, \tag{2}$$

$$x_{er}^k \beta_k \leq SINR_e \qquad \forall\, k \in K, \forall\, e \in A, \forall\, r \in R, \tag{3}$$

$$\sum_{k \in K \setminus T(i)} y_i^k a_m^{s_k} \leq c_m^i \qquad \forall\, i \in V, \forall\, m \in C^d, \tag{4}$$

$$2x_{er}^k - y_i^k - y_j^k \leq 0 \qquad \forall k \in K, \forall r \in R \forall (i,j) = e \in A, \tag{5}$$

$$x_{er}^k + x_{fr}^{k'} \leq z^r \qquad \forall r \in R, \forall k, k' \in K, \forall e \in A, \forall f \in D(e), \tag{6}$$

$$x_{er}^k \in \{0,1\} \qquad \forall k \in K, \forall e \in A, \forall r \in R, \tag{7}$$

$$y_i^k \in \{0,1\} \qquad \forall k \in K, \forall i \in V, \tag{8}$$

$$z^r \in \{0,1\} \qquad \forall r \in R. \tag{9}$$

This formulation has a polynomial number of variables and constraints. The objective (1) is to minimize the total costs composed of non-negative routing and radio resource utilization costs. The first set of inequalities (2) are the flow conservation constraints. They ensure that each demand is routed along a unique path between its origin node and its destination node. Note that such a routing path can either span arcs corresponding to D2D links or use a node of U corresponding to some BS. Inequalities (3) guarantee that a demand for a given service is routed along edges whose SINR satisfies the quality threshold required by this service. (4) express the capacity constraints in every node for the different types of hardware resources, with $T(i)$ being the set of demands $k \in K$ that have $i \in V$ as origin or destination. These capacity constraints are needed only on intermediate nodes, that is, the nodes that are not the origin nor the destination of a given demand $k \in K$. Inequalities (5) are linking constraints and inequalities (6) guarantee that the same radio resource is not assigned to different edges unless they are distant enough, where $D(e)$ is a set of arcs that are close to a given arc $e \in A$. Note that, as the objective is to minimize the resources, in any optimal solution one radio resource at most will be allocated to the same arc for a given demand. Finally, (7)-(9) are the integrity constraints.

The ILP formulation (1)-(9) can be strengthened by replacing inequalities (3) by:

$$x_{er}^k \leqslant \lfloor \frac{SINR_e}{\beta_k} \rfloor, \forall k \in K, e \in A, r \in R. \tag{10}$$

2.2.2 Symmetry. Due to the inherent symmetry of this problem, there is possibly a large number of feasible solutions. The breaking symmetry constraints (11) are inspired by classical inequalities in combinatorial optimization (see [8]), and can help in reducing the number of symmetric solutions in the formulation.

$$z^r \geqslant z^{r+1}, \qquad \forall 1 \leqslant r \leqslant |R| - 1, \tag{11}$$

(11) allow to assign the radio resources in an ordered way, forbidding to use a resource $r+1$ if r is available. Note that a vector $(\overline{x}, \overline{y}, \overline{z}) \in \{(x,y,z) \in \{0,1\}^{(m \times |R|+n) \times |K|+|R|} : (x,y,z)$ satisfies (2)–(11)$\}$ is clearly a feasible solution to the DC problem.

2.2.3 Valid inequalities. We introduce now two more classes of inequalities valid for the DC problem.

(i) Clique-based inequalities:
Given an instance of DC problem, we define the *conflict graph* associated with a node $u \in V$ as follows. For each capacity type $m \in C^d$, let $\mathcal{H}_u^m = (V_u^m, E_u^m)$ be the undirected graph obtained from the set of demands K as follows. A node v_k in V_u^m is associated with every demand $k \in K$ and there exists an edge $v_k v_l \in E_u^m$ between two nodes v_k, v_l of V_u^m if $a_m^{s_k} + a_m^{s_l} > c_m^u$. In other words, an edge in $\mathcal{H}_u^m$ exists if two demands cannot be packed together in the node u due to the lack of capacity for the resource type m. Consequently, a clique in the graph $\mathcal{H}_u^m$ corresponds to a set of demands that cannot use simultaneously the node u. Hence, we denote by $C_{\mathcal{H}}$ the set of cliques in the graph $\mathcal{H}_u^m$.

Proposition 2.1. *Let u be a node of V and $m \in C^d$ be a physical resource type. Then the following inequalities*

$$\sum_{k \in C} y_u^k \leqslant 1, \forall C \in C_{\mathcal{H}} \tag{12}$$

are valid for the DC problem.

Proof. Let $\widetilde{C}$ be a clique in $\mathcal{H}_u^m$. It is clear that if two demands k_1, k_2 from clique $\widetilde{C}$ use the resource m of node u, the capacity constraint (4) for the resource m will be violated. In other words, each edge $v_{k_i} v_{k_j}$ of the clique $\widetilde{C}$ represents an infeasible packing of the demands k_i, k_j in the node u. □

(ii) Strengthened neighborhood inequalities:
The second family of valid inequalities strengthen constraints (6). They are obtained by considering the *interference* graph $\mathcal{N} = (V_N, E_N)$ defined as follows. Every node $u \in V_N$ corresponds to an arc in A and two nodes u_e, u_f of V_N (associated respectively with the arcs e and f from A) are interconnected by an edge if e and f are close enough from each other (ie if $d(e,f) \leqslant D$). Consequently, a clique in the graph $\mathcal{N}$ corresponds to a subset of arcs in A that are pairwise close, and cannot receive the same radio resource due to interference constraints. Likewise in the conflict graph defined before, we denote by C_N the set of cliques in the graph $\mathcal{N}$.

Proposition 2.2. *The following inequalities*

$$\sum_{k \in K} \sum_{e \in C} x_{er}^k \leqslant z^r, \forall r \in R, C \subseteq C_N \tag{13}$$

are valid for the DC problem.

Proof. Let $\widetilde{C}$ be a clique in $\mathcal{N}$ and u_e, u_f two nodes of V_N that belong to clique $\widetilde{C}$. Clearly, if e and f are allocated the same resource $r \in R$ in a solution, then it cannot be feasible for the DC problem. □

Note that similar inequalities are used in [1] and [2] for the Frequency Assignment Problem.

3 BRANCH-AND-CUT ALGORITHM

3.1 Overview of the algorithm

We have developed a branch-and-cut algorithm for the DC problem based on the results presented in Section 2. The algorithm has been implemented in C++ using CPLEX 12.6 as a LP solver, with presolve heuristics and internal cuts being disabled. We have tested our approach

- first by solving formulation (1)-(9) along with the strengthened SINR inequalities (10) and symmetry breaking (11) inequalities and
- by further using the clique-based (12) and strenghthened neighborhood (13) valid inequalities, in addition to the formulation (1)-(9).

We have used two heuristic procedures to generate dynamically inequalities (12) and (13). Both separation routines rely on a greedy algorithm introduced in [10] for the Independent Set problem, that finds a clique in the conflict graph (respectively the interference graph) with appropriate weights on the nodes. We then add the corresponding violated clique-based (respectively strenghthened neighborhood) inequalities, if any, to the current LP. Both classes of valid inequalities are separated throughout the branch-and-cut tree and several inequalities may be added at each iteration of the algorithm.

3.2 Computational results

We present below some early experiments obtained for a set of small instances containing 5 to 15 nodes and up to 7 demands. For these tests, each scenario contains only 1 antenna and 2 service domains. For each instance, realistic data was provided by Orange, including the network topologies and SINR values. Table 1 shows the impact of inequalities (10) and (11) on the initial model (formulation (1)-(9)). In the first column, the name of each instance refers to the number of users (U#) and demands (D#). The next four columns show the computational time (in seconds) for the initial formulation, then when adding symmetry breaking constraints (11), strenghthened SINR constraints (10) and both constraints, respectively. We can notice that using constraints (11) allows to obtain the best execution time for two out of five instances tested ($U5\_D8$ and $U10\_D5$) while the instances $U5\_D6$ and $U10\_D7$ have the lower runtime when applying constraints (10) and (11), combined. For instance $U15\_D7$, the optimal solution could not be found after 3 hours of execution. We have

Table 1: Runtime comparison (in seconds) with strengthening constraints

Instances	Initial	Symmetry	SINR	Symmetry & SINR
U5_D2	0,05	0,06	0,05	0,06
U5_D6	187,18	11,55	77,18	**10,71**
U5_D8	600,56	**45,16**	90,70	72,43
U10_D5	1677,21	**290,33**	697,49	435,08
U10_D7	8746.32	3569.18	4453.58	**2967.45**
U15_D7	10800*	10800*	10800*	10800*

tested the impact of using our valid inequalities and compared the results to CPLEX branch-and-bound for the strengthened model (formulation (1)-(2) + (4)-(11)). Table 2 shows the results obtained on five instances (same as in Table 1) when (i) no additional cuts are used, (ii) using strenghthened neighborhood inequalities (13) in addition to formulation (1)-(9), (iii) using clique-based inequalities (12) in addition to strengthened model and (iv) both cuts are used in the branch-and-cut. We can observe that the gap at root node is substantially reduced when adding cuts (the gap value decreases from 59.65% to 23.58% for instance $U5\_D8$ when using strenghthened neighborhood cuts) and so for the size of the branch-and-cut tree (for instance $U10\_D5$, we range from 1327 nodes without cuts to 84 nodes when both cuts are used). Overall, the strenghthened neighborhood cuts are more efficient in reinforcing the strengthened model.

4 TWO-PHASE HEURISTIC

Since solving the initial formulation has impractical runtime even for small instances, we propose a solving method based on a decomposition of DC problem into two subproblems: the *routing subproblem* and the *resource allocation subproblem*, that are to be solved separately. The objective of the first subproblem is to find an elementary path for each demand while minimizing the total link utilization costs. Then, the second subproblem provides a resource allocation to each active link obtained from the routing subproblem solution.

4.1 Routing subproblem

We propose two formulations for the routing subproblem: a compact formulation obtained by relaxing the resource assignment constraints (6) from (1)-(9), and a path reformulation.

4.1.1 Compact formulation. This formulation is the compact formulation obtained by relaxing the resource allocation constraints (6) from the formulation (1)-(9). The returned solution is a set of elementary paths for each request, respecting the capacities of the nodes along the paths. Two kinds of binary variables remains in the formulation : x_e^k that takes value 1 if the link $e \in A$ is used by the request $k \in K$ and the variables y_k^i. The objective is to minimize the total cost of active links:

$$min \sum_{k \in K} \sum_{e \in E} \mu_e^{s_k} x_e^k \tag{14}$$

Solving approach: The linear relaxation of this formulation is strengthened replacing **(5)** by:

$$x_e^k - y_i^k \leq 0 \qquad \forall\, k \in K, \forall\, (i,j) = e \in E, \tag{15}$$

$$x_e^k - y_j^k \leq 0 \qquad \forall\, k \in K, \forall\, (i,j) = e \in E. \tag{16}$$

and adding the following inequalities:

$$y_i^k (a_m^{s_k} - c_m^i) \leq 0 \qquad \forall\, i \in V, \forall\, m \in C^d, \forall\, k \in K \backslash T(i). \tag{17}$$

The linear relaxation of this strengthened subproblem is solved using CPLEX. Then, a heuristic procedure is used to get a feasible integer solution. This approach is summarized in Algorithm 1. First, the linear relaxation is solved to optimality. Then, variables having an integer optimal value are fixed by updating the right hand side of the constraints (step 5). Finally, this residual ILP formulation is solved to optimality using CPLEX, giving rise to an integer solution for the whole problem. Step 11 is the rounding procedure on solution found by step 1. This heuristic is particularly efficient for this problem since most of the optimal variable values of the linear relaxation are integer.

4.1.2 Path formulation. The second formulation is a path formulation. We assume that all feasible paths P_k have been previously generated for each demand k. Thus, we define new binary variables x_p^k that take value 1 if the path $p \in P_k$ is used by

Table 2: Solution quality comparison between models with and without additional cuts

	Strengthened Formulation			Strengthened + Neighborhood cuts			Strengthened + Clique-based cuts			Strengthened + both cuts		
Instances	root gap (%)	runtime (s)	tree size	root gap (%)	runtime (s)	tree size	root gap (%)	runtime (s)	tree size	root gap (%)	runtime (s)	tree size
U5_D2	**0.00**	0.05	**1**	**0.00**	0.06	**1**	**0.00**	0.06	**1**	**0.00**	0.05	**1**
U5_D6	10.00	11.49	7	**10.00**	11.67	8	**10.00**	12.33	**2**	**10.00**	13.31	**2**
U5_D8	59.65	154.65	470	**23.58**	130.69	**291**	33.71	389.01	1225	32.00	150	599
U10_D5	57.89	677.21	1327	**33.09**	1370.90	190	48.23	1262.83	**91**	33.09	2265.83	**84**
U10_D7	62.26	3236.61	1850	**50.04**	8654.21	2040	62.26	3251.99	1850	50.04	9648.12	2040
*U15_D7**	77.23	10800	58	**72.37**	10800	8	77.23	10800	70	72.37	10800	22

Algorithm 1 LP-based Heuristic for the Routing subproblem

Solve LP (G, K)
Let *FractionalDemands* be the set of demands with associated optimal fractional variables
for each demand k **do**
  if all associated variable have integer value **then**
    update the LP by decreasing used capacities on vertices traversed by the associated paths
  else
    FractionalDemands : *FractionalDemands* $\cup \{k\}$
  end if
end for
Solve ILP (G,FractionalDemands)
Update solution
Return solution found

the demand $k \in K$, 0 otherwise. The objective now is to minimize the sum of the active path weights:

$$min \sum_{k \in K} \sum_{p \in P_k} \mu_p^{s_k} x_p^k \tag{18}$$

where $\mu_p^{s_k}$ is the cost of path $p \in P_k$, defined as the sum of the weights on the arcs of p. Let α_p^e, respectively γ_p^i, be an indicator parameter with value 1 if the arc e, respectively the node i, is in the path p. The constraints of the path formulation are:

$$\sum_{p \in P_k} x_p^k = 1 \qquad \forall\, k \in K, \tag{19}$$

$$\alpha_p^e x_p^k \beta_{s_k} \leq sinr_e \qquad \forall\, k \in K, e \in A, \forall\, p \in P_k, \tag{20}$$

$$\sum_{k \in K \setminus T(i)} \sum_{p \in P_k} \gamma_p^i x_p^k a_m^{s_k} \leq c_m^i \qquad \forall\, i \in V, \forall\, m \in C^d, \tag{21}$$

$$x_p^k \in \{0, 1\} \qquad \forall\, k \in K, \forall\, p \in P_k. \tag{22}$$

Constraints (19) assure that each demand uses one path. Constraints (20) are the SINR constraints, and (21) are the capacity constraints.

Solving approach: This formulation, restricted to a subset of paths, is solved to optimality using CPLEX. The subsets of paths are generated thanks to the algorithm proposed by Yen [12]. This algorithm returns the K-shortest loopless paths for a graph with non-negative edge costs. It uses a shortest path algorithm as an intermediary to construct the whole solution. In our context, we use Dijkstra's algorithm [5], which has a good performance and a polynomial complexity. Using Yen's algorithm gives a guarantee to generate all the feasible paths due to the characteristics of our use case: for each demand, once a path uses the base station we are sure that all the feasible D2D paths have already been generated, then the path generation is stopped. This feature is due to greater weights on BS arcs. The difference between the weights of the BS links and the D2D ones plays an important role in the construction of the paths subsets: the greater this difference is, the longer it could be to obtain all the paths that use only D2D communication. Finally, to reduce the size of the formulation, a pre-processing on the SINR constraints is operated before solving both routing formulations: only the valid SINR links constraints are kept in the formulation.

4.2 Resource Allocation subproblem

The resource allocation subproblem consists in allocating the radio resources to each active link provided by the solution of the first subproblem.

4.2.1 Solving approach. Let $G^a(V^a, E^a)$ be a graph where each vertex in the set V^a represents an active link, that is, a link used by at least one path of the routing problem solution (steps 2-5 in Algorithm 2). An edge $e \in E^a$ is associated with a pair of vertices if the corresponding active links cannot share the same radio resource due to interference (steps 6-12). The objective is to assign a minimum number of colors (resources) to the vertices of the graph G^a, in such a way there is no adjacent vertices with the same color (step 14). This falls into a classical Vertex Coloring Problem [7]. The proposed heuristic has as input the initial graph

Algorithm 2 Resource Allocation subproblem

1: Set V^a and E^a to empty sets
2: **for** each activated link $e \in A$ **do**
3:   Set Vertex $auxVertex.id \longleftarrow e.id$
4:   Set $V^a : V^a \cup auxVertex$
5: **end for**
6: **for** each pair $(u, v) \in V^a$ **do**
7:   **if** $dist(u, v) \leq criteria$ **then**
8:     Set Link $auxLink.extremity1 \longleftarrow u$
9:     Set Link $auxLink.extremity2 \longleftarrow v$
10:     Set $E^a : E^a \cup auxLink$
11:   **end if**
12: **end for**
13: Set graph $G^a(V^a, E^a)$
14: Coloring (G^a)
15: Return Resource Assignment

of the problem, the values of the routing solutions and a criteria value which represents the minimum distance for a pair of links to have the same resource. This value was fixed to $D = 100$ meters so as to be representative of realistic use cases. Hence, for *dist*() method, we calculate the distance between the opposite ends of the pair of links, that is, the origin of one with the destination of the other. The method returns *zero* for adjacent links. Finally, to find the coloring of the graph G^a (that is, allocate the resources to each active link) we use the *Coloring*() method which is the implementation of a classical greedy algorithm [9] for Vertex Coloring Problem. It is well known that the performance of this greedy algorithm is sensitive to the order of choice of the next vertex to be colored. For this reason, we randomly generate a large number of orders and choose the one that returns the minimum amount of colors. A lower bound value for the optimal solution is given by solving the associated Max-clique problem. For this purpose we use the method proposed by [4], which is an exact approach using parallel programming.

4.3 Experiments and comparative results

The experiment conditions are the same as in section 3. We generated new instances, with 6 service domains available and the total amount of UEs is evenly divided between 7 core cells with one antenna installed on each one.

4.3.1 Routing subproblem. The numerical tests of the routing subproblem are summarized in table 3. The first column of the Compact Formulation part is the percentage of *active* variables that *are not* integers in the optimal LP solution. We note that it is always less than 6%. The second column is the gap between the solution found by the relaxation and the final solution found by the LP-based heuristic. The average gap is 2.30% with standard deviation equals to 3.85%. Finally, the third column is the total run time in seconds (pre-processing, relaxed solution and LP-based heuristic). In the second part of table 3, the pre-processing and solver run times of the path formulation are presented. The average number of paths generated in pre-processing is equal to 5.26 for each demand (with standard deviation equals to 2.39). Most of the runtime of the compact formulation based approach was spent in solving the linear relaxation of the problem. Given the very low number of fractional variables in the relaxed solution, the last IP on the residual formulation is quickly solved. It is also worth noticing that the pre-processing in path formulation is the most important step, since it is responsible for most of the solving time for all instances. However, its performance is relatively better than for the compact formulation, being on average 1.30 times faster. It is important to mention that the gap between the two formulations was always less than 0.50%.

Table 3: Routing subproblem: numerical tests

	Compact formulation			Path Formulation	
Instances	Act Frac Var (%)	Gap (%)	Total Runtime (s)	Pre processing (s)	Solver (s)
U700_D175	0.05	0.03	9.16	4.81	0.02
U700_D350	1.92	1.1	13.92	9.5	0.02
U700_D525	1.84	13	21.28	14.26	0.08
U700_D700	0.87	0.04	26.72	19.22	0.13
U1400_D350	5.22	0.02	46.83	27.48	0.8
U1400_D700	3.37	3.6	77.7	55.68	0.26
U1400_D1050	3.13	3.05	152.31	82.82	0.42
U1400_D1400	1.79	0.49	143.78	110.81	0.60
U2100_D525	5.56	0.02	143.28	81.28	0.23
U2100_D1050	3.54	0.09	257.18	163.39	0.65
U2100_D1575	2.15	3.86	242.84	230	0.8

Table 4: Resource Allocation subproblem: numerical tests

Instances	Pre-processing (s)	Greedy (s)	Gap (%)
U700_D175	0.13	0.89	0
U700_D350	0.53	1.81	0
U700_D525	1.15	2.75	0
U700_D700	2.12	3.89	0
U1400_D350	0.54	1.74	0
U1400_D700	2.34	3.76	0
U1400_D1050	4.97	5.95	0
U1400_D1400	7.06	6.97	0
U2100_D525	1.32	2.07	0
U2100_D1050	4.45	5.25	0
U2100_D1575	11.21	9.51	0

4.3.2 Resource allocation subproblem. Table 4 shows the results for the resource allocation subproblem, where the *Pre-processing* column represents the time needed to transform the solution from the previous subproblem into a classic graph and find its max-clique. We can observe that even though they are extremely large graphs, the time needed to find the final solution is relatively short. This is due to the characteristics of the topology constructed from the assumptions and hypotheses previously presented. Generated graphs have an average density equals to 64.05% - standard deviation equals to 1.98%. Another important result emphases is that in all cases, we have found the optimal solution, proven by the lower bound value previously calculated by the exact max-clique algorithm (last column).

5 CONCLUDING REMARKS

In this paper, we have studied the Domain Creation problem, that is a routing and resource assignment problem arising in future 5G networks. We have proposed two algorithms: exact and heuristic, to solve it. The exact approach is based on a node-arc ILP formulation enhanced by two families of valid inequalities that are used within a branch-and-cut framework. The preliminary results show a significant impact of the cuts in strenghthening the LP relaxation and reducing the computation time. We expect that adding further classes of cuts and performing an analysis to find out the specificities of difficult instances (regardless of their size) will allow to solve even larger instances. A natural question would be to consider a non-compact formulation, based on path variables and propose a column generation based algorithm to solve it. On an other hand, our experiments show that the heuristic approach performs well, even on large instances with up to 2100 devices and 1500 service requests on 7-cell networks. It would be interesting and most probably very powerful to use it as a primal heuristic to boost efficiency of an exact algorithm. On a practical note, a tough but interesting extension is to include users mobility or temporal aspect in radio resource assignment.

ACKNOWLEDGMENTS

This work is supported by the french Agence Nationale de la Recherche (ANR), Project MAESTRO-5G ANR-18-CE25-0012.

REFERENCES

[1] K. I. Aardal, A. Hipolito, S. P. M. van Hoesel, and B. Jansen. 1996. A branch-and-cut algorithm for the frequency assignment problem. *Research Memorandum 96/011, Maastricht University* (1996).

[2] K. I. Aardal, S. P. M. van Hoesel, A. M. C. A. Koster, C. Mannino, and A. Sassano. 2007. Models and solution techniques for frequency assignment problems. *Annals of Operations Research* 153, 1 (01 Sep 2007), 79–129.

[3] Arash Asadi, Qing Wang, and Vincenzo Mancuso. 2014. A Survey on Device-to-Device Communication in Cellular Networks. *IEEE Communications Surveys and Tutorials* 16, 4 (2014), 1801–1819.

[4] Matjaz Depolli, Janez Konc, Kati Rozman, Roman Trobec, and Dusanka Janezic. 2013. Exact parallel maximum clique algorithm for general and protein graphs. *Journal of chemical information and modeling* 53, 9 (2013), 2217–2228.

[5] Edsger W Dijkstra. 1959. A note on two problems in connexion with graphs. *Numerische mathematik* 1, 1 (1959), 269–271.

[6] B. Jaumard, C. Meyer, and B. Thiongane. 2006. *ILP Formulations for the Routing and Wavelength Assignment Problem: Symmetric Systems.* Springer US, 637–677.

[7] Enrico Malaguti and Paolo Toth. 2010. A survey on vertex coloring problems. *International transactions in operational research* 17, 1 (2010), 1–34.

[8] François Margot. 2010. *Symmetry in Integer Linear Programming.* 647–686.

[9] David W Matula, George Marble, and Joel D Isaacson. 1972. Graph coloring algorithms. (1972), 109–122.

[10] G. L. Nemhauser and G. Sigismondi. 1992. A Strong Cutting Plane/Branch-and-Bound Algorithm for Node Packing. *Journal of the Operational Research Society* 43, 5 (1992), 443–457.

[11] A. E. Ozdaglar and D. P. Bertsekas. 2003. Routing and Wavelength Assignment in Optical Networks. *IEEE/ACM Trans. Netw.* 11, 2 (April 2003), 259–272.

[12] Jin Y Yen. 1970. An algorithm for finding shortest routes from all source nodes to a given destination in general networks. *Quart. Appl. Math.* 27, 4 (1970), 526–530.

[13] M. Zulhasnine, C. Huang, and A. Srinivasan. 2010. Efficient resource allocation for device-to-device communication underlaying LTE network. *2010 IEEE 6th International conference on wireless and mobile computing, networking and communications* (2010), 368–375.