Bi-dimensional Assignment in 5G Periodic Scheduling

Giulia Ansuini, Antonio Frangioni, Laura Galli, Giovanni Nardini, Giovanni Stea

Abstract We consider a scheduling application in 5G cellular networks, where base stations serve periodic tasks by allocating conflict-free portions of the available spectrum, in order to meet their traffic demand. The problem has a combinatorial structure featuring bi-dimensional periodic allocations of resources. We consider four variants of the problem, characterized by different degrees of freedom. Two types of formulations are presented and tested on realistic data, using a general-purpose solver.
Knapsack Problems invited session

1 Introduction

5G cellular networks provide ubiquitous wireless access on licensed spectrum, with very low latencies and high reliability, thus being a viable solution for real-time applications, such as vehicular communications. The Cellular Vehicular-to-everything (C-V2X) standard for New Radio 5G networks allows vehicles to request exclusive access to some spectrum resources, which they can use for inter-vehicle communications. Resource allocation is done centrally by the base station, which is in charge of a (possibly large) coverage area, and needs to fulfill several such requests simultaneously. The base station allocates spectrum resources in both time and frequency. On every Transmission Time Interval (TTI), in the order of 1 millisecond or less,

G. Ansuini
Dipartimento di Matematica, Università di Pisa, Largo B. Pontecorvo 5, 56127 Pisa, Italy.

A. Frangioni, L. Galli
Dipartimento di Informatica, Università di Pisa, Largo B. Pontecorvo 3, 56127 Pisa, Italy. e-mail: antonio.frangioni@unipi.it, laura.galli@unipi.it

G. Nardini, G. Stea
Dipartimento di Ingegneria dell'Informazione, Università di Pisa, Largo Lucio Lazzarino 1, 56122 Pisa, Italy. e-mail: giovanni.nardini@unipi.it, giovanni.stea@unipi.it

the base station can allocate an *airframe*, i.e., a vector of several tens of *resource blocks* (RBs). Each RB can be allocated to only one requesting entity at a time. The transmission overhead associated with serving a single request is non negligible, and – quite often – the communication needs of a vehicle are long-term (imagine a vehicle transmitting to another the live video acquired from its camera). For this reason, the C-V2X standard allows vehicles to request *periodic* allocation of resources. This forces the base station to run complex algorithms to compose requests coming from different vehicles.

In such 5G Periodic Scheduling, tasks have a *period* π, expressed in number of TTIs, and a *demand* w, expressed as a number of RBs. A base station serves tasks requested by vehicles by allocating w *contiguous* RBs in a TTI, and on every π-th subsequent TTI thereafter. Once the sub-vector is assigned to a task, it will be used again by the same task in each TTI that it is aired (depending on its period). In other words, the assignment of a sub-vector of RBs to a task periodically reserves the same portion of bandwidth to the same task. We recall that an RB can only be assigned to one task in each TTI – the assignment of a RB to more than one task in the same TTI is called a "conflict".

The basic combinatorial problem consists of finding a conflict-free RB assignment (aka RB *schedule*) for a given set of "new" tasks to be served, taking also into account the resources previously assigned to a set of "old" tasks that are already in place.

We consider four variants of the basic problem by combining two degrees of freedom: (*i*) old tasks are either *fixed* or *movable*; (*ii*) new tasks are either *forced* to be scheduled or their scheduling is *optional*. These variants are solved in sequence, starting from the most rigid one, and loosening it up if the current model turns out to be infeasible. In particular, in the first attempt the old tasks are fixed and all the new ones are forced to be scheduled. If this has no solution, we then consider the old tasks to be movable in order to place all the new tasks. Finally, the last two variants are always feasible, because the scheduling of new tasks is optional (while the old ones are either fixed or movable).

We propose two types of formulations, called "Conflict-Based" and "Matrix-Based", that are solved (for each variant) using the general-purpose MILP solver `CPLEX`. The formulations are tested on several instances corresponding to realistic 5G settings. Our results show that the sequential approach is generally more efficient than immediately solving the most flexible variant (i.e., Movable-Optional), and even large instances can often be solved in fairly short computing times.

2 System Model

We represent the airframe as a vector of RBs of size M. Let $N = \{1, \ldots, n\}$ be the set of new periodic tasks to be scheduled. Each task $i \in N$ corresponds to an ordered pair (π_i, w_i), where π_i is the task period, and w_i is the number of RBs it demands. The RBs assigned to a task must be contiguous elements of the airframe, i.e., a sub-vector of size w_i. The assignment of a sub-vector to a task means reserving the same sub-vector to the task in each TTI it is aired (according to its period). Given a task set N, the hyperperiod H is defined as the least common multiple of the task periods: $H = \mathrm{lcm}_{i \in N}(\pi_i)$. The hyperperiod corresponds to the minimum number of TTI after which the schedule is repeated.

The aim of Periodic Scheduling is to find a RB *schedule* for the task set N in the corresponding hyperperiod. A RB schedule S consists of n ordered pairs (h_i, t_i), one for each task $i \in N$:

- $h_i \in \{1, \ldots, M - w_i + 1\}$ is the position of the *first* RB in the sub-vector $\{h_i, \ldots, M - w_i + 1\}$ of size w_i assigned to task i;
- $t_i \in \{0, \ldots, \pi_i - 1\}$ is the *first* TTI in which task i is aired within the hyperperiod.

Note that given a schedule S, all the (periodic) RB allocations within the hyperperiod are uniquely defined for all $i \in N$: $t_i,\ t_i + \pi_i,\ t_i + 2\pi_i,\ \ldots,\ t_i + \left(\frac{H}{\pi_i} - 1\right)\pi_i$.

Each RB can be assigned to only one task in any TTI, hence the challenge is to avoid simultaneous overlap in time and space, i.e., a conflict. Two tasks $i, j \in N$ that are first aired in TTIs t, t', respectively, will overlap in time if and only if there exist $n_i, n_j \in \mathbb{N}$ such that $t + n_i\pi_i = t' + n_j\pi_j$. This is a Diophantine equation, where n_i, n_j are integer unknowns. The equation has a solution if and only if $t' - t \equiv 0\ (mod\ \mathrm{gcd}(\pi_i, \pi_j))$. Let h, h' be the first RB positions in the airframe for tasks i and j, respectively. To check space (i.e., frequency) overlap one can just consider the intersection between the corresponding sub-vectors: $[h, h + w_i - 1] \cap [h', h' + w_j - 1] \neq \emptyset$.

Feasibility requires that any two tasks can overlap in at most one dimension. Indeed, if the overlap is both in time and space, then there exists (at least) one RB that is assigned to different tasks at the same time, generating a conflict. A schedule S is *feasible* for a set N of tasks, if all demands are satisfied without conflicts. A set of tasks N is said to be *schedulable* if there exist a feasible schedule S for it.

In the following, we also assume that a set F of old tasks is already scheduled. We denote by $A = N \cup F$ the overall set of tasks (old and new). We consider different scheduling strategies: old tasks can either be fixed in their previous position, or movable to facilitate the placement of new tasks; new tasks N can either be forced to be scheduled, or their scheduling can be optional. These strategies generate four variants of our problem with different levels of "flexibility".

To our knowledge, the variants with optional scheduling correspond to a new multiple-period version of the knapsack problem, that has not appeared in the literature so far [1, 2]. In the problem domain, no works that we know of have addressed the problem dealt with in this paper. Some works (e.g., [3]) deal with adjusting the offsets of tasks to minimize the delay between the activation of a periodic task instance and its scheduling in the hyperperiod. Others (e.g., [4]) try to predict which tasks should be scheduled periodically and which should not. For the sake of brevity, we only present the formulations corresponding to the two variants with *movable* old tasks.

3 Conflict-Based (CB) formulations

This formulation uses binary variables to represent overlaps in time and space between pairs of tasks:

$$ct_{i,j} = \begin{cases} 1 & \text{if tasks } i \text{ and } j \text{ overlap in time} \\ 0 & \text{otherwise} \end{cases} \qquad i, j \in A,$$

$$ch_{i,j} = \begin{cases} 1 & \text{if tasks } i \text{ and } j \text{ overlap in space} \\ 0 & \text{otherwise} \end{cases} \qquad i, j \in A.$$

To identify time overlaps we define the set $T(i,t,j)$ that contains all TTIs t' for which task j would generate a time overlap in the hyperperiod with task i aired in TTI t. The set is defined for all task pairs $i, j \in A$, and for all possible TTIs $t \in \{0, \dots, \pi_i - 1\}$ of task i:

$$T(i,t,j) = \left\{ t' \in \{0, \dots, \pi_j - 1\} \mid t' - t \equiv 0 \ (mod \ \mathrm{gcd}(\pi_i, \pi_j)) \right\}.$$

Note that we only need to identify the initial TTIs (i.e., in the first period of a task) because these uniquely define all the subsequent periodic allocation times. Similarly, the set $H(i,h,j)$ contains all the initial RB positions h' for which task j would generate a space overlap with task i whose first RB position is h:

$$H(i,h,j) = \left\{ h' \in \{1, \dots, M - w_j + 1\} \mid \ [h, h + w_i - 1] \cap \left[h', h' + w_j - 1\right] \neq \emptyset \right\}.$$

Next, we use binary variables to represent the assignment to a task of the initial TTI

$$x_{i,t} = \begin{cases} 1 & \text{if task } i \text{ is first aired in TTI } t \\ 0 & \text{otherwise} \end{cases} \qquad i \in A, \ t \in \{0, \dots, \pi_i - 1\},$$

and the initial RB

$$y_{i,h} = \begin{cases} 1 & \text{if task } i \text{ has first RB position } h \\ 0 & \text{otherwise} \end{cases} \qquad i \in A, \ h \in \{1, \dots, M - w_i + 1\}.$$

The model for the **Movable-Forced (M-F)** variant looks as follows:

$$\text{(CB-M-F)}\ \min \sum_{i \in F} z_i \tag{1}$$

$$\text{s.t.} \sum_{t=0}^{\pi_i - 1} x_{i,t} = 1 \qquad i \in A \tag{2}$$

$$\sum_{h=1}^{M-w_i+1} y_{i,h} = 1 \qquad i \in A \tag{3}$$

$$ct_{i,j} \geq x_{i,t} + \sum_{t' \in T(i,t,j)} x_{j,t'} - 1 \qquad i,j \in A,\ t \in \{0,\dots,\pi_i - 1\} \tag{4}$$

$$ch_{i,j} \geq y_{i,h} + \sum_{h' \in H(i,h,j)} y_{j,h'} - 1 \qquad i,j \in A,\ h \in \{1,\dots,M - w_i + 1\} \tag{5}$$

$$ct_{i,j} + ch_{i,j} \leq 1 \qquad i,j \in A \tag{6}$$

$$z_i \geq 1 - x_{i,t_i}/2 - y_{i,h_i}/2 \qquad i \in F \tag{7}$$

$$x_{i,t} \in \{0,1\} \qquad i \in A,\ t \in \{0,\dots,\pi_i - 1\} \tag{8}$$

$$y_{i,h} \in \{0,1\} \qquad i \in A,\ h \in \{1,\dots,M - w_i + 1\} \tag{9}$$

$$ct_{i,j} \in \{0,1\} \qquad i,j \in A \tag{10}$$

$$ch_{i,j} \in \{0,1\} \qquad i,j \in A \tag{11}$$

$$z_i \in \{0,1\} \qquad i \in F \tag{12}$$

This variant of the problem forces all new tasks to be scheduled, but we are allowed to move old ones. In constraints (7) we use variables x_{i,t_i} and y_{i,h_i} to represent the "original" position in time and space, respectively, of an old task $i \in F$. If any of the two variables is set to zero, the old task is moved (in time and/or in space). The objective (1) is to minimize the number of old tasks moved, so we define binary variables z_i, $\forall i \in F$ to keep track of old tasks moved:

$$z_i = \begin{cases} 1 & \text{if task } i \text{ is moved} \\ 0 & \text{otherwise} \end{cases} \qquad i \in F.$$

Note that equations (2)–(3), expressing the assignment of an initial TTI and RB to the tasks, refer to both new and old tasks ($i \in A$), since old tasks can be assigned a different schedule (if they are moved). Constraints (4)–(5) keep track of time and space overlaps, while inequalities (6) forbid simultaneous overlaps in both dimensions, hence conflicts.

A more flexible variant, called **Movable-Optional (M-O)**, is obtained by making the scheduling of new tasks optional, the model is:

$$\text{(CB-M-O)} \quad \max \sum_{i \in N} \sum_{t=0}^{\pi_i - 1} x_{i,t} - \frac{1}{|F|+1} \sum_{i \in F} z_i \tag{13}$$

$$\text{s.t.} \quad \sum_{t=0}^{\pi_i - 1} x_{i,t} = 1 \qquad i \in F \tag{14}$$

$$\sum_{h=1}^{M-w_i+1} y_{i,h} = 1 \qquad i \in F \tag{15}$$

$$\sum_{t=0}^{\pi_i - 1} x_{i,t} \leq 1 \qquad i \in N \tag{16}$$

$$\sum_{h=1}^{M-w_i+1} y_{i,h} \leq 1 \qquad i \in N \tag{17}$$

$$\sum_{t=0}^{\pi_i - 1} x_{i,t} - \sum_{h=1}^{M-w_i+1} y_{i,h} = 0 \qquad i \in N \tag{18}$$

$$(4),\ (5),\ (6),\ (7),$$
$$(8), (9),\ (10),\ (11),\ (12)$$

In this case the objective function (13) consists of two terms, one maximizes the number of new tasks that are scheduled, while the other minimizes the number of old tasks moved. The weight of the latter is < 1, thus among the two policies the former prevails. Note that with respect to the previous variant, the assignment constraints are slightly different, as one needs to distinguish between old and new tasks. Equations (14)–(15) guarantee that old tasks are always scheduled, as they either keep their old schedule or they are moved. The scheduling of new tasks, instead, is optional, as expressed by inequalities (16)–(17); constraints (18) make sure that if a new task receives a position in one dimension, it also receives a position in the other dimension.

4 Matrix-Based (MB) formulations

We use three-index binary variables representing the assignment to both dimensions:

$$x_{i,h,t} = \begin{cases} 1 & \text{if task } i \text{ has first RB position } h \text{ and first TTI } t \\ 0 & \text{otherwise} \end{cases}$$

for all $i \in A,\ h \in \{1, \ldots, M - w_i + 1\},\ t \in \{0, \ldots, \pi_i - 1\}$.

To identify conflicts we define a set for each space-time position (h, t), $h \in \{1, \ldots, M\}$, $t \in \{0, \ldots, H-1\}$ containing all the schedules that use RB h at TTI t:

$$C(h,t) = \{ \; (i, h', t') \; i \in A, \; h' \in \{1, \ldots, M - w_i + 1\}, \; t' \in \{0, \ldots, \pi_i - 1\} \mid \\ \text{if } i \text{ is in } (h', t'), \; i \text{ has assigned also the RB } (h, t) \; \}.$$

We denote, as before, by h_i and t_i the original space and time positions respectively of an old task $i \in F$. To keep track of old tasks that are moved, we define the following parameter:

$$\alpha_{i,h,t} = \begin{cases} 1 & \text{if } h \neq h_i \text{ or } t \neq t_i \\ 0 & \text{otherwise} \end{cases} \qquad i \in F, \; h \in \{1, \ldots, M{-}w_i{+}1\}, \; t \in \{0, \ldots, \pi_i{-}1\}.$$

The model for the **Movable-Forced (M-F)** variant is:

$$\text{(MB-M-F)} \qquad \min \sum_{i \in F} \sum_{h=1}^{M-w_i+1} \sum_{t=0}^{\pi_i - 1} \alpha_{i,h,t} x_{i,h,t} \tag{19}$$

$$\text{s.t.} \sum_{h=1}^{M-w_i+1} \sum_{t=0}^{\pi_i-1} x_{i,h,t} = 1 \qquad i \in A \tag{20}$$

$$\sum_{(i,h',t') \in C(h,t)} x_{i,h',t'} \leq 1 \qquad h \in \{1, \ldots, M\}, \; t \in \{0, \ldots, H-1\} \tag{21}$$

$$x_{i,h,t} \in \{0, 1\} \quad i \in A, \; h \in \{1, \ldots, M - w_i + 1\}, \; t \in \{0, \ldots, \pi_i - 1\} \tag{22}$$

Note that the M-B formulation does not need additional variables to represent old tasks that are moved, since the condition is given by parameter $\alpha_{i,h,t}$ associated to all assignment variables in the objective function. Assignment in time and space (for all tasks) is obtained via constraints (20). Conflicts are forbidden by inequalities (21), which, for each space-time position (h, t), make sure that at most one schedule in $C(h, t)$ (i.e., a schedule using (h, t)) is selected.

Next, we give the formulation for the **Movable-Optional (M-O)** variant, for which, as we already observed, the scheduling of new tasks is optional:

$$\text{(MB-M-O)}\ \max \sum_{i\in N}\sum_{h=1}^{M-w_i+1}\sum_{t=0}^{\pi_i-1} x_{i,h,t} - \frac{1}{|F|+1}\sum_{i\in F}\sum_{h=1}^{M-w_i+1}\sum_{t=0}^{\pi_i-1} \alpha_{i,h,t}x_{i,h,t} \tag{23}$$

$$\text{s.t.}\ \sum_{h=1}^{M-w_i+1}\sum_{t=0}^{\pi_i-1} x_{i,h,t} = 1 \qquad i\in F \tag{24}$$

$$\sum_{h=1}^{M-w_i+1}\sum_{t=0}^{\pi_i-1} x_{i,h,t} \le 1 \qquad i\in N \tag{25}$$

(20), (21).

The objective function (23), as already observed for the C-B formulation, maximizes the number of new tasks scheduled and minimizes the number of old tasks moved. Assignment constraints (24) guarantee a schedule for all the old tasks $i \in F$, while for new tasks $i \in N$ the scheduling is optional (25).

5 Computational results

The models are implemented and solved using `CPLEX Callable Library`. We generated realistic instances taking into account technical aspects of the application. An instance of our problem is characterized by the following elements:

- size M of the airframe
- RB demand and period of new tasks (w_i, π_i), $i \in N = \{1 \dots n\}$
- set of fixed tasks F

The idea underlying the construction of four variants for the problem is to solve them in sequence. Namely, if one variant turns out to be infeasible we set to solve the next one:

1. **-F-F: this is a purely feasibility problem, that checks if all new tasks can be scheduled without moving the old ones;
2. **-M-F: in this variant we are free to move old tasks if this allows to schedule all new task, the objective function minimizes the number of old tasks moved;
3. **-F-O: in this variant old tasks are fixed, but the scheduling of new tasks is optional, which guarantees to find a feasible solution;

4. **-M-O: this is the most flexible and complex "knapsack-like" variant having both degrees of freedom (i.e., old tasks can be moved and the scheduling of new tasks is optional).

In our experiments we compare the sequential approach with the direct solution of the last variant **-M-O. The comparison is performed for both types of formulations, so "**" is either CB (Conflict-Based) or MB (Matrix-Based). We use a time limit of 3600 seconds. In the sequential approach the time limit is split among the different variants:

- 600 seconds for **-F-F
- 1200 seconds for **-M-F
- 1800 seconds for **-F-O

In Tables 1 and 2 we report, for formulations CB and MB respectively, the average solution times (in seconds) on all the task periods considered, for each value of M and $|N|$:

	M=10		M=25		M=50		M=100	
$\|N\|$	CB-seq	CB-M-O	CB-seq	CB-M-O	CB-seq	CB-M-O	CB-seq	CB-M-O
20	14	32	29	94	6	37	9	70
30	165	141	79	166	108	119	132	108
50	526	866	951	2752	709	1310	1605	1725
60	1803	2925	1450	2376	518	1307	1146	1752
90	3256	3014	3109	2839	3251	2849	3236	2841

Table 1: Solution times (seconds) of sequential approach *vs.* M-O variant for the Conflict-Based formulation.

	M=10		M=25		M=50		M=100	
$\|N\|$	MB-seq	MB-M-O	MB-seq	MB-M-O	MB-seq	MB-M-O	MB-seq	MB-M-O
20	69	902	1229	1814	4	2242	914	–
30	101	1203	545	1850	334	2291	1336	–
50	305	1939	1753	2798	1545	2936	22	–
60	975	2444	1031	2576	1452	3314	1222	–
90	2472	2903	2401	2913	1806	3600	44	–

Table 2: Solution times (seconds) of sequential approach *vs.* M-O variant for the Matrix-Based formulation.

These preliminary results show that the sequential approach consistently performs better than the **-M-O variant for both formulations, except for the largest instances ($|N| = 90$) for which CB-M-O has a better performance. Indeed 43% of the tested instances can be solved by the first two variants (**-F-F and **-M-F) during the sequential approach. The instances in the last column of Table 2 ("–") could not be solved within the time limit.

Acknowledgements The authors gratefully acknowledge partial financial support from the University of Pisa, under the project “Analisi di reti complesse: dalla teoria alle applicazioni” (grant PRA_2020_61), and from the Italian Ministry of University and Research, under the CrossLab grant (Departments of Excellence).

References

1. Cacchiani, V. and Iori, M. and Locatelli, A. and Martello, S. (2022) Knapsack problems – An overview of recent advances. Computers & Operations Research, doi 10.1016/j.cor.2021.105692.
2. Kellerer, H. and Pferschy, U. and Pisinger, D. (2004). *Knapsack Problems.* Berlin, Springer.
3. Jiang, N., and Aijaz, A. and Jin, Y. (2021): Recursive Periodicity Shifting for Semi-Persistent Scheduling of Time-Sensitive Communication in 5G. GLOBECOM 2021: 1-6.
4. He, Q. and Dán, G. and Koudouridis, G. P. (2021): Semi-Persistent Scheduling for 5G Downlink based on Short-Term Traffic Prediction, GLOBECOM 2020 - 2020 pp. 1-6.