Rule-Based Scheduling for MPARs Performing Sensing and Communications

Augusto Aubry
DIETI
University of Naples "Federico II"
Napoli, Italy
augusto.aubry@unina.it

Antonio De Maio
DIETI
University of Naples "Federico II"
Napoli, Italy
ademaio@unina.it

Luca Pallotta
Department of Engineering (DiING)
University of Basilicata
Potenza, Italy
luca.pallotta@unibas.it

***Abstract*—Multifunction phased array radar (MPAR) is capable of performing sensing and communications by functionally grouping a phased array into tailored sub-apertures, each dedicated to a distinct task. Because of limited available resources, such as bandwidth, power aperture product, and time, it is important to properly allocate them to each sub-aperture. This article examines a rule-based task scheduling algorithm wherein communication (COM) looks are employed to fill the vacant time left by the radar tasks that are allocated first (namely, volume and cued search, update and confirmation tracking). The allocation of looks is carried out for each time slot based on task priorities; however, some tasks (i.e., volume, cued, and COM) are executed in parallel when the available bandwidth and power-aperture product (PAP) permit. Simulations conducted in scenarios demonstrate the validity of the proposed allocation strategy in terms of bandwidth utilization and time occupancy.**

***Index Terms*—multifunction phased array radar (MPAR), priority-based strategy, sensing and communications, task scheduling.**

I. Introduction

The aperture array can be divided into several sub-apertures and multiple beams can be steered in different directions, depending on the task at hand. To this end, the multifunction phased array radar (MPAR) requires a radar resource manager (RRM), which is responsible for an optimal allocation of the limited radar resources (e.g., bandwidth, PAP, time, etc.) to each demanding function (or tasks associated to them) [1]–[4]. Each radar task produces one or more look requests for every update interval, with each request necessitating a specific amount of resources to achieve its objectives. A radar look stores details regarding the timing, location, and duration for which the MPAR needs to direct its beam (with the description of its related parameters), along with the transmitted waveform characteristics. The RRM should perform a scheduling of the tasks to be executed over time. During the resource distribution process, due to the limited budget, some tasks cannot be allocated in the current time slot (or update interval) under consideration, and should be postponed. A timeline should be established that describes the tasks to execute inside each update time, defining their starting time, duration, and the resources they will employ [5]–[8]. Tasks ranking according to priority values [2], [3], [9]–[11] is a key process to perform before any scheduling method.

An example of task scheduling based on a rule-based priority assignment for a MPAR doing volume search, cued search, and tracking [2] is provided in [10]. At each update interval, the scheduler organizes the look requests based on their start times, giving precedence to tasks with higher priority. The shared resources to be considered include bandwidth, PAP, and time. Depending on the amount of resources (e.g., bandwidth and PAP) required by each involved task, volume and cued search can be conducted simultaneously, while tracking demands all available resources during its execution. To address the limitation of this approach (where unused time may remain in each update interval) a radar-centric strategy is proposed in [11], which assigns COM looks to fill the gaps left by radar tasks during the update interval. The scheduler initially allocates higher radar priority tasks, then employs COM operations to maximize the efficiency of the available time resources. COM tasks hold the lowest priority, but they can run concurrently with volume search tasks when a cued search is not performed. Additionally, they are scheduled during the idle periods left unutilized by radar operations in each update interval. Non-scheduled tasks (also known as deferred tasks) are postponed to the subsequent update intervals and their priority is increased, except for COM tasks.

In this paper, the strategy outlined in [11] is adjusted to enable the system to perform certain volume tasks concurrently to restore the desired timeline interrupted by an unexpected overload. Simulations carried out in relevant practical scenarios demonstrate that the proposed framework remains effective in both of its variants.

II. Problem description

A key requirement for the MPAR to perform various tasks simultaneously is its ability to partition its phased array into multiple subarrays. The tasks (precisely the associated look) to be executed are arranged (as best as possible) within the forthcoming update time. In each update time, a radar task generates one or more look requests; each of them incorporates several information useful to properly allocate resources (e.g., bandwidth, transmitting power, steering angle, dwell time, etc.) as needed to achieve the task goal. Subsequently, the RRM, on the base of its scheduling algorithm, arranges all the generated looks into a feasible timeline that is composed by successive update times, each of them comprises the scheduled looks. Hence, once the timeline for the upcoming update interval has been completed, the RRM moves the non-scheduled tasks into a list of deferred tasks to be scheduled next.

A. System resources

In this paper, the bandwidth, the PAP, and the time duration are considered to be limited resources that should be shared among the tasks. With regards to bandwidth, it is assumed that the tracking task (and hence each required look) requires the entire available bandwidth for its execution. In contrast, the cued search necessitates a lesser bandwidth requirement, whereas the volume task possesses the smallest bandwidth requirements. Lastly, for COM operations the bandwidth is set to allow them to be executed in parallel with a volume task.

The PAP of the h-th is the activated subarray is defined as the product of the average transmitted power, P_{av}, and the aperture size, A_e, assigned to it, namely [12]

$$\mathrm{PAP}_h = P_{av}^h A_e^h = P_t^h \tau \mathrm{PRF} A_e^h, \tag{1}$$

with P_t^h the peak transmitted power, τ the pulse width, and PRF the pulse repetition frequency.

Furthermore, each look requires a different duration depending on several factors. Regarding the radar tasks, the duration of each look is determined by the dwell time, that is the pulse repetition interval (PRI) multiplied by the number of pulses in a burst. Thus, denoting by $D_x(P_d, P_{fa})$ the radar detectability factor (that is a quantity determining the received signal energy needed to achieve the desired probability of detection P_d given the probability of false alarm P_{fa}), the number of pulses for each radar look is [12]

$$n = \left\lceil \frac{(4\pi)^3 k_B T_s D_x(P_d, P_{fa}) R^4 L_s L_h(\theta, \phi)}{P_t^h \tau G_h^2 \lambda_0^2 \sigma} \right\rceil, \tag{2}$$

with $k_B = 1.38 \times 10^{-23}$ J/K the Boltzmann constant, T_s the system noise temperature, R the target range, G_h the gain of the subarray, λ_0 the operating wavelength and σ the target radar cross section (RCS). The function $\lceil \cdot \rceil$ extracts the integer part from its argument. Finally, L_s and $L_h(\phi, \theta)$ are the system loss and scanning gain loss, with the latter depending on the pointing azimuth ϕ and elevation θ angles.

B. Volume and cued search tasks

Each volume search look is characterized by a pointing direction in a grid of directions. Based on this, beam and waveform features are established accordingly. In particular, in order to accomplish a comprehensive scanning of a large area, the volume task is distinguished by a broad beam (coarse grid of directions) and a limited bandwidth. If during a volume search a detection is obtained, a cued search task is initialized. A cued search look consists of seven or five beams that should be sequentially transmitted in order to refine the search around the specific direction where the volume search has triggered a detection. In this regard, the cued search is distinguished by a narrower beam and a broader bandwidth in comparison to the volume search. If the detection is confirmed by the cued, the respective measurements are passed to the processing unit, which will then activate the tracking process.

C. Tracking task

The tracking task may involve in general two different type of looks. The latter is a track confirmation that is activated when the detection resulting from the cued task cannot be associated with any other existing tracks. When this is the case, the tracking task must initiate a new unconfirmed track through a request for a confirmation look. In contrast, if the measurement associated with the cued detection can be associated with an existing track, the tracking task proceeds with a request for a track update look. Generally, the task of confirmation may be deemed more significant than the update task, hence necessitating a shorter revisit time.

The tracker exploits an Interacting Multiple Model (IMM) filter designed for highly maneuvering targets [12]–[14]. It addresses the uncertainty of target motion by utilizing multiple models simultaneously for a maneuvering target. All of them are used to come up with a combined state estimate, while also utilizing the model probabilities and model switching probabilities. Particularly noteworthy is the utilization (within the IMM framework) of two extended Kalman filters (EKFs) in parallel, specifically tailored for two distinct motion models, viz., constant velocity (for non-maneuvering targets) and constant turn (for maneuvering targets).

In the case studied in this paper, the model transition probabilities are both set equal to 0.99. When a tracking task requests a track update, the revisit time is adaptively selected in accordance with the target maneuver. If the model probability of the maneuvering model is greater than a predefined value (e.g., $P_m > 0.5$), the revisit time is chosen equal to the lowest possible, otherwise it is selected as the highest possible.

D. COM task

In addition to sensing operations, it is assumed that the MPAR could communicate with a number of U users. Hence, the system can transmit a signal given by the superposition of U frequency (or code) orthogonal waveforms. Once the COM range for the k-th user is identified $R_{k,\mathrm{COM}}$, the PAP is computed as the value such that the channel capacity per bandwidth (expressed in bit/s/Hz) is equal to a specific value C_{desired}, that is [11], [15]

$$\overline{PAP}_k = \frac{\left(2^{C_{\mathrm{desired}}} - 1\right) \lambda_0^2 R_{k,\mathrm{COM}}^2 L_s^{\mathrm{COM}} L_{\mathrm{steer}}^{\mathrm{COM}} k_B T_s^{\mathrm{COM}} B^{\mathrm{COM}}}{A_e^{\mathrm{rx},k}}, \tag{3}$$

with $A_e^{\mathrm{rx},k}$ the effective area of the k-th user receiving antenna, L_s^{COM} the COM system operational loss, $L_{\mathrm{steer}}^{\mathrm{COM}}$ the scanning loss, B^{COM} the bandwidth available in the MPAR for COM operations, and T_s^{COM} the noise system temperature.

III. RULE-BASED TASK SCHEDULING ALGORITHM

The scheduling algorithm considered in [11] relies on priorities assigned to each task, which range from 1 (low) to 5 (high). The RRM tries to distribute the temporal resource according to the required start times, while respecting the prioritization. Furthermore, if both PAP and bandwidth can be simultaneously shared between two tasks, they can also be scheduled for being executed in parallel. Hence, priority are assigned as 1, 2, 3, 4, and 5 to COM, volume search, cued search, track update, and track confirmation, respectively. Nevertheless, a dynamic allocation of priority is assumed, too. Indeed, all tasks that are not assigned for execution in the upcoming update interval are placed on a list of deferred tasks, and their priority is increased by a factor of 0.25 (with the exception of COM tasks) to a maximum priority less than 5. Additionally, COM tasks are assigned in parallel with volume search tasks, whenever a cued search is not performed, ensuring that resource constraints are not violated. Furthermore, COM looks are allocated to fill the gaps within the update interval that are left by other radar tasks (for instance, during intra-slot time). Regarding the strategy developed in this paper, certain volume tasks are also permitted to be executed in parallel to recover the search volume timeline. The proposed method can be further examined in relation to the block diagram presented in Figure 1.

The algorithm initiates the acquisition of all look requests from tasks whose start time falls within the current update window. Then, it extracts from that list the confirmation tracking looks, which have the highest priority. Thus, these tracking looks are scheduled in order to turn on the filling process of the update interval. If confirmation tracking looks are no longer present, and there is a temporal margin to insert other looks in the update interval, then the RRM continues to explore the look requests list. The RRM schedules looks having priorities higher than 3 (according to their priority), which could be both update tracking tasks and other deferred radar tasks. After that, the RRM checks for the presence of a cued search in the list (unless a previously deferred task has reached a higher priority). They are scheduled sequentially just after the already scheduled tracking looks and in parallel to possible volume search looks. If the remaining tasks have a duration that is not in line with the residual time until the end of the update window, they are moved into the list of deferred tasks.

In [11], the volume search tasks are scheduled sequentially because of the lack of necessity to increase the scan rate of the search grid for typical surveillance scenarios [12]. However, when there are multiple track operations, volume tasks could be also done in parallel. This modification to the scheduling algorithm is proposed in this paper, where volume tasks can be also allocated to be executed in parallel. As a matter of fact, there could be situations where, due to an unexpected overload, the radar could execute some volume tasks in parallel to re-establish a desired timeline. It is noteworthy to mention that a situation of practical significance where it is

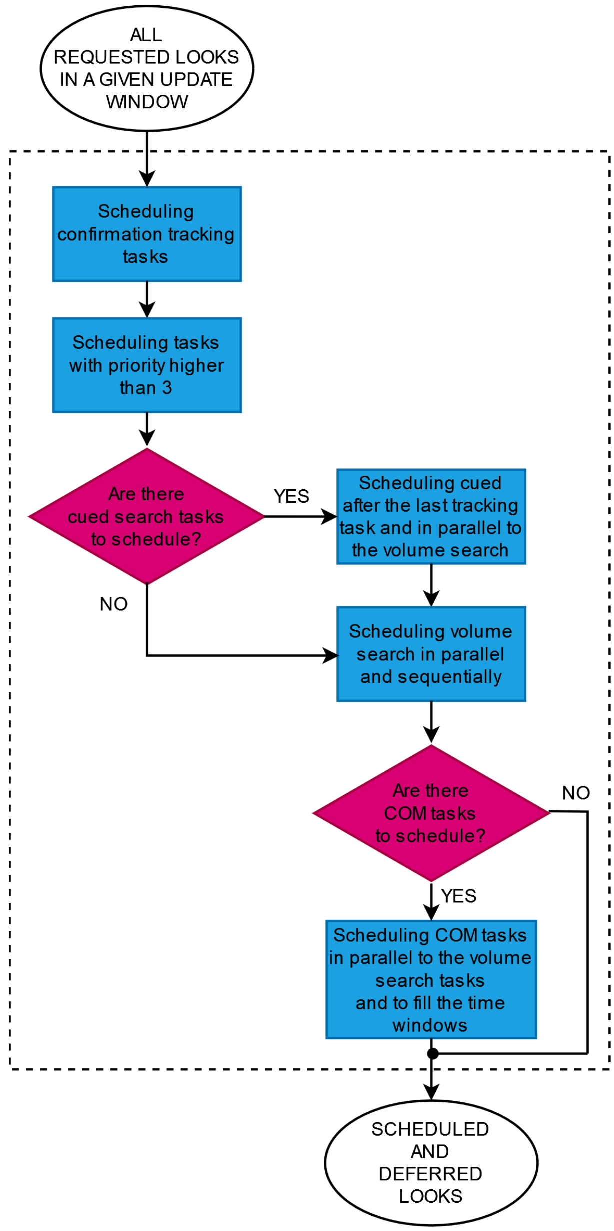

Fig. 1. Proposed algorithm for task scheduling in a MPAR system performing volume search, cued search, tracking, and COM operations.

preferable to refrain from simultaneously executing volume search arises when energy harvesting is required, such as when it is necessary to probe the environment with a burn-through waveform. However, the choice of considering or not volume search looks in parallel is demanded to the user depending on the specific scenario under consideration.

Once the sensing task scheduling has been terminated, the RRM involves the completion of the update interval with the COM tasks. According to the amount of available resources, COM looks can be executed in parallel to all single volume search tasks, or when there is sufficient empty space in the update window. In other words, COM looks are employed to fill the gap between radar tasks and the end of the update interval, as well as fill the spectral holes, in order to optimally utilize the timeline.

IV. Simulation results and discussion

This section examines the results of the devised resource allocation algorithm described in Section III. The proposed approach is compared with the algorithm of [11], which does not perform parallelization of volume search tasks, and that of [10], which solely considers sensing tasks.

The simulation setting is the same as that adopted in [10], [11], where a long-range S-band MPAR ($\lambda_0 = 0.09$ m) is examines. The simulating scenario obtained using the Mathworks Matlab toolbox *Multibeam Radar for Adaptive Search and Track* [10] assumes the presence of diverse maneuvering targets and trajectories. The targets are simulated according to non-fluctuating Swerling 0 (SW0) model with RCS $\sigma^2 = 1$ m^2, whereas the MPAR is characterized by having a peak transmit power of 100 kW, an overall available bandwidth of 10 MHz, a noise figure of 5 dB, and a beamwidth of $2°$ and $3°$ in azimuth and elevation, respectively, when the beam points at boresight.

For each involved task, their bandwidths and PAP are assigned as provided in Table I, where also the other simulation parameters are summarized (note that the tracking tasks require all the available PAP, i.e., 1661.9 Wm^2). Regarding to the PAP, it is computed for each sensing task according to (1) with the parameters of Table I. Differently, for the COM task the PAP is computed through (3), setting $C_{desired} = 8$ bit/s/Hz, and randomly choosing the range at each simulating trial as a realization of a uniform random variable in the interval $[18, 22]$ km. Moreover, the time duration for each radar task is quantified by its dwell that is in turn obtained from the number of pulses n evaluated through (2). Differently, the duration of the COM signals is maintained constant and equal to 0.01 s.

TABLE I
Tasks simulation parameters.

parameter	**value**			
	Tracking	Volume	Cued	COM
Az. beamwidth ($°$)	2	8	4	4
El. beamwidth ($°$)	3	10	5	5
bandwidth (MHz)	10	0.5	3	9.5
PAP (Wm^2)	1661.9	498.58	997.15	adaptive
τ (μs)	2.5	10	5	N/A
PRF (Hz)	1500	1500	1500	N/A
P_d	0.9	0.9	0.9	N/A
P_{fa}	10^{-6}	10^{-6}	10^{-6}	N/A

Other simulation parameters are the range and angular limits for the volume search, which are set equal to 75 km (range limit), $[-60°, 60°]$ (azimuth limit), and $[-30°, 0°]$ (elevation limit). Moreover, the space between adjacent beams in the volume search grid is set equal to 0.85 beamwidth fraction in both azimuth and elevation. The tracking tasks are performed considering a confirmation rate equal to 20 Hz, whereas for the update tracking, the revisit time is adaptively chosen as described in Section II-C, with the highest value set equal to 2 s and the lowest to 0.2 s. Furthermore, the update time is set equal to a constant value of 0.05 s. Finally, as to the COM user equipment, the following parameters setting is selected, $L_s^{\text{COM}} = 27$ dB, $T_s^{\text{COM}} = 916$ K, and $A_e^{\text{rx}} = 0.7 \times 10^{-3}$ m^2. Finally, the simulation is conducted for a total duration of 4 s, and it starts at time 0 s with a volume look.

Let us now analyze the task scheduling produced by the RRM with all the tasks executed in each update time together with their duration. To this end, Figures 2 shows some snapshots of the timeline for the scheduled tasks along the conducted simulation. Each subplot in the figure refers to a specific update slot (arranged in time increasing order, top-down), with the specific values of the time interval on the x-axis. Moreover, subplots refer to the scheduling algorithm of [10] (first column), algorithm of [11] (central column), and the proposed (column on the right). As expected, thanks to the possibility of sharing bandwidth and PAP, volume and cued search are executed in parallel as can be seen from subplots (d), (e), and (f) of Figure 2. Similarly, COM operations are performed parallel to the volume search (in absence of cued activities) subplots (e) and (f) of Figure 2 for the algorithm of [11] and the proposed, respectively. Moreover, they also occupy in a useful way the empty time left by radar tasks. As expected the tracking tasks (both confirmation and update) are executed alone since they require all the available resources. This is evident in the same subplots for confirmation track as well as in subplots (g), (h), and (i) for both confirmation track and track update. Interestingly, in subplot (i) two volume tasks are executed in parallel by the proposed algorithm, where one volume instead of being deferred as done by the method [11] is anticipated in the current update time. By doing so, the overall scheduling results to be modified. In fact, in the next update time while the algorithm of [11] still performs volume searches, see subplot (k), the proposed strategy realize a cued search in parallel to the volume, see subplot (l), since a detection occurred in the previous update time, by the "anticipated" search look.

With reference to the above described scenario, Figure 3 shows the average utilized bandwidth, computed within each update interval, depending on how the tasks are executed over time. As expected, for both the proposed algorithm and that of [11], the occupied bandwidth is close to its maximum available value, since a very high bandwidth is associated to the COM tasks (viz. 9.5 MHz) that is the task performed more frequently. The areas where the occupied bandwidth reduces correspond to the update times where cued searches are performed (having a bandwidth of 3 MHz alone). As to the algorithm of [10] instead, it has a very low usage of the bandwidth resource, whose average value is close to 0.6 MHz. This is not surprising since this method is a sub-case of the proposed framework and it does not perform COM operations. To have a quantitative measure of the bandwidth utilization, the bandwidth efficiency, defined as the average bandwidth occupancy in the overall execution time, is introduced. It is

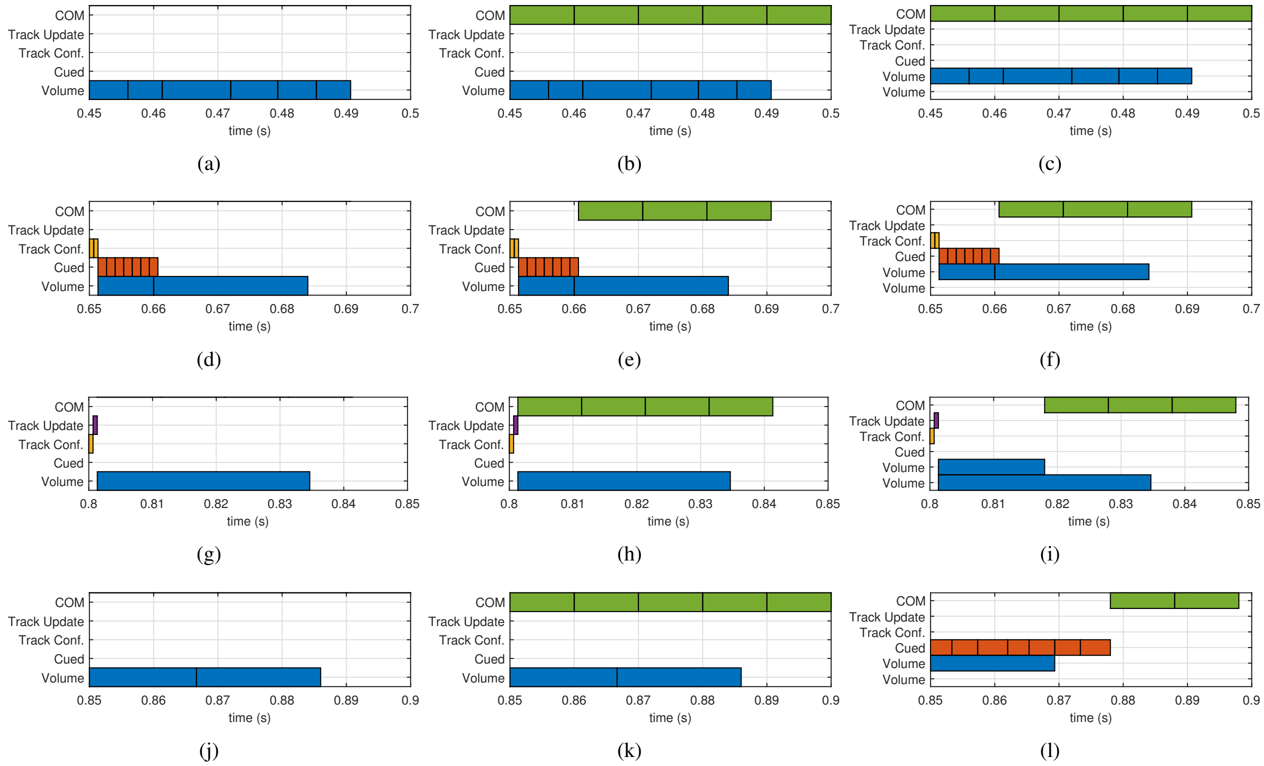

Fig. 2. Some snapshots of the timeline for the scheduled tasks along the conducted simulation from 0 to 4 s. Subplots refer to the scheduling algorithm of [10] (first column), procedure of [11] (central column), and the proposed framework (column on the right).

computed as the mean value of the average bandwidths over each update time. The respective values are shown in the first column of Table II (other information detailed later is also reported), where the evidence is that the proposed framework and that of [11] allow to increase the bandwidth usage with respect to the competitor, reaching an almost complete utilization of it. Similarly, Figure 4 shows the average utilized PAP versus time. Interestingly, all the considered methods have almost the same PAP consumption. This is essentially due to fact that COM tasks and volume in parallels do not have high PAP requirements.

Finally, Figure 5 emphasizes the time occupancy of the MPAR in the considered scenario of the proposed framework. It is computed as the ratio between the time for which at least one look is executed over the total update time, and expressed as a percentage. As expected, the proposed approach allows to exploit the time resource very efficiently. As a matter of fact, the time occupancy is often close to 100% (see the black dashed curve), with a limited reduction in the update intervals where it is not possible to allocate COM tasks to fill all the possible time gaps. As per the bandwidth analysis, the time efficiency is used to glean a synthetic quantitative measure of the effectiveness of the proposed method. Specifically, it is defined as the average occupancy of time with respect to the entire observation time (expressed in percentage). The time efficiency values are also reported in Table II, with the proposed strategy showing the highest value of 96.44%.

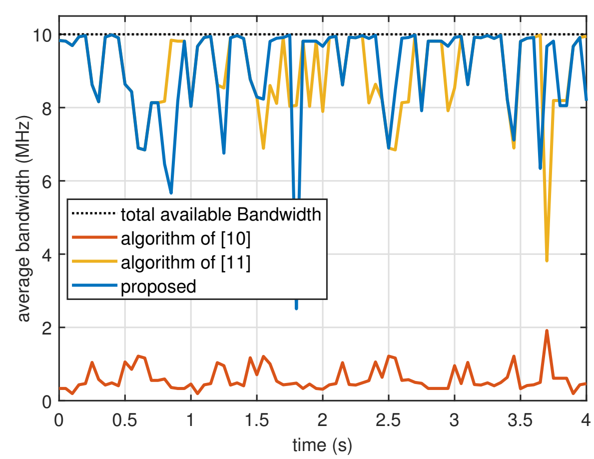

Fig. 3. Use of bandwidth (average on each update time) resource versus time. Subplots refer to the algorithm of [10] (top), algorithm of [11] (central), proposed strategy (bottom).

V. Conclusions

This paper examines the issue of task scheduling in MPARs performing multiple sensing and communication activities. In this respect, a rule-based approach has been developed after assigning different priorities to the involved tasks and looks. Therefore, the guidelines governing the scheduling of

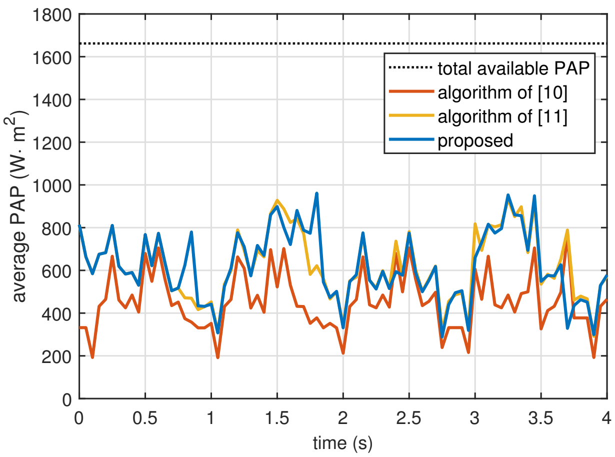

Fig. 4. Use of PAP (average on each update time) resource versus time. Subplots refer to the algorithm of [10] (top), algorithm of [11] (central), proposed strategy (bottom).

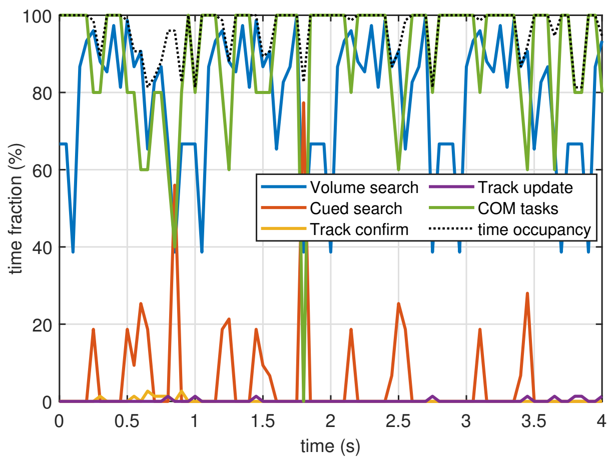

Fig. 5. Time occupancy versus time for the proposed scheduling strategy.

TABLE II
BANDWIDTH AND TIME EFFICIENCY (EXPRESSED AS A PERCENTAGE).

algorithm	value	
	bandwidth utilization (%)	time occupancy (%)
algorithm of [10]	5.98	80.02
algorithm of [11]	89.55	94.45
proposed strategy	90.56	96.44

tasks within the timeline have been established to enhance the efficiency of the bandwidth and time utilization across the entire architecture. In particular, the COM operations are used during the scheduling process as a gap-filler to minimize the amount of unused time. These looks are also organized so that the available bandwidth at the MPAR results is almost completely filled. Furthermore, volume tasks are also executed in parallel, when possible, to maintain as close as possible the search timeline to the nominal one. The validity of the proposed solution, which can be selected by the user as an alternative to [11] based on the situational awareness, has been demonstrated through simulation results conducted on some intriguing scenarios.

ACKNOWLEDGMENTS

The work of Augusto Aubry and Antonio De Maio was supported by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, partnership on “Telecommunications of the Future” (PE00000001 - Program “RESTART”).

REFERENCES

[1] A. Farina, S. Pardini, and G. Roselli, “General Purpose Control of a Multifunctional Phased-Array Radar,” *Rivista Tecnica Selenia*, vol. 6, no. 2, pp. 12–20, 1979.

[2] P. Moo and Z. Ding, *Adaptive Radar Resource Management*. Academic Press, 2015.

[3] A. Charlish and F. Katsilieris, “Array Radar Resource Management,” *Novel Radar Techniques and Applications: Real Aperture Array Radar, Imaging Radar, and Passive and Multistatic Radar*, vol. 1, pp. 135–171, 2017.

[4] A. Charlish and F. Hoffmann, “Cognitive Radar Management,” *Novel Radar Techniques and Applications: Waveform Diversity and Cognitive Radar, and Target Tracking and Data Fusion*, vol. 2, pp. 157–193, 2017.

[5] W. K. Stafford, “Real Time Control of a Multifunction Electronically Scanned Adaptive Radar (MESAR),” in *IEE Colloquium on Real-Time Management of Adaptive Radar Systems*. IET, 1990, pp. 7–1.

[6] A. J. Orman, C. N. Potts, A. K. Shahani, and A. R. Moore, “Scheduling for a Multifunction Phased Array Radar System,” *European Journal of operational research*, vol. 90, no. 1, pp. 13–25, 1996.

[7] J. M. Butler, “Multi-Function Radar Tracking and Control,” *London, UK: University College London*, 1998.

[8] S. Miranda, C. Baker, K. Woodbridge, and H. Griffiths, “Knowledge-Based Resource Management for Multifunction Radar: a Look at Scheduling and Task Prioritization,” *IEEE Signal Processing Magazine*, vol. 23, no. 1, pp. 66–76, 2006.

[9] S. L. C. Miranda, C. J. Baker, K. Woodbridge, and H. D. Griffiths, “Simulation Methods for Prioritizing Tasks and Sectors of Surveillance in Phased Array Radar,” *International Journal of Simulation*, vol. 5, no. 1-2, pp. 18–25, 2004.

[10] “Multibeam Radar for Adaptive Search and Track,” https://it.mathworks.com/help/radar/ug/multibeam-radar-for-adaptive-search-and-track.html, Mathworks Matlab.

[11] A. Aubry, A. De Maio, and L. Pallotta, “A Priority-Based Scheduling Scheme for Search, Track, and Communications in MPARs,” *IEEE Transactions on Radar Systems*, vol. 2, pp. 471–481, 2024.

[12] M. A. Richards, J. Scheer, W. A. H., and W. L. Melvin, *Principles of Modern Radar*. Citeseer, 2010, vol. 1.

[13] H. A. P. Blom and Y. Bar-Shalom, “The Interacting Multiple Model Algorithm for Systems with Markovian Switching Coefficients,” *IEEE transactions on Automatic Control*, vol. 33, no. 8, pp. 780–783, 1988.

[14] Y. Bar-Shalom, P. K. Willett, and X. Tian, *Tracking and Data Fusion*. YBS publishing Storrs, CT, USA, 2011, vol. 11.

[15] A. Aubry, A. De Maio, and L. Pallotta, “Power-Aperture Resource Allocation for a MPAR with Communications Capabilities,” *IEEE Trans. on Vehicular Technologies*, 2024.