Journal of Systems Engineering and Electronics
Vol. 32, No. 5, October 2021, pp.1119 – 1129

Online adaptive dwell scheduling based on dynamic template for PAR

TAN Qianqian, CHENG Ting*, and LI Xi

School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China

Abstract: An adaptive dwell scheduling algorithm for phased array radar (PAR) is proposed in this paper. The concept of online dynamic template is introduced, based on which a general pulse interleaving technique for PAR is put forward. The pulse interleaving condition of the novel pulse interleaving is more intuitive and general. The traditional adaptive dwell scheduling algorithm combined with the general novel pulse interleaving technique results in the online adaptive dwell scheduling based on dynamic template for PAR is given. The proposed algorithm is suitable for radar tasks with multiple pulse repetition intervals (PRIs), which can be utilized in the actual radar system. For the purpose of further improving the scheduling efficiency, an efficient version is proposed. Simulation results demonstrate the effectiveness of the proposed algorithm and the efficient one. The proposed efficient algorithm can improve the time utilization ratio (TUR) by 9%, the hit value ratio (HVR) by 3.5%, and reduce the task drop ratio (TDR) by 6% in comparison with existing dwell scheduling algorithms considering pulse interleaving in PAR and the proposed efficient one.

Keywords: dynamic template, dwell scheduling, pulse interleaving.

DOI: 10.23919/JSEE.2021.000096

1. Introduction

Phased array radar (PAR) can switch its beam direction rapidly, so it has the ability of multi-function. The system resources are shared by different radar tasks, therefore the effective resource management algorithm is very important for PAR to make better use of its performance. Radar resource management involves prioritization [1], parameter selection [2−7], and scheduling [8−31]. We focus on the dwell scheduling problem in this paper.

The dwell scheduling based on the template is the pioneer scheduling method [8−10]. As the templates are often designed offline and fixed, they lack the flexibility and adaptability to the working environment of the radar system. The adaptive dwell scheduling algorithms [11−13] are more flexible and effective than the methods based on templates. The execution time of scheduled dwells in each scheduling interval is allocated according to the working priority of tasks in [11], which is adaptive to the system resource. In [12], the quadratic programming can obtain the best execution time of tasks. A new adaptive dwell scheduling algorithm was proposed in [13], where the concept of time pointer was introduced to make the task priority dynamic during scheduling. Based on time pointer analysis method, the proposed algorithm of [14] solves the problem of two-dimensional resource management in PAR. The analysis method based on time pointer was proposed to schedule tasks for the air-defense phased array radar in [15]. In previous adaptive dwell scheduling algorithms, radar task was regarded as a whole and cannot be preempted. For the purpose of further improving the utilization of radar system time and energy resources, the pulse interleaving technique was put forward in [16]. In [17], a novel adaptive dwell scheduling algorithm based on pulse interleaving technique was proposed by modifying the time pointer's sliding step in the algorithm of [13]. In [18], a pulse interleaving technique based on the analysis on the state of remaining resource was proposed. In [19], the pulse interleaving in phased array radar was considered which was realized through checking if three pulse overlapping conditions are not met. The pulse interleaving was realized by updating the remaining time pieces and checking if the transmitting and receiving durations can be executed in these pieces in [20]. Adaptive dwell scheduling method based on pulse interleaving for digital array radar (DAR) was proposed in [21], where the receiving duration of tasks can be overlapped in DAR. The start and end time of the receiving duration was used to analyze time con-

Manuscript received December 24, 2020.
*Corresponding author.
This work was supported by the National Natural Science Foundation of China (61032010).

straints of pulse interleaving in [22], which reduces the complexity of the interleaving analysis. In [23], a pulse interleaving technique based on the state analysis of the scheduling interval was proposed, which was combined with convential dwell scheduling algorithm to realize dwell scheduling for DAR. A simplified pulse interleaving method was given in [24] for DAR. However, the task model is not a realistic one, which limits the application of the method. The pulse interleaving that makes full use of transmitting, waiting and receiving durations of radar dwells were proposed in MIMO radars in [25,26]. This paper focuses on the dwell scheduling for PAR.

Based on the works above, it can be seen that

(i) Adaptive dwell scheduling methods and the ones based on templates are regarded to be independent. The combination of them has never been considered so far.

(ii) In existing pulse interleaving methods, only the dwell tasks with the same pulse repetition interval (PRI) and PRI number can be interleaved [21,25], or the tasks with only one single PRI are interleaved [22,27−30]. In practice, multiple PRIs are contained in actual radar task to obtain the desired signal to noise ratio (SNR).

Therefore, the concept of online dynamic template is introduced, which represents the occupied time and energy situation of the remaining time in a scheduling interval (SI). Combined with the traditional adaptive dwell scheduling algorithm, an online adaptive dwell scheduling algorithm based on dynamic template for PAR is put forward, whose contributions include

(i) The proposed algorithm is a combination of the adaptive dwell scheduling algorithm and online dynamic template.

(ii) The novel pulse interleaving based on online dynamic template is put forward, whose conditions of successful interleaving are more intuitive.

(iii) Because the proposed algorithm is based on the radar task model with multiple PRIs, it can be utilized in actual radar.

(iv) Based on the proposed algorithm, an efficient version is developed to improve the execution efficiency of the algorithm.

The rest of the paper is organized as follows: Section 2 describes the model formulation of dwell scheduling. Section 3 puts forward the online adaptive dwell scheduling algorithm based on dynamic template for PAR. Section 4 develops an efficient version of the proposed one in Section 3. The simulation results are shown in Section 5, and the conclusions of this paper are drawn in Section 6.

2. Model formulation of dwell scheduling in PAR

To obtain desired SNR, multiple PRIs are included in a radar task. In each PRI, the system transmits, waits, and receives the echoes, which can be depicted in Fig. 1.

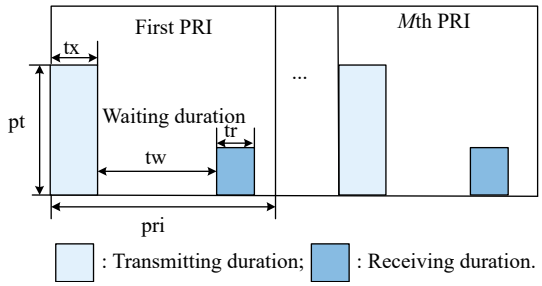


Fig. 1 Radar task model

Therefore, construct the model of Task i as follows:

$$T_i = \{w_i, \mathrm{rt}_i, d_i, l_i, \mathrm{tx}_i, \mathrm{tw}_i, \mathrm{tr}_i, M_i, \mathrm{pri}_i, \mathrm{Pt}_i\}. \tag{1}$$

The parameters of the radar task model are described in Table 1, where the calculation of other parameters of the task can obtain the deadline in [31]. According to the expected execution time and the time window, we can get the earliest and latest execution time of a task, which are $\mathrm{rt}_i - l_i$ and $\mathrm{rt}_i + l_i$ respectively.

Table 1 Description of the radar task model

Parameter	Description
w_i	The working priority
d_i	The deadline
rt_i	The expected execution time
Pt_i	The transmitting power
l_i	The time window
M_i	The number of PRIs
pri_i	The PRI
tx_i	The transmitting duration
tw_i	The waiting duration
tr_i	The receiving duration

In the process of dwell scheduling, firstly the radar time resource constraints should be satisfied, including:

(i) All scheduled tasks must be completed before the deadline.

(ii) The transmitting durations and receiving durations of these tasks do not overlap with each other.

When pulse interleaving is considered during scheduling, the duration of transmitting is prolonged. For example, in Fig. 2 which shows that the interleaving in the first PRI of Task0 and Task1, $\mathrm{tx}_0 + \mathrm{tx}_1$ is the total transmitting duration length of radar system after Task0 and Task1 interleaving. Therefore, the energy constraint should be considered:

$$E(t) \leqslant E_{\text{th}} \tag{2}$$

where $E(t)$ is the radar system energy consumption at time t and can be calculated as follows:

$$E(t) = \int_0^t P(x)\mathrm{e}^{(x-t)/\tau}\mathrm{d}x \tag{3}$$

where τ is the look-back period and $P(x)$ is the power function. E_{th} is a threshold of the maximum energy consumption of the system.

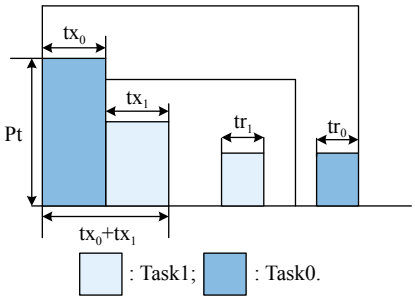


Fig. 2 Interleaving of two tasks in the first PRI

Note that the tasks with different PRIs and PRI numbers may be interleaved in this paper. It is more general compared with existing pulse interleaving techniques, which will be explained in detail in Section 3.

In the process of arranging the task scheduling sequence, the priority and deadline of the radar task is considered comprehensively. Assume M radar tasks are requested to be scheduled at the moment, the synthetic priority sw_i for T_i is calculated as follows [13]:

$$\mathrm{sw}_i = ((M+2-\eta)\cdot Nd_i + \eta\cdot \mathrm{Np}_i)/(M+1) \tag{4}$$

where Np_i and Nd_i are the serial number of T_i in the task request queues arranged according to the working mode priority and deadline respectively. η is a controllable parameter.

Assume there are N dwell tasks applied to be executed in current SI. Based on above synthetical priority and constraints, the dwell scheduling optimization model is given:

$$\max_{N_1,N_2,N_3} \sum_{i=1}^{N_1} \mathrm{sw}_i$$

$$\text{s.t.}\begin{cases} \max(\mathrm{rt}_i - l_i, t_0) \leqslant \mathrm{et}_i < \min(\mathrm{rt}_i + l_i, t_{\text{end}}),\ \ i = 1,2,\cdots,N_1 \\ \bigcap\limits_{i=1}^{N_1}\left\{\bigcup\limits_{j=1}^{M_i}[\mathrm{et}_i + (j-1)\cdot \mathrm{pri}_i, \mathrm{et}_i + (j-1)\cdot \mathrm{pri}_i + \mathrm{tx}_i]\right\} = \varnothing,\ \ i = 1,2,\cdots,N_1 \\ \bigcap\limits_{i=1}^{N_1}\left\{\bigcup\limits_{j=1}^{M_i}[\mathrm{et}_i + (j-1)\cdot \mathrm{pri}_i + \mathrm{tx}_i + \mathrm{tw}_i, \mathrm{et}_i + (j-1)\cdot \mathrm{pri}_i + \mathrm{tx}_i + \mathrm{tw}_i + \mathrm{tr}_i]\right\} = \varnothing,\ \ i = 1,2,\cdots,N_1 \\ \left\{\bigcup\limits_{j=1}^{M_i}[\mathrm{et}_i + (j-1)\cdot \mathrm{pri}_i, \mathrm{et}_i + (j-1)\cdot \mathrm{pri}_i + \mathrm{tx}_i]\right\} \cap \\ \qquad \left\{\bigcup\limits_{j=1}^{M_k}[\mathrm{et}_k + (j-1)\cdot \mathrm{pri}_k + \mathrm{tx}_k + \mathrm{tw}_k, \mathrm{et}_k + (j-1)\cdot \mathrm{pri}_k + \mathrm{tx}_k + \mathrm{tw}_k + \mathrm{tr}_k]\right\} = \varnothing,\ \ \forall i,k \in [1,N_1] \\ E(t) \leqslant E_{\text{th}},\ \ t \in [t_0, t_{\text{end}}) \\ \mathrm{rt}_u + l_u \geqslant t_{\text{end}},\ \ u = 1,2,\cdots,N_2 \\ \mathrm{rt}_v + l_v < t_{\text{end}},\ \ v = 1,2,\cdots,N_3 \end{cases} \tag{5}$$

where t_0 is the beginning time of the SI and t_{end} is the end of the SI. et_i is the actual execution time of T_i. Obviously, $N_1 + N_2 + N_3 = N$. And the task numbers of scheduled, delayed and deleted sequence are N_1, N_2, and N_3 respectively.

There are seven constraints in the above dwell scheduling optimization model. The first one means that the actual execution time allocated to each task should be within its executable time range. The second to the fourth constraints correspond to no interrupt during transmitting and receiving. The energy constraint is shown in the fifth inequality. The conditions for delayed tasks and deleted tasks are described by the last two inequalities in the dwell scheduling optimization model respectively.

3. Online adaptive dwell scheduling based on dynamic template for PAR

3.1 Introduction of online dynamic template

The conventional adaptive dwell scheduling algorithm in [13] is chosen as the basic dwell scheduling algorithm as its superiority over other conventional adaptive dwell scheduling algorithms. It includes the following steps:

Step 1 Initialize the time pointer $\mathrm{tp}(\mathrm{tp} \geqslant t_0)$ of the scheduling interval, and $i = 0$.

Step 2 Select the task requests meeting that tp is larger than the latest execution time of them. Assume the number of selected tasks is n, then delete them and $i = i + n$.

Step 3 Choose the task requests satisfying that tp is larger than the earliest execution time of them. And the

synthetic priorities of them are calculated according to (4).

Step 4 Choose the task request with the highest synthetic priority to be the scheduled one, and denote it as T_j.

Step 5 Update $\text{tp} = \text{tp} + \text{pri}_j \times M_j$ and $i = i + 1$. If $i > N$ or $\text{tp} > t_0 + L_{\text{SI}}$, the process of scheduling ends, otherwise go to Step 2.

It can be seen in Step 5 that once a task is scheduled, the time pointer will slide the length of its dwell time as shown in Fig. 3. Therefore, the waiting duration of this task is not utilized.

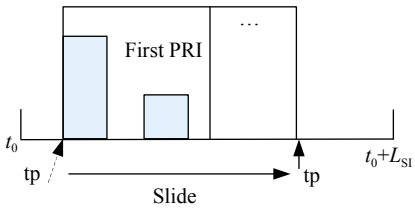


Fig. 3 Sliding process of the time pointer

As shown in Fig. 3, it can be seen that the time pointer continuously slides in the scheduling interval and the left resource in the SI is the one during the time pointer and the ending time of this SI. Based on the sliding time pointer, a dynamic template can be constructed, which can be described as follows:

$$B = \{\text{tp}, t_0 + L_{\text{SI}}, \Delta t, \boldsymbol{S}, \boldsymbol{E}\} \tag{6}$$

where tp is the start time of template B and $t_0 + L_{\text{SI}}$ is the end time of template B. L_{SI} is the length of the current SI. Δt is the length of the slot in the template B. The time resource occupation, energy consumption of template B are described by the vectors $\boldsymbol{S}$ and $\boldsymbol{E}$. Fig. 4 shows the illustration of the template. The length of $\boldsymbol{S}$ and $\boldsymbol{E}$ is

$$n_{\text{tp}} = \left\lceil \frac{(t_0 + L_{\text{SI}}) - \text{tp}}{\Delta t} \right\rceil \tag{7}$$

where rounding up is denoted by $\lceil \cdot \rceil$.

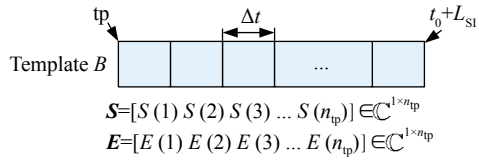


Fig. 4 Characterizations of dynamic template

It can be seen that $[\text{tp}, t_0 + L_{\text{SI}}]$ is discretized to form the template. The occupancy slots in B by all scheduled tasks is reflected by $\boldsymbol{S}$. The consumed energy after each slot in B is reflected by $\boldsymbol{E}$.

3.2 A general pulse interleaving analysis based on dynamic template

When developing the dwell scheduling algorithm for PAR, this paper fully considers the pulse interleaving technique. Based on dynamic template B, a general pulse interleaving analysis method can be designed. Suppose task T_1 is scheduled, then analyze if the system can schedule T_2 at tp according to the following way:

(i) The state variation vectors caused by T_2 are calculated. When the system schedules task T_2 at tp, the situation of time and energy resource in template B will be changed where $\Delta\boldsymbol{S}$ and $\Delta\boldsymbol{E}$ describe variations in time and energy resource consumption caused. $\Delta\boldsymbol{S}$ and $\Delta\boldsymbol{E}$ are calculated as follows:

$$\Delta S(j) = \begin{cases} 1, & j \in \bigcup_{k=1}^{M} \left(\left(\frac{(k-1)\text{pri}}{\Delta t} + 1, \frac{(k-1)\text{pri} + \text{tx}}{\Delta t} \right] \cup \right. \\ & \left. \left(\frac{(k-1)\text{pri} + \text{tx} + \text{tw}}{\Delta t} + 1, \frac{(k-1)\text{pri} + \text{tx} + \text{tw} + \text{tr}}{\Delta t} \right] \right), \\ 0, & \text{else} \end{cases} \tag{8}$$

$$\Delta E(j) = \sum_{k=1}^{M} \Delta E_k(j), \tag{9}$$

$$\Delta E_k(j) = \begin{cases} \text{Pt}\,\tau \left[1 - \exp\left(\frac{(k-1)\text{pri} - j\Delta t}{\tau} \right) \right], & j \in \left(\frac{(k-1)\text{pri}}{\Delta t} + 1, \frac{(k-1)\text{pri} + \text{tx}}{\Delta t} \right] \\ \Delta E_k\left(\frac{(k-1)\text{pri} + \text{tx}}{\Delta t} \right) \exp\left(\frac{(k-1)\text{pri} + \text{tx} - j\Delta t}{\tau} \right), & j \in \left(\frac{(k-1)\text{pri} + \text{tx}}{\Delta t} + 1, n_{\text{tp}} \right] \\ 0, & \text{else} \end{cases}. \tag{10}$$

In (9), the energy state variation of the jth slot in the dynamic template B is denoted as $\Delta E_k(j)$, which is caused by the kth PRI of task T.

(ii) The constrains in dynamic template B are judged. Check whether the inequalities (11) and (12) meet the following time and energy constraints:

$$\max(\boldsymbol{S} + \Delta\boldsymbol{S}) \leqslant 1, \tag{11}$$

$$\max(\boldsymbol{E} + \Delta\boldsymbol{E}) \leqslant E_{\text{th}}, \tag{12}$$

where (11) denotes that task T_2 and other scheduled tasks will not interrupt with each other during execution. Equa-

tion (12) indicates that the energy by consumed of dynamic template B will not exceed the threshold, which corresponds to the fifth constraint in (5). If (11) and (12) are met, the system can schedule T_2 at tp. The parameters $\boldsymbol{S}$ and $\boldsymbol{E}$ of the template will be updated. Fig. 5 shows the detailed $\Delta\boldsymbol{S}$ and $\Delta\boldsymbol{E}$. Obviously, the two tasks have different PRIs and the number of PRIs.

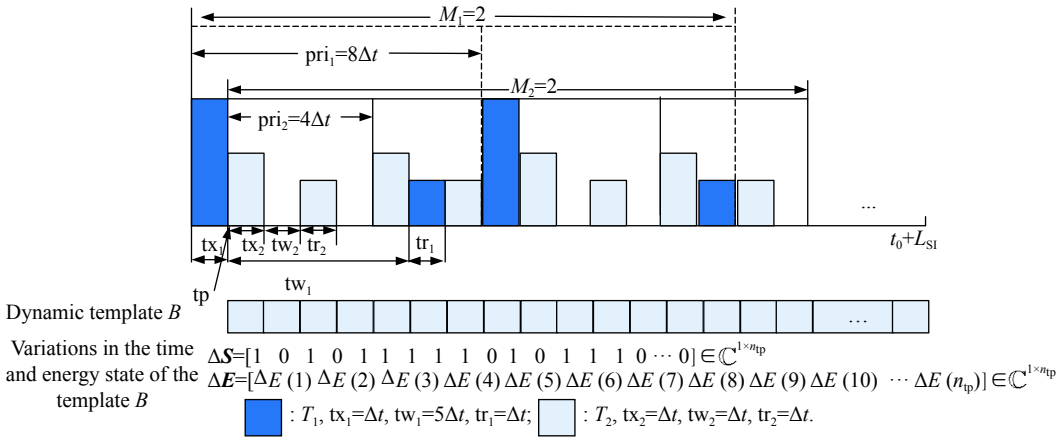


Fig. 5 Analysis process of pulse interleaving in dynamic template

Based on the dynamic template, the pulse interleaving involved in the above scheduling analysis method is easier and general.

3.3 Online adaptive dwell scheduling algorithm based on dynamic template

Combing the adaptive dwell scheduling algorithm in [13] and the above novel online pulse interleaving method, the online adaptive scheduling algorithm based on dynamic template is obtained. Assume during the current SI $[t_0, t_0+L_{SI}]$, N dwell tasks applied to be scheduled are denoted as $\boldsymbol{T}=\{T_1, T_2, \cdots, T_N\}$. The steps of the online adaptive scheduling algorithm based on dynamic template are as follows:

Step 1 Dynamic template B is initialized. The start time of initial template is t_0 and the end time of initial template is t_0+L_{SI}. Let $i=0$. The time pointer tp and the state vectors $\boldsymbol{S}$ and $\boldsymbol{E}$ of initial template are set as follows:

$$\text{tp}=t_0, \tag{13}$$

$$\boldsymbol{S}=\boldsymbol{0}\in\mathbb{C}^{1\times n_{tp}}, \tag{14}$$

$$\boldsymbol{E}=\boldsymbol{0}\in\mathbb{C}^{1\times n_{tp}}, \tag{15}$$

where n_{tp} is calculated by (7).

Step 2 Select the task requests meeting that tp is larger than the latest execution time of them. Assume the number of selected tasks is n, then delete them and $i=i+n$.

Step 3 Choose the task requests satisfying that tp is larger than the earliest execution time of them. And the synthetic priorities of them are calculated according to (4). Denote the task with the highest synthetic priority as T_{test} and judge whether T_{test} is scheduled at tp in the dynamic template B according to Section 3.2.

Step 4 If T_{test} can be scheduled at tp, parameters are updated as follows:

$$\Delta\text{tp}=\text{tx}, \tag{16}$$

$$i=i+1, \tag{17}$$

where tx is the transmitting duration of T_{test}. Otherwise, $\Delta\text{tp}=\Delta t$.

Step 5 Let $\text{tp}=\text{tp}+\Delta\text{tp}$, other parameters in the dynamic template B are updated as follows:

$$\boldsymbol{S}=\boldsymbol{S}+\Delta\boldsymbol{S}, \tag{18}$$

$$\boldsymbol{E}=\boldsymbol{E}+\Delta\boldsymbol{E}, \tag{19}$$

$$\boldsymbol{S}=\boldsymbol{S}\left[\frac{\Delta\text{tp}}{\Delta t}+1\text{:end}\right], \tag{20}$$

$$\boldsymbol{E}=\boldsymbol{E}\left[\frac{\Delta\text{tp}}{\Delta t}+1\text{:end}\right]. \tag{21}$$

Step 6 If $i>N$ or $\text{tp}>t_0+L_{SI}$, the analysis process of task scheduling ends, otherwise go to Step 2.

In the proposed algorithm, the complexity is mainly based on (18) and (19). The time pointer slides a time slot Δt each time. At the beginning of SI, the lengths of $\boldsymbol{S}$ and $\boldsymbol{E}$ are $\dfrac{L_{SI}}{\Delta t}$ and the calculation of (18) and (19) involves $2\times\dfrac{L_{SI}}{\Delta t}$ times summation. Then it involves $2\times\left[\dfrac{L_{SI}}{\Delta t}-1\right]$

times summation in the next step and so on. Therefore, in general, the complexity of the algorithm can be described as $2\times\sum_{i=0}^{\frac{L_{\mathrm{SI}}}{\Delta t}-1}\left(\frac{L_{\mathrm{SI}}}{\Delta t}-i\right)=\left(\frac{L_{\mathrm{SI}}}{\Delta t}\right)^2+\frac{L_{\mathrm{SI}}}{\Delta t}$.

4. An efficient version of online adaptive dwell scheduling algorithm based on dynamic template for phased array radar

In the proposed algorithm, the time pointer slides according to the length of the transmitting duration of the scheduled task. However, if the occupancy of the dynamic template B is already high, then the time pointer can slide further to improve the execution efficiency of the proposed algorithm. In order to evaluate the occupancy of B, the time utilization ratio of a period after tp is considered.

As shown in Fig. 6, the positions of the red lines are between the continuously occupied time slot and the next unoccupied time slot adjacent to it in the dynamic template B. They are described by vector $\boldsymbol{P}$ as follows:

$$\boldsymbol{P}=[P(1),P(2),\cdots,P(m)] \tag{22}$$

where m is the number of red lines. For example, $P(1)=2,P(2)=5,P(3)=8$ in Fig. 6. The utilization ratio corresponds to $\boldsymbol{P}$ is denoted as

$$\boldsymbol{U}=[u(1),u(2),\cdots,u(m)] \tag{23}$$

where $u(j)$ is the utilization ratio of $[\mathrm{tp},\mathrm{tp}+P(j)\times\Delta t]$, $j=1,2,\cdots,m$. The index j^* of the first element in vector $\boldsymbol{U}$ that makes the following inequality hold is found:

$$\begin{cases}u(j^*+1)<r_{\mathrm{th}}, & j>j^*\\ u(j)\geqslant r_{\mathrm{th}}, & j\leqslant j^*\end{cases} \tag{24}$$

where r_{th} is an expected utilization ratio. The sliding length of the time pointer is updated accordingly as $\Delta\mathrm{tp}=P(j^*)\times\Delta t$. Especially, $\Delta\mathrm{tp}=\mathrm{tx}$ when $u(1)<r_{\mathrm{th}}$.

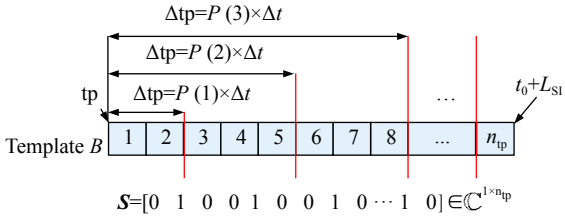


Fig. 6 Illustration of the red lines in the dynamic template

Fig. 7 describes the flow chart of the efficient algorithm, where Step 4 of the original proposed algorithm in Section 3.3 is replaced by Steps 4A−4D.

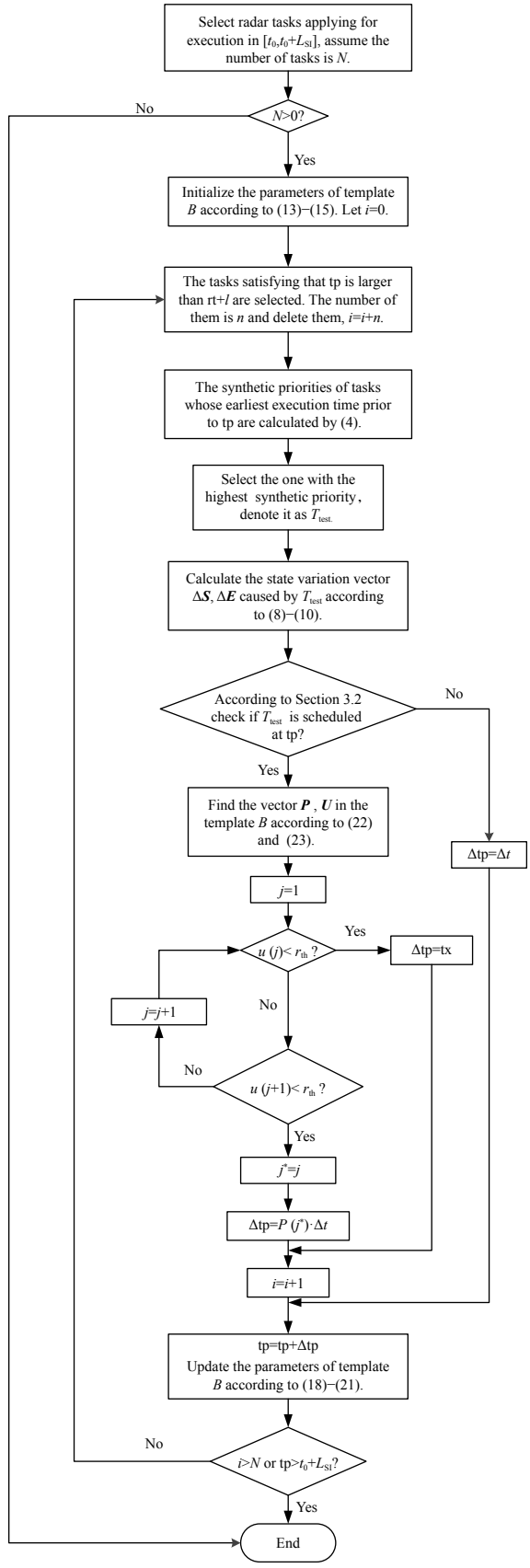


Fig. 7 Flow chart of the efficient version

Step 4A If T_{test} can be scheduled at tp, go to Step 4B. Otherwise, $\Delta\text{tp} = \text{tx}$ and $i = i+1$, go to Step 5 in Section 3.3.

Step 4B Find the vector $\boldsymbol{P}$ in the dynamic template B according to (22) and calculate the utilization ratio of $[\text{tp}, \text{tp}+P(j)\times\Delta t], j = 1,2,\cdots,m$ to form vector $\boldsymbol{U}$. If $u(1) < r_{\text{th}}$, $i = i+1$, update Δtp according to (16) and go to Step 5. Otherwise, go to Step 4C.

Step 4C Find the index j^* of the first element in vector $\boldsymbol{U}$ that satisfies (24).

Step 4D Update the sliding length as follows:

$$\Delta\text{tp} = P(j^*)\cdot\Delta t. \tag{25}$$

It can be seen that in the efficient version of the proposed algorithm, the sliding step can be adaptive with the utilization ratio of the template. If it has already been fully used, the sliding step is relatively large. Otherwise, it slides as before. It is specially noted that if r_{th} is selected as 100%, the efficient algorithm is the original one proposed in Section 3.3. In the efficient algorithm, the smaller is r_{th}, the easier is the time pointer slides forward. Therefore, the complexity of the efficient version is proportional to $(1-r_{\text{th}})\left[\left(\frac{L_{\text{SI}}}{\Delta t}\right)^2 + \frac{L_{\text{SI}}}{\Delta t}\right]$.

5. Simulation results

Considering horizon searching, airspace searching, precise tracking, normal tracking and confirmation tasks in the simulation scene, set the whole simulation time as 4 s, $\text{SI} = 4$ ms, $E_{\text{th}} = 10$ J, $\tau = 200$ ms and $\Delta t = 0.5$ ms. The searching task requires many beams to complete the searching of a given area, therefore, the dwell number is more than 1. For the confirmation task, the position where there is possibly a target should be illuminated. It is a random event, and usually involves one dwell. For the tracking task, at each sampling moment, the beam should be transmitted towards the predicted position of the target, and only one dwell is included. The detailed radar task parameters are given in Table 2 [13]. Increasing the target number from 0 to 100, the ratio of targets with precise tracking and ones with normal tracking is set as 1:4. The performances indices of the proposed algorithm are compared with two existing dwell scheduling algorithms, which are the algorithm in [17] (Algorithm A) and the one in [13] (Algorithm B).

Table 2 Parameters of radar tasks

Task type	w	Dwell number	Period/ms	L/ms	Pt/kw	tx/ms	tw/ms	tr/ms	Pri/ms	M
Confirmation	5	1	–	15	5	0.5	2.5	0.5	4	2
Precise tracking	4	1	500	15	4	0.5	1	0.5	2.5	2
Normal tracking	3	1	100	25	3	0.5	2.5	0.5	4	2
Horizon search	2	100	2000	–	4	0.5	–	2	3	4
Airspace search	1	150	4000	–	3	0.5	–	1.5	2.5	4

(i) Task drop ratio (TDR): it is defined as the ratio between the number of radar tasks deleted and the total number of tasks requested to be executed. It can be calculated as

$$\text{TDR} = \frac{N_{\text{lose}}}{N_{\text{total}}} \tag{26}$$

where N_{total} is the total number of tasks requested to be executed, N_{lose} is the number of radar tasks deleted.

(ii) Time utilization ratio (TUR): it is defined as the ratio between the total time of the transmitting and receiving durations of radar tasks scheduled and the total simulation time. It can be calculated as

$$\text{TUR} = \frac{\sum_{i=1}^{N_{\text{suc}}}(\text{tx}_i + \text{tr}_i)}{T_{\text{total}}} \tag{27}$$

where T_{total} is the simulation time.

(iii) Hit value ratio (HVR): it is defined as the ratio of the hit value of radar tasks scheduled to that of all tasks requested to be scheduled. It can be expressed as

$$\text{HVR} = \sum_{i=1}^{N_{\text{suc}}} w_i \Big/ \sum_{i=1}^{N_{\text{total}}} w_i \tag{28}$$

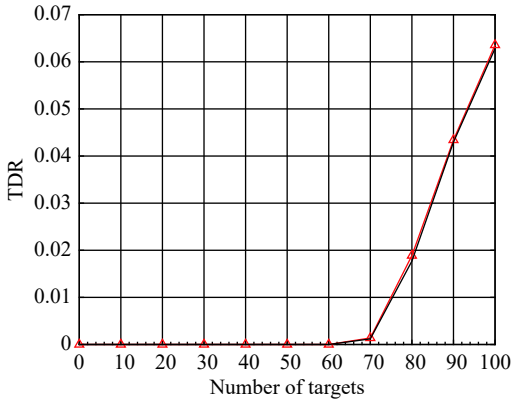
where N_{suc} is the number of successfully scheduled tasks.

Fig. 8 to Fig. 13 show the average results of 100 Monte Carlo simulations.

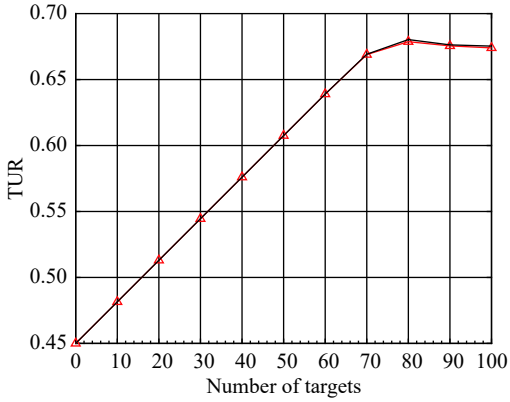
When using the proposed algorithm, parameter r_{th} should be given firstly. Consider r_{th} is set to be 0.25, 0.5, 0.75 and 1.00.

Firstly, the performances of the original proposed algorithm and the efficient one with $r_{\text{th}} = 1$ are compared. Fig. 8(a) shows the comparison of TDRs. Fig. 8(b) shows the comparison of TURs. Fig. 8(c) shows the comparison of HVRs. It can be seen that they have similar perform-

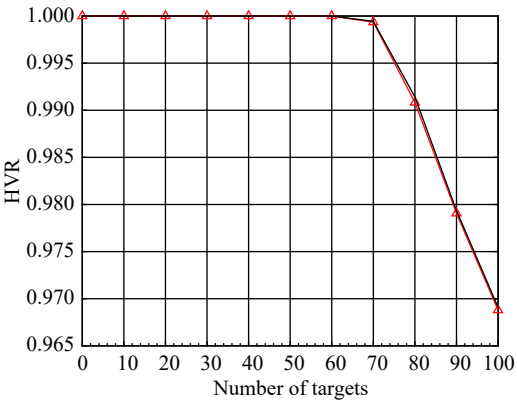
ances. The reason is that when the threshold is 1, the effect is equivalent to sliding the time pointer Δt each time. Obviously, the TDRs, HVRs and TURs of two algorithms are almost the same. Fig. 9 shows the cost time comparison. The dwell scheduling efficiency is slightly improved by the efficient version with $r_{\text{th}} = 1$. Therefore, to obtain obvious efficiency improvement, r_{th} should be decreased further.



(a) Comparison of TDRs



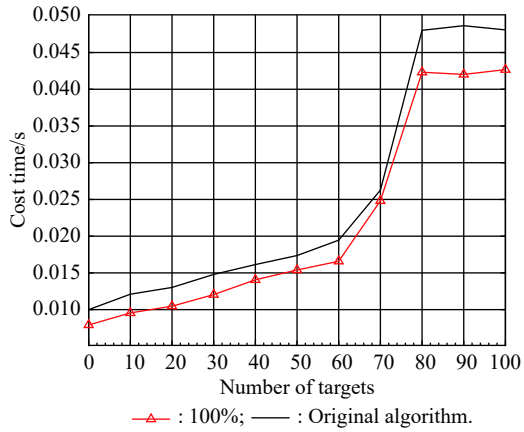
(b) Comparison of TURs



(c) Comparison of HVRs

—△— : 100%; —— : Original algorithm.

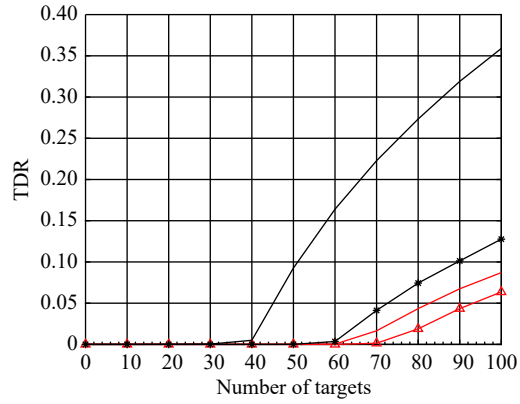
Fig. 8 Comparison of the performances of the original proposed algorithm and the efficient one with $r_{\text{th}} = 1$



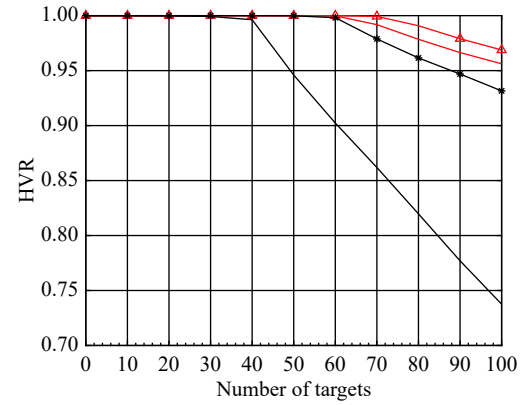
—△— : 100%; —— : Original algorithm.

Fig. 9 Comparison of cost time of the original proposed algorithm and the efficient one with $r_{\text{th}} = 1$

Fig. 10 shows the comparison results with different r_{th} in the proposed algorithm. Obviously, the HVR declines more slowly and the TDR rises faster as r_{th} increases from 0.25 to 1.00. However, when r_{th} increases from 0.25 to 1.00, more time is spent by the scheduling process. The introduction of r_{th} can save the cost time compared with the original proposed algorithm. Moreover, with the increase of r_{th}, TDR and HVR will become closer to that with $r_{\text{th}} = 1$. Consider the TDR, the HVR, and cost time comprehensively, r_{th} is selected as 0.75.



(a) TDRs in different r_{th} of the proposed algorithm



(b) HVRs in different r_{th} of the proposed algorithm

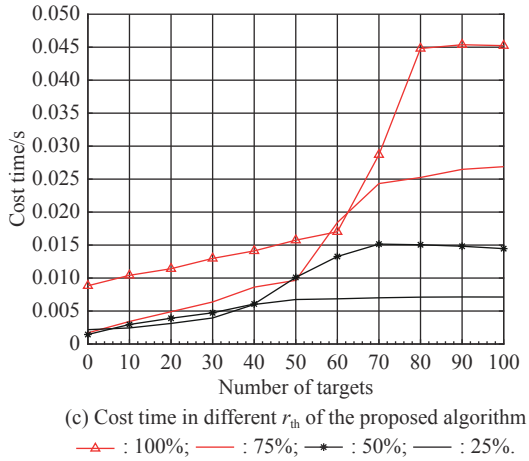


Fig. 10 Comparison of the proposed algorithm in different r_{th}

Fig. 11 shows the comparison of TDRs. The TDR of Algorithm B rises rapidly with the increase of the target number when $N = 40$. This is because the pulse interleaving technology is not used in Algorithm B. Therefore, the radar system wastes the waiting durations of tasks. Although the pulse interleaving technology is introduced to Algorithm A and the proposed one, the TDR of Algorithm A rises faster compared with the proposed one. When the targets number reaches 50, the radar tasks start to be dropped in Algorithm A. However, the target number increases as 60, the proposed one begins to lose radar tasks. That is because the tasks with different PRIs and PRI number may be interleaved in the dynamic template B. The proposed algorithm breaks the strict conditions of Algorithm A which only allows the tasks with the same PRI and PRI number to be interleaved.

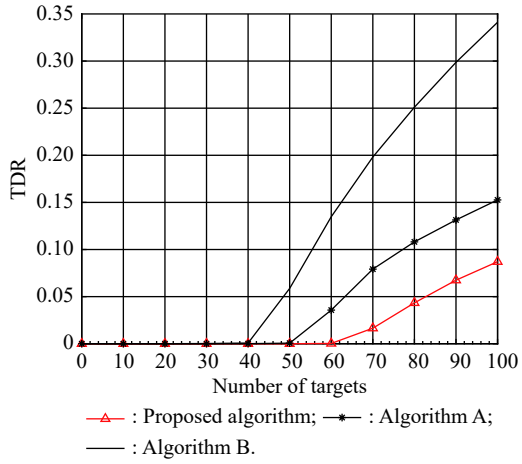


Fig. 11 Comparison of TDRs

Fig. 12 compares the TURs of these algorithms. Because the waiting durations of radar tasks are neglected by Algorithm B, the TUR of it is much lower than the proposed algorithm and Algorithm A. Increasing of the number of targets, radar resources are insufficient for scheduling more tasks in Algorithm B, which leads to the losing of more and more tasks and the TUR of Algorithm B decreases quickly after the targets number reaches 40. Because of too strict conditions of pulse interleaving in Algorithm A, the interleaved tasks are fewer than the proposed algorithm. Therefore, the proposed algorithm can schedule more tasks and the TUR of it is much higher than Algorithm A after the targets number reaches 50.

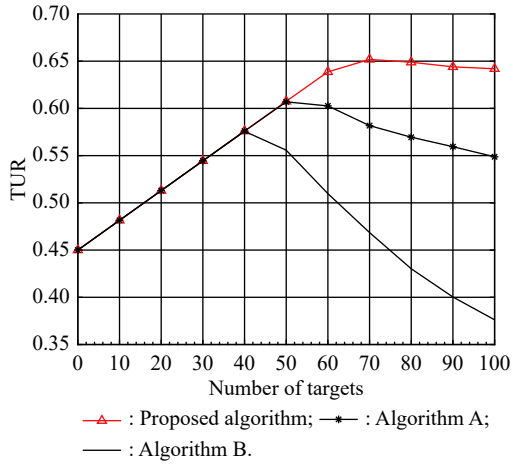


Fig. 12 Comparison of TURs

Fig. 13 compares the HVR of these algorithms. Before the targets number reaches 40, the HVRs of Algorithm B, Algorithm A and the proposed one are same as 1. The HVR of Algorithm B is the first to drop from 1. Then the HVR of Algorithm A begins to decrease correspondingly when the targets number reaches 50. Until the number of targets reaches 60, the HVR of the proposed algorithm starts to change, the decreasing trend of which is slower compared with other two algorithms.

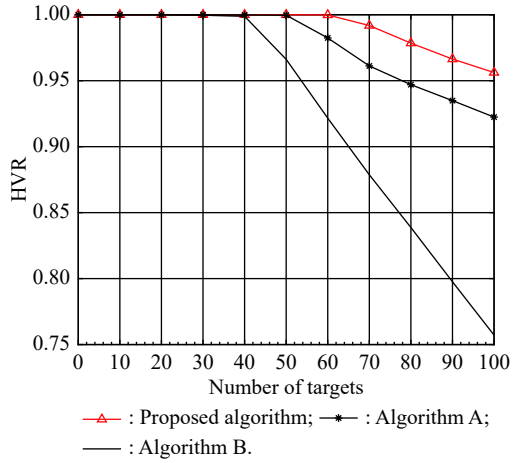


Fig. 13 Comparison of HVRs

Therefore, compared with other existing pulse interleaving algorithms in PAR, the TDR is significantly re-

duced by 6% and the TUR and the HVR are improved by 9% and 3.5% respectively in the proposed algorithm. Furthermore, the value of r_{th} is recommended to be 75%, which can balance the scheduling performance and efficiency effectively.

6. Conclusions

For the purpose of fully exerting the performance of PAR, it is necessary to manage its limited resources effectively. And it is the key point to design the dwell scheduling algorithm in PAR. The concept of online dynamic template is introduced, based on which the novel online pulse interleaving technique is proposed. Compared with existing pulse interleaving methods, it is easier and more general to realize pulse interleaving between tasks. Combining the existing dwell scheduling method and the novel pulse interleaving technique, an online adaptive dwell scheduling algorithm based on dynamic template for PAR is put forward. Moreover, in order to improve the execution efficiency of the proposed algorithm, an efficient version is developed. The simulation results show that the algorithm can effectively improve the TDR, TUR, and HVR. In addition, the scheduling performance of the efficient version of the proposed algorithm can be further improved.

References

[1] LU X, CHENG T. A dwell scheduling method for phased array radars based on new synthetic priority. Proc. of the 21st International Conference on Information Fusion, 2018: 1779−1785.

[2] CHENG T, LI S Y, ZHANG J. Adaptive resource management in multiple targets tracking for co-located multiple input multiple output radar. IET Radar, Sonar & Navigation, 2018, 12(9): 1038–1045.

[3] YUAN Y, YI W, KONG L J, et al. An adaptive resource allocation strategy for multiple target tracking with different performance requirements. Proc. of the International Conference on Radar, 2018: 874–879.

[4] YUAN Y, YI W, KIRUBARAJAN T, et al. Scaled accuracy based power allocation for multi-target tracking with colocated MIMO radars. Signal Processing, 2019, 158: 227–240.

[5] ZHANG H W, XIE J W, SHI J P, et al. Joint beam and waveform selection for the MIMO radar target tracking. Signal Processing, 2019, 156: 31–40.

[6] LI X, CHENG T, SU Y, et al. Joint time-space resource allocation and waveform selection for the collocated MIMO radar in multiple targets tracking. Signal Processing, 2020, 176: 107650.

[7] SU Y, CHENG T, HE Z, et al. Adaptive simultaneous multibeam resource management for colocated MIMO radar in multiple targets tracking. Signal Processing, 2020, 172: 107543.

[8] SHIH C S, GOPALAKRISHNAN S, GANTI P, et al. Template-based real-time dwell scheduling with energy constraint. Proc. of the IEEE 9th Real-Time and Embedded Technology and Applications Symposium, 2003: 19−27.

[9] SHIH C S, GOPALAKRISHNAN S, GANTI P, et al. Scheduling real-time dwells using tasks with synthetic periods. Proc. of the IEEE 24th Real-Time Systems Symposium, 2003: 210−219.

[10] SHIH C S, GANTI P, GOPALAKRISHNAN S, et al. Synthesizing task periods for dwells in multi-function phased array radars. Proc. of the IEEE Radar Conference, 2004: 145−150.

[11] ZENG G, LU J B, HU W D. Research on adaptive scheduling algorithm for multifunction phased array radar. Modern Radar, 2004, 26(6): 14–18. (in Chinese)

[12] LU J B, HU W D, YU W X. Adaptive scheduling algorithm for real-time dwells in multifunction phased array radars. Systems Engineering and Electronics, 2005, 27(12): 1981–1987. (in Chinese)

[13] LU J B, HU W D, YU W X. Study on real-time task scheduling of multifunction phased array radars. Acta Electronica Sinica, 2006, 34(4): 732–736. (in Chinese)

[14] TIAN T F, ZHANG Q, CHEN Y J, et al. A task scheduling algorithm for multifunctional radar based on two-dimensional resource management. Acta Aeronautica et Astronautica Sinica, 2018, 39(12): 295–305. (in Chinese)

[15] ZHANG H, XIE J, ZONG B, et al. Dynamic priority scheduling method for the air-defence phased array radar. IET Radar, Sonar & Navigation, 2017, 11(7): 1140–1146.

[16] FARINA A, NERI P. Multitarget interleaved tracking for phased-array radar. IEE Proceedings F−Communications, Radar and Signal Processing, 1980, 127(4): 312−318.

[17] CHENG T, HE Z S, TANG T. Novel radar dwell scheduling algorithm based on pulse interleaving leaving. Journal of Systems Engineering and Electronics, 2009, 20(2): 247–253.

[18] TAN Q Q, CHENG T, LI X. Adaptive dwell scheduling based on a novel online pulse interleaving technique for phased array radar. Proc. of the International Conference on Control, Automation and Information Sciences, 2019. DOI: 10.1109/ICCAIS46529.2019.9074634.

[19] XIE X X, ZHANG W, CHEN J, et al. A time pointer-based on-line pulse interleaving algorithm for phased array radar. Journal of Systems Engineering and Electronics, 2013, 11(2): 185–191.

[20] ZHANG H, XIE J, GE J, et al. Task interleaving scheduling for phased array radar in multi-target tracking. Proc. of the IEEE 4th International Conference on Control Science and Systems Engineering, 2018: 347−350.

[21] CHENG T, HE Z S, LI H Y. Adaptive dwell scheduling for digital array radar based on online pulse interleaving. Chinese Journal of Electronics, 2009, 18(3): 574–578.

[22] ZHANG H W, XIE J W, ZHANG Z J, et al. Pulse interleaving scheduling algorithm for digital array radar. Journal of Systems Engineering and Electronics, 2018, 29(1): 67–73.

[23] LU X Y, CHENG T, PENG H, et al. Novel adaptive dwell scheduling algorithm for digital array radar based on pulse interleaving. Proc. of the 22th International Conference on Information Fusion, 2019: 9011216.

[24] ZHANG H W, XIE J W, ZHANG Z J, et al. Online task interleaving scheduling for the digital array radar. International Journal of Electronics and Communications, 2017, 79(9): 250–256.

[25] CHENG T, LIAO W, HE Z. MIMO radar dwell scheduling based on novel pulse interleaving technique. Journal of Systems Engineering and Electronics, 2013, 24(2): 234–241.

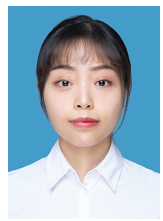
[26] ZHANG H W, XIE J W, GE J A, et al. Optimization model

and online task interleaving scheduling algorithm for MIMO radar. Computers & Industrial Engineering, 2018, 127: 865–874.
[27] ZHANG H W, XIE J W, GE J A, et al. An entropy-based PSO for DAR task scheduling problem. Applied Soft Computing, 2018, 73: 862–873.
[28] ZHANG H W, XIE J W, HU Q, et al. A hybrid DPSO with Levy flight for scheduling MIMO radar tasks. Applied Soft Computing, 2018, 71: 242–254.
[29] ZHANG H W, XIE J W, ZHANG Z J, et al. Variable scheduling interval task scheduling for phased array radar. Journal of Systems Engineering and Electronics, 2018, 29(5): 937–946.
[30] ZHANG H W, XIE J W, GE J A, et al. Hybrid particle swarm optimization algorithm based on entropy theory for solving DAR scheduling problem. Tsinghua Science and Technology, 2019, 24(3): 281–290.
[31] XU B, YANG C Y. Optimal region search strategy in phased array radar. Acta Electronica Sinica, 2000, 28(12): 69–73.

Biographies

TAN Qianqian was born in 1995. She received her B.S. degree from Anhui University, and M.S. degree from the University of Electronic Science and Technology of China (UESTC). She has been studying at the UESTC since 2018. Her primary research interests include statistical signal processing, detection, and dwell scheduling.
E-mail: 2210359327@qq.com

CHENG Ting was born in 1982. She received her Ph.D. degree from the University of Electronic Science and Technology of China (UESTC) and she is an associate professor and postgraduate supervisor at the UESTC, focusing on radar resource management and array signal processing methods.
E-mail: citrus@uestc.edu.cn



LI Xi was born in 1995. She is currently a master student in the University of Electronic Science and Technology of China (UESTC). Her research interests are target tracking and radar resource management.
E-mail: 779617448@qq.com