

# Variable scheduling interval task scheduling for phased array radar

ZHANG Haowei<sup>1,\*</sup>, XIE Junwei<sup>1</sup>, ZHANG Zhaojian<sup>2</sup>, SHAO Lei<sup>1</sup>, and CHEN Tangjun<sup>3</sup>

1. Air and Missile Defense College, Air Force Engineering University, Xi'an 710051, China;

2. Air Force Early Warning Academy, Wuhan 410039, China;

3. Unit 94921 of the PLA, Jinjiang 362200, China

**Abstract:** A scheduling algorithm is presented aiming at the task scheduling problem in the phased array radar. Rather than assuming the scheduling interval (SI) time, which is the update interval of the radar invoking the scheduling algorithm, to be a fixed value, it is modeled as a fuzzy set to improve the scheduling flexibility. The scheduling algorithm exploits the fuzzy set model in order to intelligently adjust the SI time. The idle time in other SIs is provided for SIs which will be overload. Thereby more request tasks can be accommodated. The simulation results show that the proposed algorithm improves the successful scheduling ratio by 16%, the threat ratio of execution by 16% and the time utilization ratio by 15% compared with the highest task mode priority first (HPF) algorithm.

**Keywords:** phased array radar, task scheduling, variable scheduling interval (SI), fuzzy set.

**DOI:** 10.21629/JSEE.2018.05.06

## 1. Introduction

The cycle over which a radar transmits (radiates) energy, waits, and receives the reflected energy, is termed a radar task. It is often the case that multiple radar tasks such as search, confirmation and tracking should be performed simultaneously in the modern battlefield. In contrast to conventional radar systems, where multiple sensors each perform a dedicate task, a phased array radar is capable of supporting the multiple and simultaneous tasks (functions). The efficient scheduling algorithm as such is the key of releasing the multifunctional potential of phased array radar.

The radar task scheduling problem has been involving in intensive research. Deb et al. [1,2] studied the scheduling problem by the task mode priority. They preset the task mode priority for each kind of task and prior schedule tasks

with the highest priority. Butler et al. [3,4] proposed the scheduling algorithm based on the time-balance, which indicated the urgency of each task. The algorithm prior scheduled the tasks with imminent deadline. However, these algorithms only prioritize tasks by a single attribute. Such a prioritization may lead to partial results. In contrast, Huizing et al. [5,6] presented the queue-based scheduling algorithm. The algorithm divided the tasks into several queues by ordinations or the value of the task parameter, and the earliest deadline first (EDF) algorithm or the first in first out (FIFO) algorithm was used in each queue. Zhang et al. [7,8] set all request tasks into dynamic priorities. The most important and urgent task can be prior scheduled. The important case occurring in [5–8] was that multiple attributes were considered. Such a prioritization was more reasonable and effective. Additionally, Taner et al. [9,10] developed heuristic scheduling algorithms for the mechanically rotating radar. Chen et al. [11,12] took the mission of imaging into account during scheduling, extending the application scope of the scheduling algorithm. To optimize the whole resource utility, Charlish et al. [13,14] proposed the continuous double auction parameter selection (CDAPS) algorithm to select the multiple task parameters. However, the radar task was regarded as a whole dwell in [1–14] above. When a complete radar task structure is modeled, the pulse interleaving technique can be utilized to further enhance the time utility. However, the pulse interleaving leads to a more complicated scheduling mode. Gopalakrishnaniet al. [15–17] tried to solve the problem by the preset templates. They utilized the offline-designed templates to schedule interleaved tasks as many as possible. The major drawback is that the templates cannot adapt to the dynamic workload of the radar system. As such, Chen et al. [18–21] offered the online pulse interleaving technology in the radar task scheduling. Different workload situations are well matched. The works [22,23]

Manuscript received April 18, 2017.

\*Corresponding author.

This work was supported by the National Youth Foundation (61503408).

put forward the notion of the variable dwell time, which made it more flexible to schedule radar tasks. Abdelaziz et al. [24–26] proposed the scheduling method based on the swarm search. Abdelaziz et al. [24] offered the tabu search scheduling algorithm, Zhang et al. provided the chaos adaptively genetic algorithm [25] and Zhang et al. proposed the hybrid genetic particle swarm algorithm [26]. They all enhanced the stability and the robustness of task scheduling for the phased array radar.

While the existing body of work approaches the scheduling problem from a variety of perspectives, they share the common assumption of modeling the scheduling interval (SI), which is the update interval of the radar invoking the scheduling algorithm, as a fixed value. However, when the variability of SI is allowable, it may be exploited to increase the number of scheduled tasks. For example, time of low time load SIs can be provided for overload SIs to accommodate more request tasks. However, the inconformity of SI time will influence the real-time property of radar system. The radar controller should decide on the range of minimum/maximum acceptable SI time to the regular (most desirable) SI time. This naturally leads to a representation of the SI time based on a fuzzy set approach. In a previous paper [8], the author introduced the threat level of targets into scheduling and proposed the dynamic priority based algorithm. The resources in the phased array radar can be utilized efficiently. However, the SI time in [8] was also regarded as fixed. On the basis of [8], the scheduling analysis is extended to several SIs, and a heuristic scheduling algorithm where SI time can be adjusted is structured. It treats the SI time as a fuzzy set, thereby allowing for greater flexibility when scheduling radar tasks. The extensive simulations verify the efficiency of the proposed algorithm.

## 2. Model structure

### 2.1 Radar task model

Typical tasks in the multifunction radar include periodical tasks and non-periodical ones. The former type includes the search task and the tracking task. The non-periodical tasks include confirmation, tracking initialization, calibration task, etc. These tasks usually consume one or several beam dwell time, which are far less than the search and tracking tasks. The request task order in the radar is usually as follows: search, confirmation, tracking initialization and tracking.

#### 2.1.1 Search task model

Based on the prior information, the radar will divide the whole surveillance region into several parts by the optimal search theory. Different parameters are adopted in

each region so that the optimal search performance can be achieved. Assume that there are  $I_s$  regions in the surveillance space, and the  $i$ th region includes  $J_s$  search beams (tasks). Then, the  $j$ th search task in the  $i$ th region can be expressed as

$$T_{si}^j = \{P_{si}, t_{asi}^j, t_{xsi}, t_{wsi}, t_{rsi}, P_{tsi}, t_{dws i}, w_{si}, t_{dsi}^j, \Delta t_{si}, I_{ni}\} \quad (1)$$

where  $P_{si}$  stands for the search task mode,  $t_{asi}^j$  is the task request time,  $t_{xsi}$  is the transmit interval,  $t_{wsi}$  is the wait interval,  $t_{rsi}$  is the receive interval,  $P_{tsi}$  is the power consumption of the search task,  $t_{dws i}$  is the task dwell time, which satisfies

$$t_{dws i} = t_{xsi} + t_{wsi} + t_{rsi}. \quad (2)$$

$w_{si}$  is the time window, and  $t_{dsi}^j$  is the task deadline. They satisfy

$$t_{dsi}^j = t_{asi}^j + w_{si}. \quad (3)$$

$\Delta t_{si}$  is the sample interval between two search tasks. The tasks will arrive sequentially after the radar beginning to search. Thus, the request time between two search tasks satisfies

$$t_{asi}^{j+1} = t_{esi}^j + \Delta t_{si} \quad (4)$$

where  $t_{esi}^j$  is the execution time of the former task.  $I_n$  can be described as

$$I_n = \{T_Y, R, v, \theta, h, p_{th}\} \quad (5)$$

where  $T_Y$  stands for the target type,  $R$  is the radial distance from the target to the radar,  $v$  is the target velocity,  $\theta$  is the heading angle of the target,  $h$  is the height of the target, and  $p_{th}$  is the threat level of the target. They all satisfy the threat model of targets in [8].

#### 2.1.2 Tracking task model

Assume that there are  $M_t$  kinds of tracking tasks in the radar, where the  $m$ th kind of tracking task contains  $N_t$  targets. The tracking task of the  $n$ th target is

$$T_{tm}^n = \{P_{tm}, t_{atm}^n, t_{xtm}^n, t_{wtm}^n, t_{rtm}^n, P_{ttm}^n, t_{dwtm}^n, w_{tm}^n, t_{dtm}^n, \Delta t_{tm}^n, I_{nm}\}. \quad (6)$$

The parameters in (6) are similar to the search task model.

#### 2.1.3 Other task model

Due to the less relativity among the non-periodical tasks, an individual model should be established for each kind of task. Assume that there are  $H$  kinds of non-periodical tasks, and the  $h$ th kind of task can be expressed as

$$T_{oh} = \{P_{oh}, t_{aoh}, t_{xoh}, t_{woh}, t_{roh}, P_{toh}, t_{dwoh}, w_{oh}, t_{doh}, \Delta t_{oh}, I_{nh}\}. \quad (7)$$

The parameters satisfy (2) to (5).

## 2.2 Resource constraint model

### 2.2.1 Time resource constraint

The scheduling procedure in the radar is based on the SI. It is the update interval of the radar invoking the scheduling algorithm. During an SI, the radar processes the returning signals in the former SI and determines the scheduling sequences in the next SI [8]. Thus, all the tasks to be executed in an SI must satisfy

$$\sum_{i=1}^{N_{\text{ex}}} t_{\text{dwi}} \leq t_{\text{SI}}. \quad (8)$$

At the same time, they should satisfy the time window constraint:

$$\begin{aligned} \max(t_{\text{start}}, t_{\text{ai}} - w_i) &\leq t_{\text{ei}} \leq \\ \min(t_{\text{di}}, t_{\text{start}} + t_{\text{SI}} - t_{\text{dwi}}) \end{aligned} \quad (9)$$

where  $N_{\text{ex}}$  is the number of tasks to be executed,  $t_{\text{SI}}$  is the scheduling interval time, and  $t_{\text{start}}$  is the start time of the SI. If the task cannot be executed in the SI, it will be delayed to the latter SIs or be deleted. In this paper, radar tasks are regarded as non-preemptive ones. It means a task cannot be interrupted once beginning to execute. Thus,  $N_{\text{ex}}$  tasks also should satisfy

$$\bigcap_{i=1}^{N_{\text{ex}}} (t_{\text{ei}}) + t_{\text{xi}} + t_{\text{ri}} = \emptyset. \quad (10)$$

### 2.2.2 Energy resource constraint

Many algorithms [1–6,9–14,22,23] neglect the energy constraint during scheduling radar tasks. Due to the restriction of its heat dissipation performance, the radar transmitter must satisfy the transient energy constraint all the time [8,21,27]:

$$P_{\tau}(t) \leq \overline{P}_{\tau \text{ max}} \quad (11)$$

where  $P_{\tau \text{ max}}$  is the power threshold and  $P_{\tau}(t)$  is the power consumed by the radar in time  $t$ .  $P_{\tau}(t)$  can be expressed as

$$P_{\tau}(t) = \frac{1}{\tau} \int_0^t p(x) e^{(x-t)/\tau} dx \quad (12)$$

where  $p(x)$  is the instantaneous power dissipation in the radar, and  $\tau$  is the look-back period.  $\tau$  characterizes the heat dissipation performance of the radar. Assume that  $P_{\tau}(t + t_c + t_x) = P_{\tau \text{ max}}$ , the cool duration  $t_c$  can be calculated as

$$t_c = -\tau \ln \frac{\overline{P}_{\tau \text{ max}} - P_t(1 - e^{-t_x/\tau})}{P_{\tau}(t) e^{-t_x/\tau}}. \quad (13)$$

Equation (13) shows that the energy constraint can be converted into the time constraint, which simplifies the calculation of the radar's time load in the following parts.

### 2.3 Time load model in the extending interval

To analyze the time load in several SIs, let a  $1 \times S_{\text{ex}}$  row vector  $\mathbf{T}_{\text{rei}}^{\text{P}}$  denote the request condition of the  $i$ th periodical tasks in each SI.  $S_{\text{ex}}$  is the number of the SIs. The  $S_{\text{ex}}$  SIs are called the extending intervals. The request condition of these tasks is not unique. For example, suppose that there are three periodical tasks  $T_1$ ,  $T_2$ , and  $T_3$  in the first SI. Their update intervals are  $\Delta t_1 = t_{\text{SI}}$ ,  $\Delta t_2 = 2t_{\text{SI}}$ , and  $\Delta t_3 = 4t_{\text{SI}}$ . If  $S_{\text{ex}} = 4$ , the request conditions of the three tasks are

$$\begin{cases} \mathbf{T}_{\text{re1}}^{\text{P}} = [t_{\text{dw1}} & t_{\text{dw1}} & t_{\text{dw1}} & t_{\text{dw1}}] \\ \mathbf{T}_{\text{re2}}^{\text{P}} = [t_{\text{dw2}} & 0 & t_{\text{dw2}} & 0] \\ \mathbf{T}_{\text{re3}}^{\text{P}} = [t_{\text{dw3}} & 0 & 0 & 0] \end{cases} \quad (14)$$

where each element is either the task dwell time if it is the task request time, or zero if it is not request time. Thus, the  $k$ th element in  $\mathbf{T}_{\text{rei}}^{\text{T}}$  can be expressed as

$$T_{\text{rei}}^{\text{P}}(k) = \begin{cases} t_{\text{dwi}}, & k = c \left\lceil \frac{\Delta t_i}{t_{\text{SI}}} \right\rceil; \quad c \in \mathbf{Z} \\ 0, & \text{otherwise} \end{cases} \quad (15)$$

On the basis of the above models, the radar time load in the extending interval is

$$\boldsymbol{\xi} = \sum_{i=1}^{N_{\text{re}}^{\text{P}}} \mathbf{T}_{\text{rei}}^{\text{T}} + \sum_{h=1}^H \mathbf{T}_{\text{reh}}^{\text{o}} + \mathbf{T}_{\text{c}} \quad (16)$$

where  $\boldsymbol{\xi}$  is a  $1 \times S_{\text{ex}}$  row vector,  $N_{\text{re}}^{\text{P}}$  is the number of periodical request tasks in the first SI,  $\mathbf{T}_{\text{reh}}^{\text{o}}$  is the time load vector of the  $h$ th kind of non-periodical task in the extending interval.  $\mathbf{T}_{\text{c}}$  is a  $1 \times S_{\text{ex}}$  vector, whose element is the cool duration needed if all request tasks are to be executed in an SI. The  $k$ th element can be expressed as

$$T_{\text{c}}(k) = \sum_{i=1}^{N_{\text{re}}(k)} t_{\text{ci}} \quad (17)$$

where  $N_{\text{re}}(k)$  is the number of request tasks in the  $k$ th SI. Equation (16) denotes that the total time load includes two parts: the time load of request tasks and the cool duration needed to execute all request tasks. It provides the basis of scheduling analysis for the extending interval.  $\boldsymbol{\xi}(k) > t_{\text{SI}}$  denotes that the radar will be overload in the  $k$ th SI, and some actions should be taken to balance the radar time load. Though the update intervals in tracking tasks or in search tasks may be adjusted due to the change of the information of targets, the scheduling analysis for the extending interval is updated by the SI (which will be shown

in the next section). Thus, the time load model can be well matched.

### 3. Heuristic scheduling algorithm

#### 3.1 Performance evaluation metrics

The following metrics are chosen to evaluate the algorithm performance.

(i) The successful scheduling ratio (SSR), which is the ratio between the number of all successfully scheduled tasks  $N_{\text{suc}}$  and that of all request tasks  $N_{\text{total}}$ . The SSR [7,11,12,18,20,21,25,26] can be expressed as

$$S_{\text{SR}} = N_{\text{suc}}/N_{\text{total}}. \quad (18)$$

Under the finite resources constraints, the more tasks can be scheduled successfully, the better the algorithm performs.

(ii) The time utilization ratio (TUR), which is defined as the ratio between the time consumption of the successfully scheduled tasks and the totally available time resource  $T_{\text{total}}$ . The TUR [1–8,11,12,15–27] can be expressed as

$$T_{\text{UR}} = \sum_{i=1}^{N_{\text{suc}}} t_{\text{dwi}}/T_{\text{total}}. \quad (19)$$

During scheduling tasks, the algorithms should use the time resource efficiently to schedule as many tasks as possible.

(iii) The threat ratio of execution (TRE) [8]. The importance of the task ranges along with the threat level of the target, and it cannot be determined by the subjectivity. Therefore, the TRE is introduced to replace the high value ratio (HVR) [18,19]. TRE is the ratio between the total threat level of targets in successfully scheduled tasks and that in all request tasks. It can be expressed as

$$T_{\text{RE}} = \sum_{i=1}^{N_{\text{suc}}} p_{\text{thi}} / \sum_{i=1}^{N_{\text{total}}} p_{\text{thi}}. \quad (20)$$

It can be seen from (20) that the higher TRE means the better performance.

#### 3.2 Scheduling algorithm for extending interval

Since the task scheduling problem in the phased array radar is NP-hard, heuristic algorithms are usually adopted to solve the problem. This section develops an efficient heuristic algorithm, which incorporates two parts: the time adjustment for SIs and the scheduling algorithm for individual SI.

On the basis of Section 1.3, the radar time load in the extending interval can be got ahead of scheduling. If there are situations where little tasks request time in some SIs

but radar is overload in other SIs, the SI time can be adjusted in advance. By shortening the former SI time and lengthening the overload SI time, the time resource can be utilized more efficiently and the radar time load in each SI can be well balanced.

Though the SIs time can be adjusted, its adjustment must be in some bound [19]. In addition, the inconformity of several SIs time will influence the real-time demand of the radar system. The desirability of the adjusted SI time ( $\Delta t_{\text{SI}}$ ) lying in a certain range leads to the treatment of  $\Delta t_{\text{SI}}$  as a fuzzy set. Each acceptable value that  $\Delta t_{\text{SI}}$  may take on is associated with a particular value of the desirability function of an SI.

$$\mu(\Delta t_{\text{SI}}) = \begin{cases} 0, & \Delta t_{\text{SI}} < t_{\text{SImin}} \\ \frac{\Delta t_{\text{SI}} - t_{\text{SImin}}}{t_{\text{SI}} - t_{\text{SImin}}}, & t_{\text{SImin}} \leq \Delta t_{\text{SI}} \leq t_{\text{SI}} \\ \frac{t_{\text{SImax}} - \Delta t_{\text{SI}}}{t_{\text{SImax}} - t_{\text{SI}}}, & t_{\text{SI}} \leq \Delta t_{\text{SI}} \leq t_{\text{SImax}} \\ 0, & \Delta t_{\text{SI}} > t_{\text{SImax}} \end{cases} \quad (21)$$

where  $t_{\text{SImin}}$  is the lower bound and  $t_{\text{SImax}}$  is the upper bound of  $\Delta t_{\text{SI}}$ . When  $\Delta t_{\text{SI}} = t_{\text{SI}}$ , the desirability function achieves the maximum value 1. Values of  $\Delta t_{\text{SI}}$  that exceed  $t_{\text{SImax}}$  or are less than  $t_{\text{SImin}}$  are not desirable. To measure the desirability of adjusted SIs in the extending intervals, the Zadeh operator is introduced:

$$\eta = \min_{k \in \{\alpha_{\text{ad}}\}} \mu(\Delta t_{\text{SI}})_k \quad (22)$$

where  $\eta$  is the desirability function of the extending interval.  $\{\alpha_{\text{ad}}\}$  is the set of SIs whose time has been adjusted. The drawback of (22) is that it provides the same result whether the time of one or of multiple SIs has been adjusted. This warrants the definition of an alternative desirability function that reflects the number of SIs which has been adjusted. In this paper, such a function is defined as the average value of the sum of the desirability function of the individual SI:

$$\eta = \left\{ \sum_{k=1}^{|\alpha_{\text{ad}}|} \mu(\Delta t_{\text{SI}})_k + (S_{\text{ex}} - |\alpha_{\text{ad}}|) \right\} / S_{\text{ex}}. \quad (23)$$

Equation (23) reflects that a higher value is more desirable. Therefore, when adjusting SI time to meet the requirement of radar task scheduling, a higher desirability value should be taken into consideration.

Above all, the scheduling algorithm is structured aiming at the following two points: adjusting SI time to accommodate more tasks and well balancing the radar time load. At the same time, achieve the maximum of  $\eta$ . The scheduling algorithm is described as follows.

### 3.2.1 Time adjustment for SIs

**Step 1** After analyzing the request tasks in the first SI, initialize the next  $S_{\text{ex}} - 1$  SIs as the extending interval (including the first SI). Set the extending interval pointer  $j_{\text{ex}} = 1$ .

**Step 2** Calculate the time load of each SI in the extending interval according to Section 1.3. Test whether there are overload SIs.

$$\xi(k) > t_{\text{SI}} \quad (24)$$

If there is no overload SI, turn to Step 6. If there are some overload SIs, test whether the overload SIs can be balanced by adjusting SIs time.

$$S_{\text{ex}} t_{\text{SI}} - \sum_{k=1}^{S_{\text{ex}}} \xi(k) \geq 0 \quad (25)$$

If the overload SIs cannot be balanced, turn to Step 6. Otherwise, turn to Step 3.

**Step 3** Assume that there are  $S_{\text{ov}}$  overload SIs. Calculate the sum of synthetic priorities of request tasks in each SI according to [8]. Assume that  $P_{\text{SI}}(k)$  is the sum of priorities of request tasks in the  $k$ th SI. Then, sort the  $S_{\text{ov}}$  SIs diminishingly by their  $P_{\text{SI}}(k)$ . Sign the SI which has the highest  $P_{\text{SI}}(k)$  as  $S(k_{\text{ov}})$ , and set  $k_{\text{ov}} = 1$ .

**Step 4** Seek the SIs which are next to  $S(k_{\text{ov}})$  and not overload sequentially in the extending interval. Sign them as SI1 and SI2. If SI1 =  $\emptyset$  and SI2 =  $\emptyset$ , set  $k_{\text{ov}} = k_{\text{ov}} + 1$ , and turn to Step 5. Otherwise, calculate if

$$(N+1)t_{\text{SI}} - \sum_{k=1}^N \min \Delta t_{\text{SI}}(k) \geq \xi(k_{\text{ov}}), \quad N = 1, 2 \quad (26)$$

where  $\min \Delta t_{\text{SI}}(k)$  is the minimum time that the SI can be shortened when its time can accommodate all request tasks. It satisfies

$$\min \Delta t_{\text{SI}}(k) = \max\{t_{\text{SImin}}, \xi(k)\} \quad (27)$$

when (26) gets the equal sign, adjust time of  $S(k_{\text{ov}})$  and SI1 (and SI2) to their fit values. It is noticeable that when  $N = 2$ , time of SI1 should be prior shortened to  $\min \Delta t_{\text{SI}}(1)$ . Then, set  $k_{\text{ov}} = k_{\text{ov}} + 1$ , and turn to Step 5.

**Step 5** When  $k_{\text{ov}} \geq S_{\text{ov}}$ , the time adjustment for SIs ends. Otherwise, return to Step 4.

**Step 6** Schedule tasks in each SI using the scheduling algorithm for individual SI. Then, set  $j_{\text{ex}} = j_{\text{ex}} + 1$  to get a new extending interval, and return to Step 2.

The flow chart of the algorithm is shown in Fig. 1.

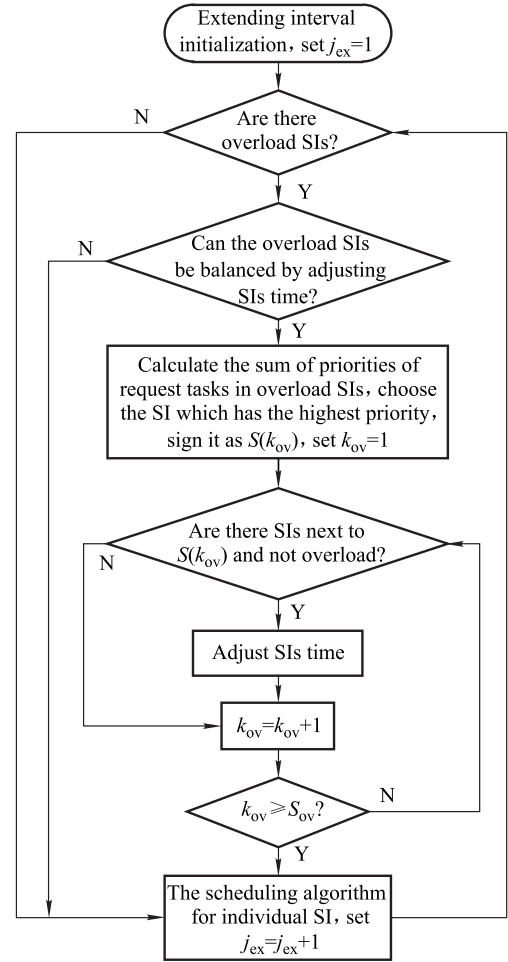


Fig. 1 Flow chart of time adjustment for SIs

### 3.2.2 Scheduling algorithm for individual SI

**Step 1** Initialize the SI pointer  $k_{\text{SI}} = 0$ .

**Step 2** Set  $k_{\text{SI}} = k_{\text{SI}} + 1$ , and the task pointer  $i_{\text{ta}} = 0$ . Get the number of the request tasks  $N_{\text{re}}$ , the time pointer  $t_p$ , the initial power consumption  $P_{t0}$  and the end time of the SI  $t_{\text{end}}$ .

**Step 3** Delete the  $M_{\text{ta}}$  tasks whose deadline are over  $t_p$ , and set  $i_{\text{ta}} = i_{\text{ta}} + M_{\text{ta}}$ .

**Step 4** Select all the tasks that can be executed earlier than  $t_p$ , sort the tasks by their synthetic priorities diminishingly and choose task  $T_l$  which has the highest priority. Set  $l = 1$ .

**Step 5** Test whether task  $T_l$  satisfies the time constraint. If it does not satisfy the time constraint, set  $l = l + 1$ ,  $i_{\text{ta}} = i_{\text{ta}} + 1$ , and turn to Step 6. If it satisfies the time constraint, test whether  $T_l$  satisfies the energy constraint. If it satisfies the constraint, update the parameters as

$$\begin{cases} t_p = t_p + t_{\text{dwl}} \\ P_{t0} = [P_{t0} e^{-t_{\text{xl}}/\tau} + P_t (1 - e^{-t_{\text{xl}}/\tau})] e^{-(t_{\text{wl}} + t_{\text{rl}})/\tau} \end{cases} \quad (28)$$

Then, send it to the execution queue and set  $i_{ta} = i_{ta} + 1$ . If it does not, calculate  $t_c$  and update the parameters as

$$\begin{cases} t_p = t_p + t_c + t_{dwl} \\ P_{t0} = \bar{P}_{\tau \max} e^{-(t_{wl} + t_{rl})/\tau} \end{cases} \quad (29)$$

Then, send it to the execution queue and set  $l = l + 1$ ,  $i_{ta} = i_{ta} + 1$ .

**Step 6** When  $t_p \geq t_{\text{end}}$  or  $i_{ta} \geq N_{\text{re}}$ , turn to Step 7; otherwise, turn to Step 3.

**Step 7** Test whether the remaining tasks can be delayed. If they do, update their parameters as  $t_{ai} = t_{\text{end}}$ ,  $t_{wi} = t_{wi} - (t_{\text{end}} - t_{ai})$ , and send them to the delayed queue. Otherwise, send them to the deleted queue.

**Step 8** The analysis of the SI ends, get the execution tasks, delayed tasks, deleted tasks,  $t_p$  and  $P_{t0}$ . Then, turn to Step 2.

The flow chart of the algorithm is shown in Fig. 2.

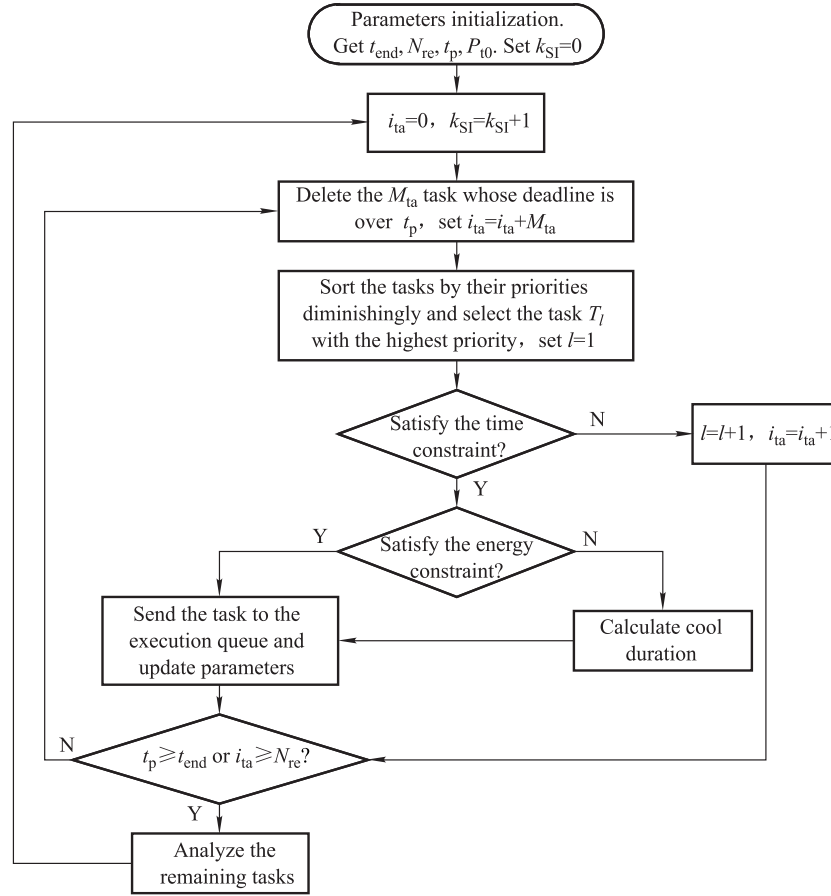


Fig. 2 Flow chart of scheduling algorithm for individual SI

#### 4. Computational results and analysis

The whole simulation frame is established based on references [8,20,21]. In simulations, set  $t_{SI} = 50$  ms,  $P_{\tau \max} = 1.25$  kW,  $\tau = 200$  ms,  $t_{SI \min} = 25$  ms,  $t_{SI \max} = 100$  ms,  $S_{\text{ex}} = 4$ . The task parameters are shown in Table 1. The dynamic algorithm [8], the HPF algorithm [2] and the SI adjustment based algorithm (SI adjustment algorithm) [28] are used as the baseline to compare against the performance of the proposed algorithm. Note that the SI time also can be adjusted in the last algorithm. However, the algorithm does not treat the SI time as a fuzzy set, just tweaks it according to the sample intervals and dwell time of tasks.

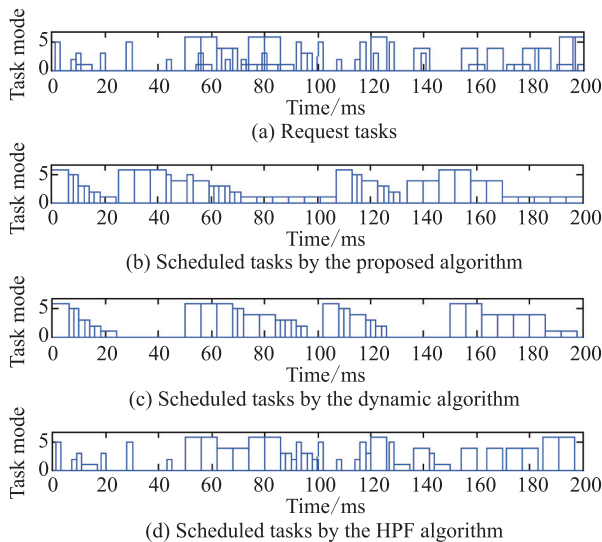
Table 1 Parameters of tasks

Task	$P$	$t_x, t_w, t_r/\text{ms}$	$P_t/\text{kW}$	$w/\text{ms}$	$\Delta t/\text{ms}$
Confirmation	6	1, -, 1	5	50	150
High precision tracking	5	0.5, -, 0.5	4	50	100–200
Tracking loss	4	1, -, 1	5	100	–
Precision tracking	3	0.5, -, 0.5	3	200	250–500
Normal tracking	2	0.5, -, 0.5	3	500	1 000
Search	1	1, -, 1	5	–	10

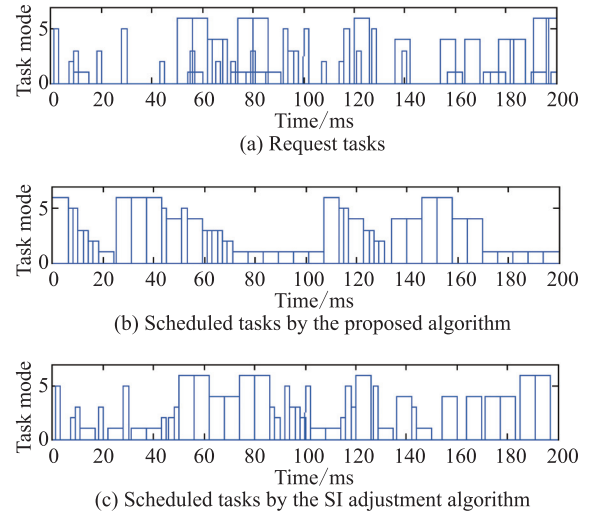
##### 4.1 Illustrative example

At first, an illustrative example is presented to show the procedure and the performance of the proposed scheduling method. The request tasks and the scheduling results

are shown in Fig. 3 and Fig. 4, where the vertical axis is the task mode, the horizontal axis is the timeline, and each rectangle represents a radar task. It is a scene that many tasks are generated randomly in an extending interval (4 SIs). The radar time load  $\xi = [24, 82, 24, 66]$ . It is obvious that the second SI (SI 2) and the fourth SI (SI 4) are overload. The sum of priorities of request tasks in the two SIs are calculated as  $P_{SI}(2) > P_{SI}(4)$ . Thus, time of SI 1 and SI 3 is firstly shortened for SI 2. SI 1 is shortened to 25 ms ( $\min \Delta t_{SI}(1)$ ), and SI 3 is shortened by 7 ms. Then, SI 3 is shortened by another 16 ms to accommodate request tasks in SI 4. After the adjustment, time of SI 1 to SI 4 is 25 ms, 82 ms, 27 ms and 66 ms respectively, and  $\eta = 0.28$ . The scheduled tasks by the proposed method are shown in Fig. 3(b) and Fig. 4(b). As a comparison, the scheduling results by the other three algorithms are also presented. It can be seen that though the dynamic algorithm uses time efficiently in each SI, the idle time in SI 1 and SI 3 is wasted. Due to the restriction of the fixed SI time, many tasks in SI 2 and SI 4 are dropped. Similar to the dynamic algorithm, the HPF algorithm wastefully ignores the utilization of the idle time in SIs which are not overload. In addition, the HPF algorithm leaves some unnecessary time between tasks. It will cause more tasks being dropped especially in overload situations. Additionally, the SI adjustment algorithm does adjust SI time to accommodate more tasks. However, the following two points should be highlighted. Firstly, much idle time is left between radar tasks, and time is not fully utilized in each SI. Secondly, the SI adjustment is arbitrary. Though more request tasks are successfully scheduled (note that not all tasks are successfully scheduled by the SI adjustment algorithm), the lopsidedness of radar time load in each SI still exists, or even worse.



**Fig. 3** Scheduling results by the proposed method vs non-SI adjustment algorithm

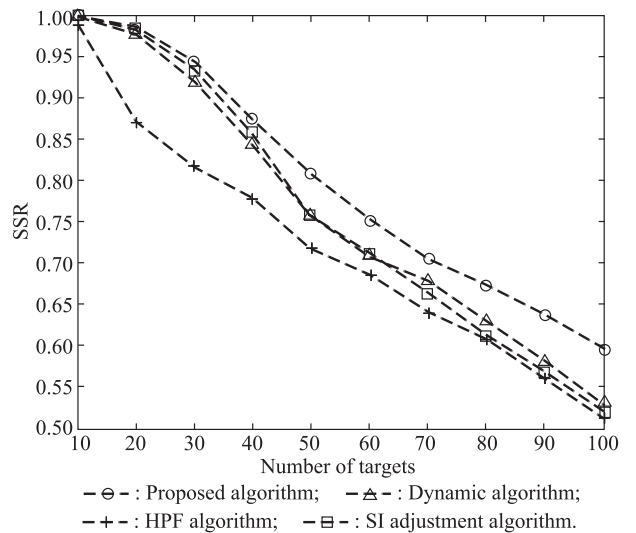


**Fig. 4** Scheduling results by the proposed method vs SI adjustment algorithm

Meanwhile, such an adjustment naturally leads to more additionally computational burden (It can be seen in the next section). However, the proposed scheduling algorithm elegantly adjusts SI time and utilizes idle time in the extending interval. Thus, all request tasks are scheduled successfully and the radar time load is well balanced.

## 4.2 Performance assessment

In assessing the performance of the proposed scheduling method, the total simulation time is 50 s (1 000 SIs). The results are shown from Fig. 5 to Fig. 7.



**Fig. 5** Comparison of SSR

Fig. 5 shows the SSR comparison as a function of the number of targets. It can be seen that when the number of targets is 10, all request tasks can be scheduled successfully by all the algorithms except the HPF algorithm.



When the number of targets is over 10, the dynamic algorithm, the SI adjustment algorithm and the proposed algorithm start to drop tasks successively. However, the decreasing speed of the proposed algorithm is much smaller than the other three algorithms. This is because the modeling of the SI time as a fuzzy set allows the proposed scheduling method to adjust SIs time, rather than simply dropping tasks as in the dynamic algorithm and the HPF algorithm. The idle time in other SIs is provided for SIs which are overload. Thus, more request tasks can be accommodated. Though the SI adjustment algorithm also adjusts SIs for this purpose, it provides inferior performance compared with the proposed algorithm. Additionally, the SI adjustment algorithm provides lower SSR than the dynamic algorithm when the number of targets is more than 60. That is because when the number of request tasks is in lower level, the SI adjustment dominates the whole performance. However, when the number of targets is sizable, the scheduling algorithm in each SI leads the main performance. However, time in each SI cannot be fully utilized by the SI adjustment algorithm. As such, its SSR decreases more quickly than the dynamic algorithm.

Fig. 6 compares the TRE of the four algorithms. Similar to the SSR, the proposed algorithm gets the highest TRE. In the HPF algorithm and the SI adjustment algorithm, the task synthetic priority is preset by the subjectivity. However, the proposed algorithm and the dynamic algorithm set all request tasks into dynamic priorities, which are calculated by the threat level of targets and deadline of tasks. The most important and urgent task thus can be prior scheduled. With the SI adjustment, the proposed algorithm further utilizes the idle time resource on the basis of the dynamic algorithm. The proposed algorithm as such gets the highest TRE.

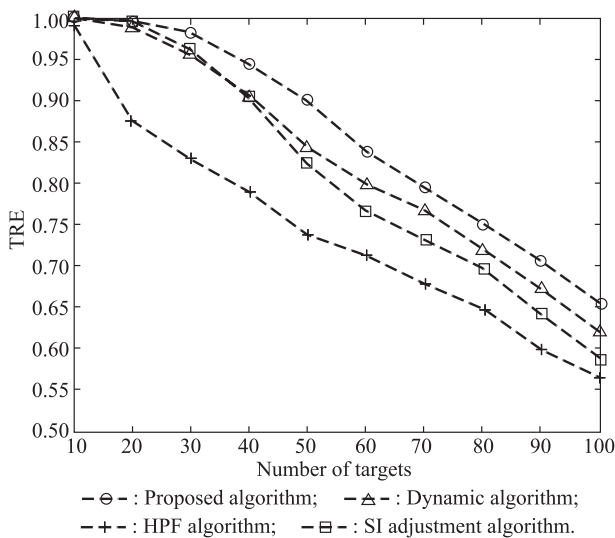


Fig. 6 Comparison of TRE

Fig. 7 is the comparison of TUR of the four algorithms. It is clear that the HPF algorithm has the worst TUR. As is shown in Fig. 3, the HPF algorithm leaves some unnecessary idle time between tasks. The time resource is not fully utilized. As the number of targets increases, the radar timeline will be overload gradually. The SI time in the proposed algorithm is elegantly adjusted to accommodate the increasing tasks. Thus, the TUR of the proposed algorithm approaches to 1. Though the SI adjustment algorithm also exploits the SI adjustment technique, it provides inferior performance than the proposed algorithm. Additionally, its TUR is even lower than the dynamic algorithm when the number of targets is over 70. That is because the scheduling algorithm in each SI affects most when faced with large-scale of request tasks, and the SI adjustment provides less contribution to the performance. However, as shown in Fig. 4, time in each SI cannot be utilized efficiently by the SI adjustment algorithm. Due to the restriction of the fixed SI time, the idle time in SIs cannot be utilized in the dynamic algorithm and its TUR is less than the proposed algorithm.

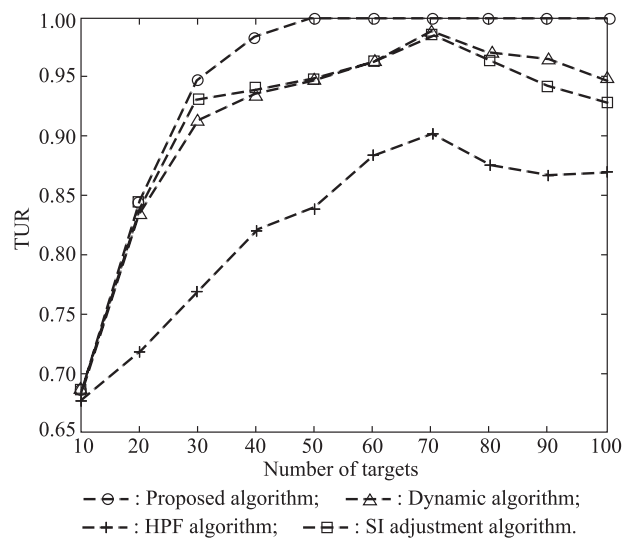


Fig. 7 Comparison of TUR

Fig. 8 shows the desirability function value of the extending interval when using the four scheduling methods. Due to the fixed SI time in the dynamic algorithm and the HPF algorithm, the function values of the two algorithms are both 1. However, the function value of the proposed algorithm and the SI adjustment algorithm both vary along with the number of targets. When the number of targets is less than 10, the scheduling algorithm for individual SI is able to handle all request tasks and the desirability function value is 1. When the number of targets is over 10, the time adjustment for SIs becomes increasingly active. SI time is adjusted to accommodate the increasing request tasks, leading to the decrease of the desirability function



value. When the target number is over 30, the radar timeline is approximately saturated. The SI time is gradually unable to be adjusted to accommodate more request tasks. The time adjustment for SIs becomes less active and the desirability function value thus returns to a higher level. However, the desirability function value of the proposed method is much higher than the SI adjustment method, indicating a more regular SI adjustment result and a more balanced time load.

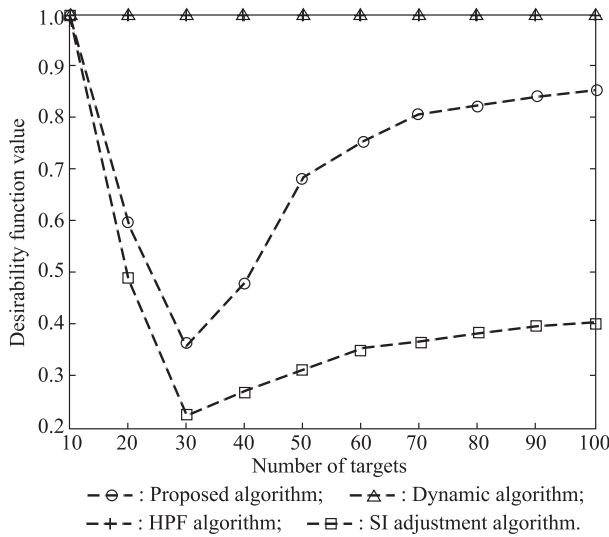


Fig. 8 Desirability function value

Table 2 Average runtime in an SI

ms

Algorithm	Number of targets									
	10	20	30	40	50	60	70	80	90	100
HPF	1.3	1.9	2.3	2.6	3.0	3.2	3.7	4.1	4.4	4.5
Dynamic priority	2.1	2.7	3.2	3.6	4.1	4.5	4.9	5.4	5.9	6.3
Proposed	4.6	6.1	8.2	9.5	10.7	12.1	13.4	14.0	14.5	14.8
SI adjustment	8.2	12.9	17.2	20.8	22.5	24.6	27.1	28.9	30.2	31.6

## 5. Conclusions

The efficient tasks scheduling algorithm is the key of releasing the multifunctional potential of the phased array radar. This paper develops a scheduling algorithm based on treating the scheduling interval time as a fuzzy set. By adjusting SIs time, the proposed algorithm intelligently provides idle time for overload SIs to accommodate more request tasks. The simulation results show that the proposed algorithm improves the SSR by 16%, the TRE by 16% and the TUR by 15% compared with the HPF algorithm. The fuzzy set notion presented in this paper can be also applied to areas such as job shop scheduling and manufacturing.

## References

- [1] DEB D, BHATTACHARJEE R, VENGADARAJAN A. Resource manager for MIMO radar. Proc. of the IEEE Radar Conference, 2015: 71–75.

That is because the optimization model is established in the proposed method, and the aim is to achieve the smallest change and the balance of time load in each SI. By contrast, the SI adjustment method arbitrarily adjusts SIs and lacks an overall consideration to the extending interval.

Table 2 compares the average runtime of the four algorithms. It can be seen that all the algorithms can respond to radar system in the real-time manner. Though the HPF algorithm provides the fastest runtime, it also provides the worst performance. Due to the SI adjustment, the proposed algorithm and the SI adjustment algorithm need more time to schedule radar tasks. However, it can be seen that the proposed algorithm is more efficient than the SI adjustment algorithm. As mentioned before, the inconformity of SI time will bring additional computational burden. Since the proposed method elegantly achieves the minimum change of SI time and the time load balance, less computational burden is added to the radar system. By contrast, the SI adjustment algorithm lacks an overall consideration when adjusting SI time. That leads to much inconformity of SI time. Meanwhile, the inconformity encumbers the radar response and retroacts to the algorithm, further delaying its runtime. As such, the SI adjustment algorithm provides the longest runtime.

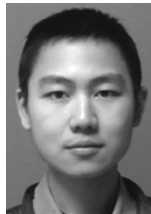
- [2] SGAMBATO P, CELENTANO S, DI DIO C. A flexible on-line scheduling algorithm for multifunctional radar. Proc. of the IEEE Radar Conference, 2016: 1–5.
- [3] BUTLER J M. Multi-function radar tracking and control. London, UK: University College London, 1998.
- [4] REINOSO-RONDINEL R, YU T Y, TORRES S. Multifunction phased-array radar: time balance scheduler for adaptive weather sensing. Journal of Atmospheric and Oceanic Technology, 2010, 27(11): 1854–1867.
- [5] HUIZING A G, BLOEMENA A F. An efficient scheduling algorithm for a multifunction radar. Proc. of the IEEE International Symposium on Phased Array Systems and Technology, 1996: 359–364.
- [6] JIMENEZ M I, DEL VAL L, VILLACORTA J J, et al. Design of task scheduling process for a multifunction radar. IET Radar, Sonar & Navigation, 2012, 6(5): 341–347.
- [7] ZHANG H W, XIE J W, SHENG C. Adaptive scheduling algorithm over comprehensive priority for phased array radar. Acta Armamentarii, 2016, 37(11): 2163–2169. (in Chinese)
- [8] ZHANG H W, XIE J W, ZONG B F, et al. Dynamic priority scheduling method for air defense phased array radar. IET

- Radar, Sonar & Navigation, 2017, 11(7): 1140–1146.
- [9] TANER M R, KARASAN O E, YAVUZTURK E. Scheduling beams with different priorities on a military surveillance radar. *IEEE Trans. on Aerospace and Electronic Systems*, 2012, 48(2): 1725–1739.
  - [10] BHAT S S, NARAYANAN R M, RANGASWAMY M. Bandwidth sharing and scan scheduling in multimodal radar with communications and tracking. *IETE Journal of Research*, 2013, 59(5): 551–562.
  - [11] CHEN Y J, LUO Y, ZHANG Q, et al. Adaptive scheduling algorithm for phased array radar based on cognitive ISAR imaging. *Journal of Electronics & Information Technology*, 2014, 36(7): 1566–1572. (in Chinese)
  - [12] CHEN Y J, ZHANG Q, YUAN N, et al. An adaptive ISAR-imaging-considered task scheduling algorithm for multi-function phased array radars. *IEEE Trans. on Signal Processing*, 2015, 63(19): 5096–5110.
  - [13] CHARLISH A, WOODBRIDGE K, GRIFFITHS H. Multi-target tracking control using continuous double auction parameter selection. *Proc. of the 15th International Conference on Information Fusion*, 2012: 1269–1276.
  - [14] CHARLISH A, WOODBRIDGE K, GRIFFITHS H. Phased array radar resource management using continuous double auction. *IEEE Trans. on Aerospace and Electronic Systems*, 2015, 51(3): 2212–2224.
  - [15] GOPALAKRISHNAN S, CACCAMO M, SHIH C S, et al. Finite-horizon scheduling of radar dwells with online template construction. *Real-Time Systems*, 2004, 33(1–3): 47–75.
  - [16] LEE C G, KANG P S, SHIH C S, et al. Schedulability envelope for real-time radar dwell scheduling. *IEEE Trans. on Computers*, 2006, 55(12): 1599–1613.
  - [17] GOPALAKRISHNAN S, CACCAMO M, SHA L. Sharp thresholds for scheduling recurring tasks with distance constraints. *IEEE Trans. on Computers*, 2008, 57(3): 344–358.
  - [18] CHEN J, WANG L, ZHANG W, et al. Multifunction phased radar resource management via maximal pulse interleaving technique. *Arabian Journal for Science & Engineering*, 2013, 38(1): 3081–3091.
  - [19] CHENG T, LIAO W W, HE Z S. MIMO radar dwell scheduling based on novel pulse interleaving technique. *Journal of Systems Engineering and Electronics*, 2013, 24(2): 234–241.
  - [20] ZHANG H W, XIE J W, ZHANG Z J, et al. Online task interleaving scheduling for the digital array radar. *AEU-International Journal of Electronics and Communications*, 2017, 79: 250–256.
  - [21] 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): 67–78.
  - [22] MIR H S, ABDELAZIZ F B. Cyclic task scheduling for multifunction radar. *IEEE Trans. on Automation Science and Engineering*, 2012, 9(3): 529–537.
  - [23] MIR H, GUITOUNI A. Variable dwell time task scheduling for multifunction radar. *IEEE Trans. on Automation Science and Engineering*, 2014, 11(2): 463–472.
  - [24] ABDELAZIZ F B, MIR H. An optimization model and tabu search heuristic for scheduling of tasks on a radar sensor. *IEEE Sensors Journal*, 2016, 16(17): 6694–6702.
  - [25] ZHANG H W, XIE J W, SHENG C. Scheduling method for phased array radar over chaos adaptively genetic algorithm. *Proc. of the 6th International Conference on Information Science and Technology*, 2016: 111–116.
  - [26] ZHANG H W, XIE J W, LU W L, et al. A scheduling method

based on the hybrid genetic particle swarm algorithm for the multifunction phased array radar. *Frontiers of Information Technology & Electronic Engineering*, 2017, 18(11): 1806–1816.

- [27] GHOSH S, RAJKUMAR R R, HANSEN J, et al. Integrated QoS-aware resource management and scheduling with multi-resource constraints. *Real-Time System*, 2006, 33(1): 7–46.
- [28] LIU J, CHEN Z, MA L, et al. Radar resource scheduling algorithm based on variable length scheduling interval. *Journal of Terahertz Science and Electronic Information Technology*, 2016, 14(1): 58–63. (in Chinese)

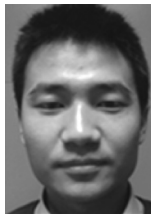
## Biographies



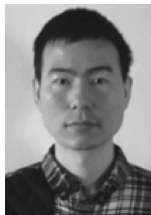
**ZHANG Haowei** was born in 1992. He received his bachelor and master degrees from the Air and Missile Defense College, Air Force Engineering University in 2014 and 2016, respectively. He is currently a doctoral student in the Air and Missile Defense College. His research interests include multi-function radar resource management and intelligent scheduling.  
E-mail: zhw\_xhzhf@163.com



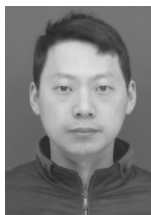
**XIE Junwei** was born in 1970. He received his bachelor, master, and doctor degrees from the Air and Missile Defense College, Air Force Engineering University in 1993, 1996 and 2009, respectively. He is currently a professor in the Air and Missile Defense College. His research interests include novel radar systems as well as jamming and anti-jamming.  
E-mail: xjw\_xjw\_123@163.com



**ZHANG Zhaojian** was born in 1989. He received his bachelor, master and doctor degrees from Air and Missile Defense College, Air Force Engineering University in 2011, 2013 and 2017, respectively. He is currently a lecturer in the Air Force Early Warning Academy. His research interest is radar signal processing.  
E-mail: zzj554038@163.com



**SHAO Lei** was born in 1982. He received his bachelor, master, and doctor degrees from Air and Missile Defense College, Air Force Engineering University in 2001, 2003, and 2007, respectively. He is an associate professor in Air and Missile Defense College. His research interest is model generation of reentry aircraft.  
E-mail: shaoleijj@126.com



**CHEN Tangjun** was born in 1983. He received his bachelor and master degrees from Air and Missile Defense College, Air Force Engineering University in 2004 and 2016, respectively. He is currently an engineer in Unit 94921 of the PLA. His interest is radar engineering.  
E-mail: ctj3959@126.com