

基于价值优化的相控阵雷达任务调度算法

杨善超^{*①} 田康生^① 刘仁争^① 郑玉军^②

^①(空军预警学院预警情报系 武汉 430019)

^②(中国人民解放军94710部队 无锡 214000)

摘要: 针对相控阵雷达时间资源分配问题, 该文提出一种基于价值优化的任务调度算法。首先建立任务调度属性参数, 对跟踪任务队列进行可行性分析和筛选操作, 确定跟踪任务调度属性。其次, 根据任务最大价值及其变化斜率, 建立关于实际执行时刻的动态任务价值函数, 并基于此构建任务调度的价值优化模型, 对跟踪任务执行时刻进行分配, 以更好满足及时性原则。最后, 利用执行跟踪任务间的空闲时间片对搜索任务进行调度。仿真结果表明, 该文算法有效减小了时间偏移量, 提升了实现价值率。

关键词: 相控阵雷达; 资源管理; 任务调度; 及时性

中图分类号: TN958.92

文献标识码: A

文章编号: 1009-5896(2020)02-0465-07

DOI: 10.11999/JEIT190147

Scheduling Algorithm Based on Value Optimization for Phased Array Radar

YANG Shanchao^① TIAN Kangsheng^① LIU Renzheng^① ZHENG Yujun^②

^①(The Early Warning Intelligence Department, Air Force Early Warning Academy, Wuhan 430019, China)

^②(The Unit 94710 of The Chinese People's Liberation Army, Wuxi 214000, China)

Abstract: A task scheduling algorithm based on value optimization is proposed for phased array radar. Firstly, the schedulability of tracking tasks is obtained through feasibility analysis and selecting operation on the task queue, using the proposed schedulability parameters. Then, a dynamic task value function about the actual execution time is established according to the peak value and value changing slope of tasks. A value optimization model for tracking task scheduling is constructed based on the task value function. Timeliness can be better achieved while adopting this model to assign execution time for tasks. Finally, searching tasks are scheduled using the idle time intervals between tracking tasks which are going to be executed. Simulation results show that proposed algorithm reduces the average time shift ratio, and improves the value achieving ratio compared with the traditional scheduling algorithms.

Key words: Phased array radar; Resource management; Task scheduling; Timeliness

1 引言

相控阵雷达具有波束无惯性捷变和波形自适应能力, 可以同时执行搜索、跟踪等多种任务^[1], 相比传统机械扫描雷达有着巨大优势。为充分发挥这种性能优势, 需要对雷达任务进行自适应调度, 以实现相控阵雷达资源的高效分配^[2]。

相控阵雷达任务调度模块主要包含两部分: 任务优先级规划和调度策略选取。典型的动态优先级算法是截止期最早最优先(Earliest Deadline First, EDF)算法^[2], 但是只利用一种参数动态确定任务优先级, 仍然不够全面。文献[3-5]将工作方式优先级和任务截止期置于同一层面, 并引入优先级表的思想来计算任务综合优先级, 提出工作方式优先级加截止期算法(High Priority And EDF, HPEDF)及其两种变形算法: 修正EDF算法(Modified EDF, MEDF)和修正工作方式优先级算法(Modified High Priority First, MHPF)。然而上述优先级规划算法没有充分利用目标先验信息, 且存在人为划定工作方式优先级而导致主观性过强的问题。针对此, 文献[6]利用目标信息构建了目标威胁度模型, 通过目

收稿日期: 2019-03-13; 改回日期: 2019-05-24; 网络出版: 2019-08-15

*通信作者: 杨善超 y_qinglinshuo@163.com

基金项目: 国家863计划(2015AA7056045), 国家自然科学基金(61601510)

Foundation Items: The National 863 Program of China (2015AA7056045), The National Natural Science Foundation of China (61601510)

标威胁度和任务截止期共同确定任务优先等级。在调度策略方面,主要有模板类策略和自适应调度策略,后者更能发挥相控阵雷达的优势。文献[7-9]将时间窗概念引入动态任务调度中,使雷达任务请求的实际执行时刻可以在其期望时间前后移动,有利于提高调度成功率和时间利用率。文献[10,11]采用了任务交错算法,使任务的等待期可以执行其他任务的发射或接收,进一步提高时间利用率。文献[12-15]提出了基于代价或收益的调度算法,构建任务调度的优化模型,并利用二次规划、启发式算法等进行求解。上述研究存在以下问题:一是将任务价值作为一个与任务优先级等价的固定参数,没有考虑任务价值随执行时刻的动态变化;二是没有充分考虑任务调度及时性及其对实现价值率的影响,尤其是对于高价值任务的及时性。

因此,本文提出一种基于价值优化的相控阵雷达任务调度算法。采用跟踪任务竞争时间资源,而搜索任务等待空闲时间片的策略^[16]。首先对跟踪任务进行调度,提出任务调度属性参数,以此判断跟踪任务请求队列可行性;对于不可行队列进行筛选操作,生成新的可行队列。随后建立关于实际执行时刻的动态任务价值函数,并构建基于价值优化的任务调度模型,使跟踪任务尽可能贴近期望时刻被执行,以满足调度及时性原则。其次,利用空闲时间片进行搜索任务调度。最后,通过仿真对本文方法进行验证。

2 调度算法总体结构

任务调度过程一般以调度间隔(Scheduling Interval, SI)为周期,在一个SI中,调度器接收输入端的任务请求队列,进行处理后输出结果是执行队列、延迟队列、删除队列。

相控阵雷达工作在跟踪加搜索(Track And Search, TAS)模式下,所执行的任务种类主要包括两种:跟踪和搜索,且跟踪任务一般比搜索任务优先级更高。跟踪任务对执行时刻要求严格,这是由于期望调度时刻由跟踪采样间隔决定,提前执行虽然能提升相应目标跟踪精度,但是会占用更多时间资源,影响其它任务执行;若延迟执行,则会降低目标跟踪精度,甚至检测不到目标,这意味着当一个跟踪驻留请求的调度被过度延迟时,会导致任务被删除。因此,任务调度过程需要尽可能满足跟踪任务的时间资源要求,并提高其调度及时性。搜索任务属于常驻任务,其驻留请求一般不受时间窗限制,可以持续发送给调度器,未被执行的任务都能够在下一调度间隔继续进入请求队列,被成功调度后依然具备有效性。

根据上述跟踪、搜索任务调度的相关特点,本文采用跟踪任务竞争时间资源,而搜索任务等待空闲时间片的思想,将调度器分成跟踪任务调度器、搜索任务调度器两个子调度器,其总体结构框图如图1所示。

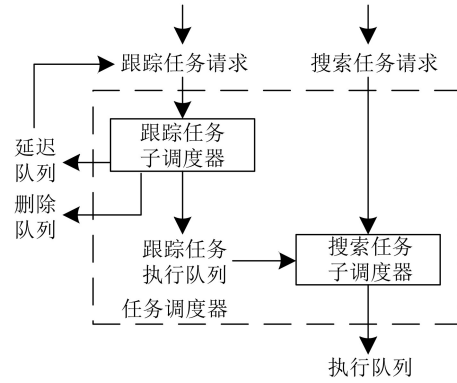


图1 任务调度模型总体结构

根据该模型,任务调度的总体过程描述如下:首先,跟踪任务请求队列输入到跟踪任务子调度器,输出跟踪任务执行队列、延迟队列和删除队列;随后搜索任务子调度器接收跟踪任务执行队列和搜索任务请求队列,利用执行跟踪任务间的空闲时间片对搜索任务进行调度,生成最终执行任务队列。

3 跟踪任务调度

3.1 任务模型

在调度间隔 $[t_0, t_0 + SI]$ 内,假设跟踪任务调度模块收到 M 个任务驻留请求,第 i 个任务请求为

$$T_i = \{P_i, t_{ei}, t_{ai}, t_{di}, ti_i, \Delta t_i, V_i^*, \delta_i, \theta_i\} \quad (1)$$

其中 P_i 为任务优先级; t_{ei} 为期望执行时刻; t_{ai} 为请求到达时刻, t_{di} 为截止期,因此任务时间窗为 $[t_{ai}, t_{di}]$; ti_i 为采样间隔; Δt_i 为任务驻留时间; V_i^* 为任务最大价值,当任务在期望时刻被调度时可达该最大价值;假设任务实际执行时刻为 t_{si} ,当 t_{si} 相对于 t_{ei} 提前或者延迟时,分别用 δ_i, θ_i 表示任务价值变化斜率大小,此外 t_{ei}, t_{si} 应满足

$$t_{ai} \leq t_{ei} \leq t_{di} \quad (2)$$

$$t_{ai} \leq t_{si} \leq t_{di} \quad (3)$$

如果 T_i 在 t_{si} 被执行,那么执行结束时刻为 $t_{si} + \Delta t_i$ 。由于相控阵雷达在同一时刻只能实施一次驻留,下一任务的执行时刻不能早于 $t_{si} + \Delta t_i$ 。由此给出以下定义:

定义 对于一个按照执行时刻 t_{si} 由小到大排列的任务队列,当其满足

$$t_{si} + \Delta t_i \leq t_{si+1}, \forall i = 1, 2, \dots, M-1 \quad (4)$$

时,该任务队列是可行的。

3.2 跟踪任务调度流程

本文假设输入的 M 个任务按照期望执行时刻由小到大排列成 $\{T_1, T_2, \dots, T_M\}$ ，即

$$t_{e1} \leq t_{e2} \leq \dots \leq t_{eM} \quad (5)$$

跟踪任务调度主要分两步，一是判断任务调度属性，即能否被调度，原则是尽可能调度多的任务；二是为能够被调度的任务分配执行时刻，原则是尽可能使任务执行时刻贴近期望时刻。跟踪任务调度模块结构框图如图2所示。

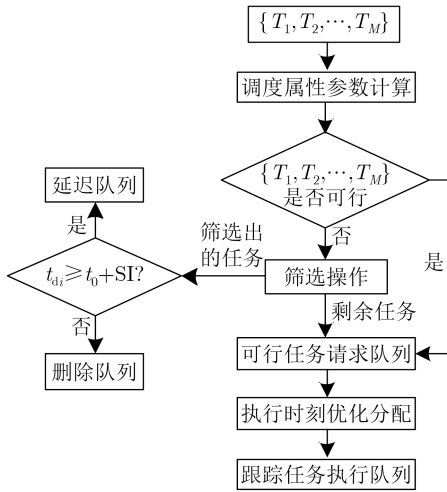


图2 跟踪任务调度流程

调度流程简要描述如下：

步骤1 对 $\{T_1, T_2, \dots, T_M\}$ 进行任务调度属性参数计算，并判断其是否可行。若可行，转步骤3；若不可行，转步骤2。

步骤2 对 $\{T_1, T_2, \dots, T_M\}$ 进行筛选操作，对筛选出的任务请求，判断其是否满足延迟条件： $t_{di} \geq t_0 + SI$ 。若满足，将其加入延迟队列；若不满足，则加入删除队列。剩余任务请求形成一个可行的可行队列，转步骤3。

步骤3 对可行任务请求队列进行执行时刻优化分配，得到跟踪任务执行队列。

3.3 任务调度属性参数计算

任务调度属性参数主要包括两方面：最早可能执行时刻 $\{t_{fi}'\}_{i=1}^M$ 和最大可能延迟时间 $\{D_i'\}_{i=1}^M$ 。参数的求解过程如下：

步骤1 令 $t_{f1}' = t_{a1}$ ， $D_1' = t_{d1} - t_{f1}'$ ，设置一个辅助变量 $q = 1$ ， $S_q = 1$ ，且 $i = 2$ 。

步骤2 令 $t_{fi}' = \max(t_{ai}, t_{fi-1}' + \Delta t_{i-1})$ ， $D_i' = t_{di} - t_{fi}'$ 。若 $t_{ai} > t_{fi-1}' + \Delta t_{i-1}$ ，则令 $G_q = t_{ai} - (t_{fi-1}' + \Delta t_{i-1})$ ， $q = q + 1$ ，且 $S_q = i$ 。

步骤3 如果 $i = M$ ，令 $Q = q$ 并退出；否则令 $i = i + 1$ ，并转至步骤2。

在上述求解任务调度属性参数过程中，将任务请求队列分成了 Q 个序列，且 S_q 表示第 q 序列中的第1个任务请求、 G_q 表示前后两个序列之间的最大时间延迟。

对于任务 T_i ， D_i' 表示其截止期和最早可能执行时刻之间的差值。用 $\{D_i'\}_{i=1}^M$ 来判定任务请求队列的可行性：如果对于 $\forall i \in \{1, 2, \dots, M\}$ ，都有 $D_i' \geq 0$ 成立，则判定 $\{T_1, T_2, \dots, T_M\}$ 是可行请求队列；否则判定 $\{T_1, T_2, \dots, T_M\}$ 不可行。

3.4 任务队列筛选操作

对于不可行的任务请求队列 $\{T_1, T_2, \dots, T_M\}$ ，要对其进行筛选操作，生成一个新的可行队列。筛选操作流程如下：

步骤1 将原任务序列按照最大价值降序排列得到： $\{O_1, O_2, \dots, O_M\}$ 。

步骤2 定义一个任务集合，并将价值最大的任务置于集合中： $C = O_1$ ，且令 $i = 2$ 。

步骤3 将 O_i 置于 C 中，其在 C 中的位置 n 满足： $t_{eC_{n-1}} \leq t_{eO_i} \leq t_{eC_{n+1}}$ 。

步骤4 对 C 进行任务调度属性参数计算，判断其是否可行。如果可行，则新的任务集合被保留；如果不可行，则对 O_i 进行分析，确定其是被延迟或删除。

步骤5 若 $i = M$ ，结束并输出可行集合 C ；否则令 $i = i + 1$ ，转至步骤3。

经过筛选操作，最终得到 $\{O_1, O_2, \dots, O_M\}$ 的一个可行子集，任务最大价值在该过程中起决定作用。 O_1 一定会被保留，对于其他任务，最大价值越大，对其进行筛选时 C 中任务数目越少，就有更大的可能性被保留。

3.5 任务执行时刻优化分配

3.5.1 任务价值函数

任务执行时刻变化会导致目标跟踪精度的不同，也必然会对任务的实现价值产生影响。因此任务的价值应该是随实际执行时刻动态变化的。本文给每个跟踪任务赋予一个价值函数 $V_i(t_{si})$ ，大小由实际执行时刻 t_{si} 决定

$$V_i(t_{si}) = V_i^* + c_i(t_{si} - t_{ei}) \quad (6)$$

$$c_i = \begin{cases} \delta_i, & t_{ai} \leq t_{si} \leq t_{ei} \\ -\theta_i, & t_{ei} < t_{si} \leq t_{di} \end{cases} \quad (7)$$

该函数中，实际执行时刻在时间窗 $[t_{ai}, t_{di}]$ 内变动，执行时刻越靠近期望时刻，得到的回波信噪比越大且跟踪精度越高，进而相应的任务实现价值越大。

3.5.2 优化分配模型

任务执行时刻优化分配模块的输入是一个可行

的任务请求队列。假设输入队列为 $\{C_i\}_{i=1}^N$ ($N \leq M$), 其相关参数 $\{t_{fi}'\}_{i=1}^N, \{D_i'\}_{i=1}^N$ 已经计算得到。 t_{fi}' 是 C_i 的最早可能执行时刻, 则所有任务的可能执行时刻可以表示为

$$t_i' = t_{fi}' + y_i, \quad \forall i = 1, 2, \dots, N \quad (8)$$

其中, y_i 是相对于最早可能执行时刻的延迟, 要满足以下条件

$$0 \leq y_i \leq D_i' \quad (9)$$

$$y_i \leq y_{i+1} + z_i \quad (10)$$

其中,

$$z_i = \begin{cases} 0, & C_i \text{ 和 } C_{i+1} \text{ 在同一序列中} \\ G_q, & C_i \text{ 位于序列 } q \text{ 而 } C_{i+1} \text{ 位于} \\ & \text{序列 } q+1 \end{cases} \quad (11)$$

C_i 在 t_i' 时刻的实现价值为

$$\begin{aligned} V_i(t_i') &= V_i(t_{fi}' + y_i) \\ &= \begin{cases} V_i^* - \delta_i(t_{ei} - t_{fi}') + \delta_i(y_i), & 0 \leq y_i \leq t_{ei} - t_{fi}' \\ V_i^* - \delta_i(t_{ei} - t_{fi}') + \delta_i(t_{ei} - t_{fi}') - \theta_i(y_i + t_{fi}' - t_{ei}), & t_{ei} - t_{fi}' \leq y_i \leq D_i' \end{cases} \\ &= \begin{cases} V_i(t_{fi}') + \delta_i(y_i), & 0 \leq y_i \leq t_{ei} - t_{fi}' \\ V_i(t_{fi}') + \delta_i(y_i) - (\delta_i + \theta_i) \cdot (y_i + t_{fi}' - t_{ei}), & t_{ei} - t_{fi}' \leq y_i \leq D_i' \end{cases} \end{aligned} \quad (16)$$

因此, 当 $t_{fi}' \leq t_{ei}$ 时 $f_i(y_i)$ 表达式为

$$f_i(y_i) = \begin{cases} \delta_i(y_i), & 0 \leq y_i \leq t_{ei} - t_{fi}' \\ \delta_i(y_i) - (\delta_i + \theta_i)(y_i + t_{fi}' - t_{ei}), & t_{ei} - t_{fi}' \leq y_i \leq D_i' \end{cases} \quad (17)$$

进一步, 可将其表示为

$$f_i(y_i) = \delta_i(y_i) - (\delta_i + \theta_i)s_i \quad (18)$$

其中,

$$\begin{aligned} s_i &= \begin{cases} 0, & 0 \leq y_i \leq t_{ei} - t_{fi}' \\ y_i + t_{fi}' - t_{ei}, & t_{ei} - t_{fi}' \leq y_i \leq D_i' \end{cases} \\ &= \max(0, y_i + t_{fi}' - t_{ei}) \end{aligned} \quad (19)$$

综合式(15), 式(18), 式(19), 得到完整的 $f_i(y_i)$ 表达式为

$$f_i(y_i) = \begin{cases} \delta_i(y_i) - (\delta_i + \theta_i) \cdot s_i, & t_{fi}' \leq t_{ei} \\ -\theta_i(y_i), & t_{fi}' > t_{ei} \end{cases} \quad (20)$$

根据以上分析, 得到基于价值优化的任务调度模型为

$$\left. \begin{aligned} \max \bar{V} &= \sum_{i=1}^N [V_i(t_i') + f_i(y_i)] \\ \text{s.t.} \quad & C_1: 0 \leq y_i \leq D_i' \\ & C_2: y_i \leq y_{i+1} + z_i \\ & C_3: s_i - (y_i + t_{fi}' - t_{ei}) \geq 0 \end{aligned} \right\} \quad (21)$$

其中, $\{t_{fi}'\}_{i=1}^N$ 经过筛选操作已得到且任务价值函数已

$$V_i(t_i') = V_i(t_{fi}') + f_i(y_i) \quad (12)$$

其中, $f_i(\cdot)$ 表示一个价值变化函数。

所有任务的总实现价值表示为

$$\bar{V} = \sum_{i=1}^N V_i(t_i') = \sum_{i=1}^N [V_i(t_{fi}') + f_i(y_i)] \quad (13)$$

以该总实现价值最大为目标, 进行任务执行时刻的优化分配。

当 $t_{fi}' > t_{ei}$ 时有

$$\begin{aligned} V_i(t_i') &= V_i(t_{fi}' + y_i) \\ &= V_i(t_{ei}) - \theta_i(y_i + t_{fi}' - t_{ei}) \\ &= V_i(t_{fi}') - \theta_i(y_i) \end{aligned} \quad (14)$$

此时 $f_i(\cdot)$ 表示为一个线性函数

$$f_i(y_i) = -\theta_i(y_i) \quad (15)$$

当 $t_{fi}' \leq t_{ei}$ 时有

知(式(6), 式(7)), 各任务在最早可能执行时刻的价值 $V_i(t_{fi}')$ 可以求出。因此, 该调度模型的实质是选择约束条件下最优的 $\{\hat{y}_i\}_{i=1}^N$, 使得 $\sum_{i=1}^N f_i(y_i)$ 达到最大

$$\{\hat{y}_i\}_{i=1}^N = \arg \max \sum_{i=1}^N f_i(y_i) \quad (22)$$

这是一个线性规划问题, 可运用简单的求解方法(例如单纯形法)进行求解。最终得到的任务实际调度时刻可以表示为

$$t_{si} = t_{fi}' + \hat{y}_i, \quad \forall i = 1, 2, \dots, N \quad (23)$$

该价值优化模型能够使各任务实际执行时刻在满足条件下最靠近期望值, 因此该模型的建立体现了任务调度及时性的原则。

4 搜索任务调度

假设搜索功能包含 P 个任务请求, 搜索任务优先级一致, 且驻留时间都为 d ; 此外, 有 Q 个空闲时间片按先后顺序构成集合 $\{\pi_q\}_{q=1}^Q$, L_q 为时间片长度。在本文中, 搜索任务属于常驻请求, 根据队列思想, 采用先入先出(First In First Out, FIFO)的方式进行调度^[17]。其流程如图3所示。

具体描述如下：

步骤 1 设各搜索任务距上一次被调度的等待时间为 ε ，所有任务按照 ε 由大到小构成队列： $\{\lambda_{hm}\}_{m=1}^P$ ，其中 h 为队列更新次数编号。队列更新次数指针初始化为： $h = 1$ 。

步骤 2 若 $Q=0$ ，则保持 $\{\lambda_{hm}\}_{m=1}^P$ 不变，等待下一调度间隔；若 $Q \geq 1$ ，则将时间片指针初始化为： $q = 1$ ，转步骤3。

步骤 3 对 π_q 的时长 L_q 进行检验，若满足 $L_q < d$ ，则保持 $\{\lambda_{hm}\}_{m=1}^P$ 不变，令 $q = q + 1$ ，转步骤4；若满足 $kd \leq L_q < (k+1)d$ ，则 $\{\lambda_{hm}\}_{m=1}^P$ 中前 k 个任务被成功调度，并按照原顺序产生新的任务请求加入队列末尾，形成新任务请求队列 $\{\lambda_{hm}\}_{m=1}^P$ ， $h = h + 1$ ，原队列中的 λ_{k+1} 成为新队列的 λ_1 ，且令 $q = q + 1$ ，转步骤4。

步骤 4 判断 $q > Q$ 是否满足，若不满足则转步骤3；若满足，则完成次间隔的搜索任务调度，输出 $\{\lambda_{hm}\}_{m=1}^P$ 等待下一调度间隔。

5 仿真验证

5.1 评价指标

通过以下指标来评判算法的性能优劣：

(1) 调度成功率^[3,4]：成功调度的任务与请求调度的任务数目之比

$$SSR = X_{\text{suc}} / X_{\text{tot}} \quad (24)$$

其中， X_{suc} 表示调度成功的任务总数， X_{tot} 表示请求调度的任务总数。

(2) 时间偏移率^[14,15]：成功调度任务的实际执行时刻相对于期望时刻的偏移程度，用以反映任务调度的及时性

$$ATSR = \frac{1}{X_{\text{suc}}} \sum_{i=1}^{X_{\text{suc}}} \frac{|t_{si} - t_{ei}|}{t_{di} - t_{ai}} \quad (25)$$

(3) 实现价值率：常用的实现价值率指标^[3,4,14,15]将各任务的优先级直接作为任务价值，用成功调度任务的总价值与所有请求任务的总价值相比得到指标值。这只能反映出任务调度成功率以及满足重要性原则的情况，而不能体现任务及时性对实现价值率的动态影响。因此根据文中动态价值函数，对实现价值率进行改进

$$HVR = \sum_{i=1}^{X_{\text{suc}}} V_i(t_{si}) / \sum_{i=1}^{X_{\text{tot}}} V_i^* \quad (26)$$

5.2 参数设置

仿真中对本文算法、MEDF算法^[5]、HPEDF算法^[5]、任务交错算法^[11]的调度性能进行对比。调度间隔50 ms，仿真时长25 s，以目标总数来代表不同的雷达负载情况。目标数量设为：10, 20, ..., 80批，高中低价值的目标跟踪任务占比为5:3:2，不同价值类型任务的参数如表1所示。每增加10批目标，进行100次蒙特卡洛仿真，取实验平均值。每一个跟踪采样的探测概率设置为1。搜索任务为常驻任务，驻留时间为4 ms，搜索间隔4 s，依编排好的1000个波位构成队列进行调度，仿真开始后按照更新周期产生。此外，搜索任务优先级设为1。

表 1 跟踪任务参数设置

跟踪任务类别	P	V^*	$\delta = \theta$	时间窗(ms)	$\Delta t(\text{ms})$	t_i
高价值	4	1000	35	30	2	300 ms
中价值	3	600	10	50	2	600 ms
低价值	2	200	3	60	2	1 s

5.3 仿真结果及分析

图4为4种算法的调度成功率对比。在目标数量超过20批的时候，4种算法的调度成功率都处于下降状态，但是随着目标数量的增加，MEDF算法的调度成功率下降速度最快，而HPEDF算法、任务交错算法和本文算法相对于MEDF优势明显。在目

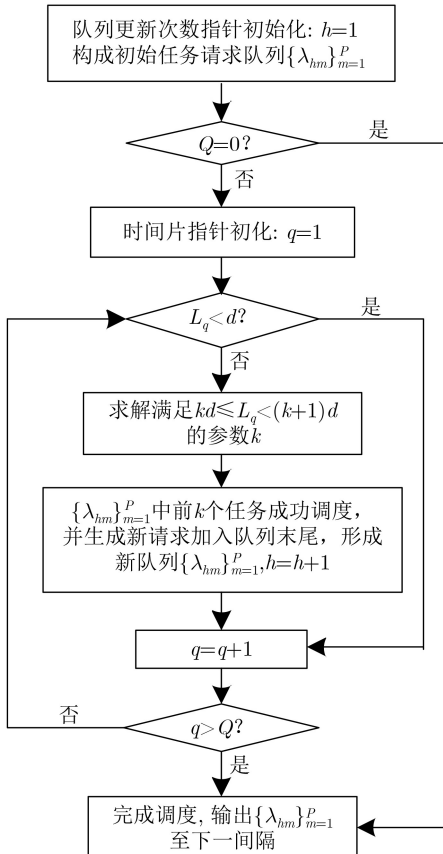


图 3 搜索任务调度框架

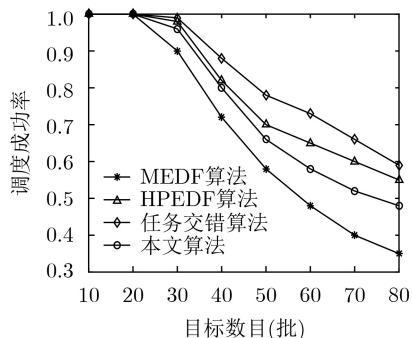


图4 调度成功率对比

标数目为80批的时候，相对于MEDF算法的调度成功率，本文算法提升了13%，而HPEDF提高了20%，任务交错算法提高了24%。这是由于HPEDF算法能够在尽量减少删除搜索任务的情况下调度更多跟踪任务，而任务交错算法可以充分利用任务驻留等待期，使得其调度成功率相对更高。

图5为4种算法的时间偏移率对比。MEDF、HPEDF和任务交错算法都没有考虑到及时性因素，随目标数目增加时间偏移率上升速率较快。此外，MEDF优先对截止期临近的任务进行调度，导致时间偏移率最高；任务交错算法为确保交错成功率，会减小执行时刻的偏移程度，其时间偏移率相对HPEDF略低。而本文算法注重任务及时性的满足，在尽可能调度多的跟踪任务基础上，使任务贴近期望执行时刻被调度。因此，本文算法时间偏移率优于另3种算法，当目标数目80批时，相对于MEDF、HPEDF、任务交错算法分别减小了48%、46%和42%。

图6为4种算法的实现价值率对比。结合图4—图6可以看出，当调度成功率为1的时候，实现价值率并不一定为1，这是由于本文中的任务实现价值率指标不仅与成功调度的任务数目有关，还受到任务及时性的动态影响。任务交错算法的调度成功率和时间偏移率指标都优于HPEDF算法，因此前者的实现价值率高于后者。此外，本文算法在调度成

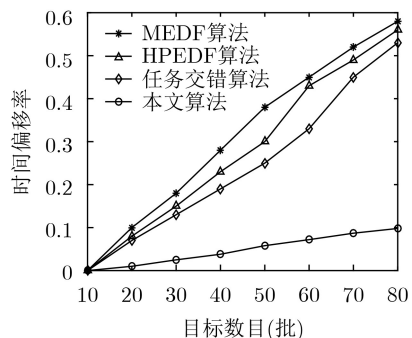


图5 时间偏移率对比

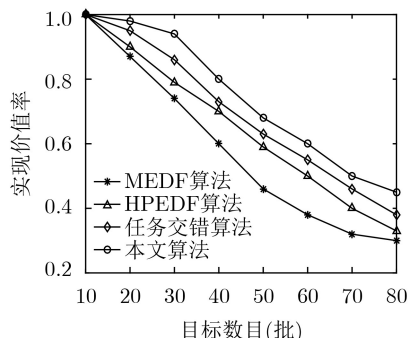


图6 实现价值率对比

功率略低于HPEDF、任务交错算法的基础上，实现价值率却比后二者更高，当目标数目80批的时候，指标分别提升了12%、7%。这是由于本文算法更能满足跟踪任务及时性要求；尤其是对于高价值任务，其时间偏移率更低。该结果说明任务及时性的变化对于实现价值率有重要影响。

图7所示为在本文算法中，不同类型任务的实现价值率对比。可以看出，在高价值跟踪任务数目和中低价值任务数目一致的情况下，本文算法对于高价值任务的实现价值率更高，在80批的时候达到了78%，比中低价值任务高47%。这说明本文算法在保证总体调度性能的基础上，注重对高价值任务的调度效果。

以上仿真结果表明，本文算法相比MEDF算法、HPEDF算法，任务交错算法能够更好满足任务调度及时性；且本文算法能够带来更高的任务实现价值率，对于高价值任务的调度效果尤为明显。

6 结束语

资源的优化调度是发挥和提升相控阵雷达效能的关键，本文提出一种基于价值优化的任务调度算法，所做的主要贡献和结论如下：(1)构建了关于实际执行时刻的动态任务价值函数，反映出任务价值随调度时刻的变化规律。(2)提出任务调度属性参数，并给出了求解算法，用以判断任务请求队列的可行性。(3)提出一种筛选算法对不可行的任务

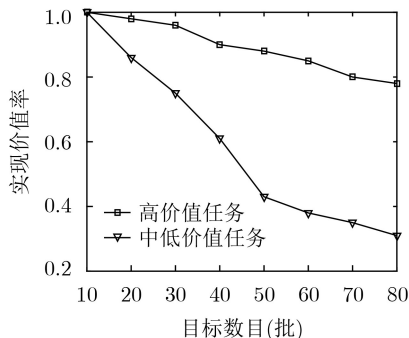


图7 不同任务实现价值率对比

请求队列进行处理, 将部分任务加入延迟或删除队列, 并得到新的可行队列。(4)构建了基于价值优化的调度模型, 使任务实际执行时刻尽可能靠近期望时刻, 满足任务调度及时性原则。(5)改进了实现价值率指标, 能更好反映出任务及时性对任务价值的影响。

参 考 文 献

- [1] BIL R and HOLPP W. Modern phased array radar systems in Germany[C]. The 2016 IEEE International Symposium on Phased Array Systems and Technology, Waltham, USA, 2016: 1–7. doi: [10.1109/ARRAY.2016.7832544](https://doi.org/10.1109/ARRAY.2016.7832544).
- [2] JIMENEZ M I, DEL VAL L, VILLACORTA J J, *et al.* Design of task scheduling process for a multifunction radar[J]. *IET Radar, Sonar & Navigation*, 2012, 6(5): 341–347. doi: [10.1049/iet-rsn.2011.0309](https://doi.org/10.1049/iet-rsn.2011.0309).
- [3] LU Jianbin, XIAO Hui, Xi Zemin, *et al.* Multifunction phased array radar resource management: Real-time scheduling algorithm[J]. *Journal of Computational Information Systems*, 2011, 7(2): 385–393.
- [4] LU Jianbin, XIAO Hui, XI Zemin, *et al.* Phased array radar resource management: Task scheduling and performance evaluation[J]. *Journal of Computational Information Systems*, 2013, 9(3): 1131–1138.
- [5] 卢建斌. 相控阵雷达资源优化管理的理论与方法[D]. [博士学位论文], 国防科学技术大学, 2007: 134–139.
LU Jianbin. Theory and method of resource optimization and management for phased array radars[D]. [Ph.D. dissertation], National University of Defense Technology, 2007: 134–139.
- [6] ZHANG Haowei, XIE Junwei, ZONG Binfeng, *et al.* Dynamic priority scheduling method for the air-defence phased array radar[J]. *IET Radar, Sonar & Navigation*, 2017, 11(7): 1140–1146. doi: [10.1049/iet-rsn.2016.0549](https://doi.org/10.1049/iet-rsn.2016.0549).
- [7] JANG D S, CHOI H L, and ROH J E. A time-window-based task scheduling approach for multi-function phased array radars[C]. 2011 International Conference on Control, Automation and Systems, Gyeonggi-do, South Korea, 2011: 1250–1255.
- [8] ORMAN A J, POTTS C N, SHAHANI A K, *et al.* Scheduling for a multifunction phased array radar system[J]. *European Journal of Operational Research*, 1996, 90(1): 13–25. doi: [10.1016/0377-2217\(95\)00307-X](https://doi.org/10.1016/0377-2217(95)00307-X).
- [9] SGAMBATO P, CELENTANO S, DI DIO C, *et al.* A flexible on-line scheduling algorithm for multifunctional radar[C]. 2016 IEEE Radar Conference, Philadelphia, USA, 2016: 1–5. doi: [10.1109/RADAR.2016.7485113](https://doi.org/10.1109/RADAR.2016.7485113).
- [10] CHENG Ting, HE Zishu, and LI Huiyong. Adaptive dwell scheduling for digital array radar based on online pulse interleaving[J]. *Chinese Journal of Electronics*, 2009, 18(3): 574–578.
- [11] 叶朝谋, 丁建江, 俞志强, 等. 基于周期分区的相控阵雷达任务交叉调度研究[J]. 电子与信息学报, 2014, 36(2): 435–440. doi: [10.3724/SP.J.1146.2013.00475](https://doi.org/10.3724/SP.J.1146.2013.00475).
- YE Chaomou, DING Jianjiang, YU Zhiqiang, *et al.* Study on task interleaving scheduling of phased array radar based on period division[J]. *Journal of Electronics & Information Technology*, 2014, 36(2): 435–440. doi: [10.3724/SP.J.1146.2013.00475](https://doi.org/10.3724/SP.J.1146.2013.00475).
- [12] CHEN Jie, TIAN Zhong, Wang Lei, *et al.* Adaptive simultaneous multi-beam dwell scheduling algorithm for multifunction phased array radars[J]. *Journal of Information & Computational Science*, 2011, 8(14): 3051–3061.
- [13] BYRNE M, WHITE K, and WILLIAMS J. Scheduling multifunction radar for search and tracking[C]. The 18th International Conference on Information Fusion, Washington, USA, 2015: 945–952.
- [14] 张浩为, 谢军伟, 张昭建, 等. 基于混合自适应遗传算法的相控阵雷达任务调度[J]. 兵工学报, 2017, 38(9): 1761–1770.
ZHANG Haowei, XIE Junwei, ZHANG Zhaojian, *et al.* Task scheduling of phased array radar based on hybrid adaptive genetic algorithm[J]. *Acta Armamentarii*, 2017, 38(9): 1761–1770.
- [15] ZHANG Haowei, XIE Junwei, LU Wenlong, *et al.* A scheduling method based on a hybrid genetic particle swarm algorithm for multifunction phased array radar[J]. *Frontiers of Information Technology & Electronic Engineering*, 2017, 18(11): 1806–1816. doi: [10.1631/FITEE.1601358](https://doi.org/10.1631/FITEE.1601358).
- [16] 胡子军, 翟海涛. 基于任务驱动的机载相控阵雷达TAS调度算法[J]. 系统工程与电子技术, 2017, 39(3): 536–541. doi: [10.3969/j.issn.1001-506X.2017.03.12](https://doi.org/10.3969/j.issn.1001-506X.2017.03.12).
HU Zijun and ZHAI Haitao. Task-driven TAS scheduling algorithm for airborne phased array radar[J]. *Systems Engineering and Electronics*, 2017, 39(3): 536–541. doi: [10.3969/j.issn.1001-506X.2017.03.12](https://doi.org/10.3969/j.issn.1001-506X.2017.03.12).
- [17] JIMÉNEZ M I, IZQUIERDO A, VILLACORTA J J, *et al.* Analysis and design of multifunction radar task schedulers based on queue[C]. The 28th IEEE/AIAA Digital Avionics Systems Conference, Orlando, USA, 2009: 6.B.3–1–6.B.3–9. doi: [10.1109/DASC.2009.5347448](https://doi.org/10.1109/DASC.2009.5347448).

杨善超: 男, 1992年生, 博士生, 研究方向为相控阵雷达资源管理、预警情报组网。

田康生: 男, 1963年生, 教授, 博士生导师, 研究方向为相控阵雷达资源管理、预警情报组网。

刘仁争: 男, 1980年生, 博士生, 研究方向为预警情报组网。

郑玉军: 男, 1988年生, 博士, 研究方向为预警情报组网。