中国科学: 信息科学 2021 年 第 51 卷 第 2 期: 247–262
SCIENTIA SINICA Informationis
论文

# 基于强化学习的舰载机保障作业实时调度方法

李亚飞, 吴庆顺, 徐明亮*, 吕培, 姜晓恒, 朱睿杰, 周兵

郑州大学信息工程学院, 郑州 450001

* 通信作者. E-mail: iexumingliang@zzu.edu.cn

收稿日期: 2020–10–12; 接受日期: 2020–11–18; 网络出版日期: 2021–01–26

国家自然科学基金 (批准号: 61972362, 62036010, 61602420, 61822701, 61772474, 61872324, 61802351)、中国博士后基金 (批准号: 2018M630836, 2018M632802)、河南省自然科学基金 (批准号: 202300410378) 和河南省重点研发与推广专项 (批准号: 192102310476) 资助

**摘要** 衡量航母作战性能的重要指标是舰载机出动架次率, 而影响舰载机出动架次率的关键因素是舰载机保障作业调度效率. 舰载机保障作业调度是指在有限时间、空间和资源约束的前提下合理安排舰载机所需保障作业顺序并高效完成舰载机的作业保障. 现有基于最优化方法 (动态规划、线性规划等) 和启发式方法 (如遗传算法、粒子群等) 的求解策略仅适用于保障作业可预知情况下的作业调度, 很难满足高动态作战场景下的实时保障作业调度需求. 基于此, 本文提出了一种新的基于 DQN (deep Q-network) 的舰载机保障作业实时调度方法, 将舰载机保障作业调度问题建模成部分可观测马尔科夫决策过程 (partially observable Markov decision processes) 问题, 利用全局与长期收益对保障作业调度过程进行优化, 并通过离线学习和在线调配的学习决策框架进行解决. 经过仿真实验验证, 该方法能显著提高舰载机保障作业调度效率并满足实时决策环境的需要.

**关键词** 舰载机, 保障作业, 实时调度, 强化学习, 仿真验证

## 1 引言

舰载机出动架次率 [1] 是衡量航母作战性能的重要指标, 而影响舰载机出动架次率的关键因素是舰载机保障作业的调度效率. 不同于陆基航空保障作业, 舰载机保障作业环境是一个由保障人员、舰载机和保障车辆等多类型异质群组构成的高动态、非完备、强实时、紧耦合的复杂 “人 – 机 – 车” 混合运动系统, 在不足陆上机场十分之一面积的飞行甲板上密集作业, 保障作业过程以多型、多架、多批次舰载机为保障对象, 在作业空间高度受限、设备种类繁多、舰面 “人 – 机车 – 物资” 分布混杂等严苛条件下高效协作与配合, 保证舰载机能够在飞行甲板上安全、持续、频繁地进行起飞、着舰、回收、调运、维修等复杂保障作业, 稍有差池就会造成整个航母甲板作业混乱, 甚至导致整个航母作战能力的丧失.

**引用格式:** 李亚飞, 吴庆顺, 徐明亮, 等. 基于强化学习的舰载机保障作业实时调度方法. 中国科学: 信息科学, 2021, 51: 247–262, doi: 10.1360/SSI-2020-0316
Li Y F, Wu Q S, Xu M L, et al. Real-time scheduling for carrier-borne aircraft support operations: a reinforcement learning approach (in Chinese). Sci Sin Inform, 2021, 51: 247–262, doi: 10.1360/SSI-2020-0316

 

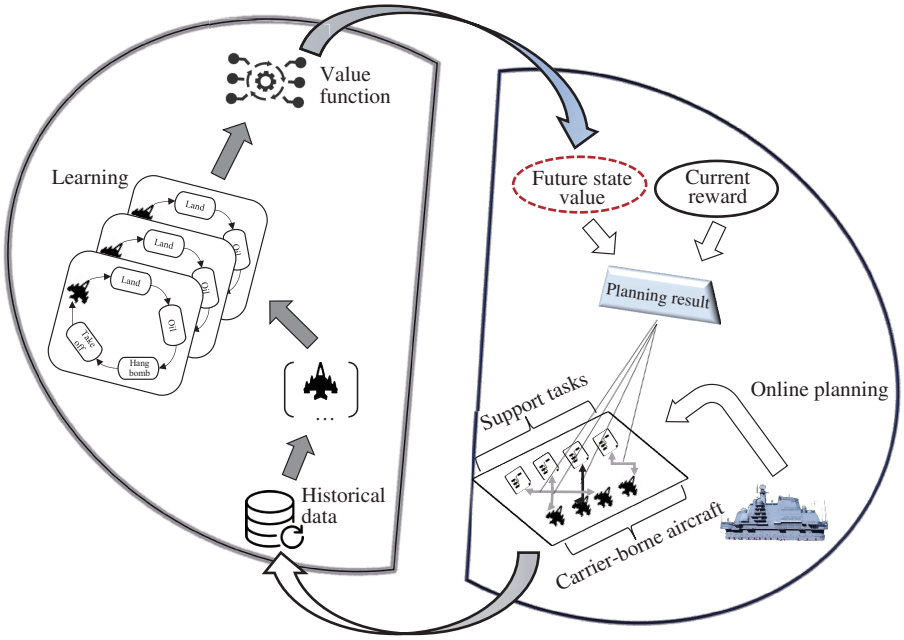


**图 1 (网络版彩图) 基于强化学习的保障作业实时调度框架**

**Figure 1** (Color online) Real-time scheduling framework for support tasks based on reinforcement learning

现有舰载机保障作业调度多采用最优化 (线性规划 [2]、动态规划 [3] 等) 和启发式 (模拟退火 [4]、遗传算法 [5] 等) 的求解方法. 其中, 最优化方法虽然能够求解出最佳作业调度序列, 但其在求解过程中需要枚举搜索空间中所有可行解, 导致算法时间复杂度非常高. 特别当作业保障需求规模较大时, 最优化方法很难在有限时间内找到最优解; 启发式方法是一种随机性较强的算法, 其基本思想是随机地产生作业调度序列, 并依据当前调度代价对作业调度序列进行迭代更新, 虽然启发式方法在求解效率上有所提升, 但其生成的作业调度序列质量很难得到保障. 此外, 上述两类方法通常适应于保障作业可预知的情形下, 但在实际作战场景中, 保障作业调度策略往往需要根据实时环境及时地进行动态调整, 上述方法过多地依赖代价模型的设计, 而不接受实际运行时的环境实时反馈, 使最终调度策略的执行效果很难满足实时场景需求.

强化学习 [6] 是一个通用的问题解决框架, 其中智能体 (agent) 以 "试错" 的方式在环境中进行探索, 同时获得奖励来指导行为, 它能够很好地从当前执行结果及环境的反馈中学习并不断优化决策质量, 特别适用于解决复杂动态的作业序列决策问题. 基于此, 本文将舰载机保障作业调度问题建模为一个基于部分可观测马尔可夫决策过程 (partially observable Markov decision processes, POMDP) 的顺序决策问题, 并提出了一种基于 DQN (deep Q-network) [7,8] 的舰载机保障作业实时调度算法, 从全局和长远的角度来优化舰载机保障作业调度, 该算法显著地获得了良好的实时调度效果. 每架舰载机的保障作业与对应保障战位匹配的决策都基于两个条件: (1) 从实时环境中获得选取此保障战位的即时奖励, 以及对未来产生的长期收益; (2) 根据历史学习中不同状态的舰载机所选取的最佳战位, 建立了统一的时空状态评估指标, 量化上述长期收益. 在舰载机保障作业调度系统中, 舰载机保障作业和保障战位之间的实时匹配被描述为一个决策问题, 并通过组合优化算法进行求解. 实时调度系统整体架构如图 1 所示.

本文仿真实验结果表明所提算法在保障作业实时调度场景中性能表现优良. 相比于传统最优化和

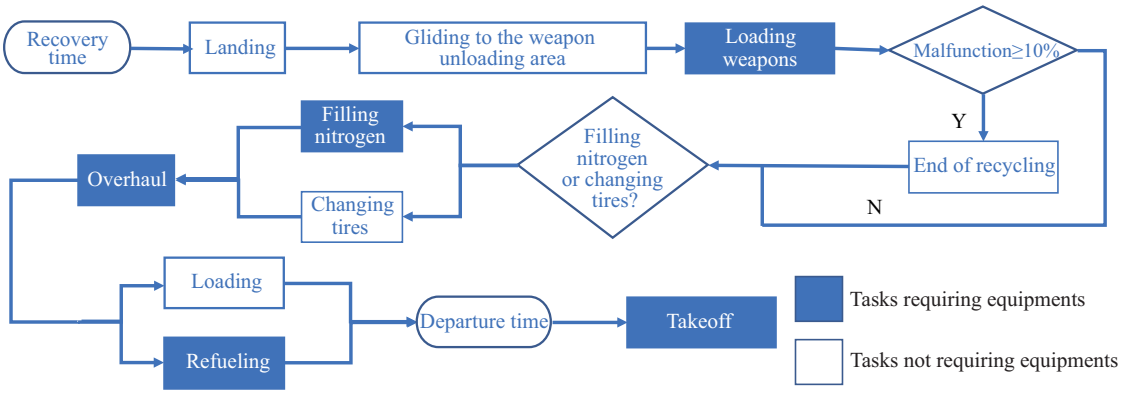


**图 2　(网络版彩图) 常见舰载机保障作业流程**

**Figure 2**　(Color online) Common process of carrier-borne aircraft support operation

启发式方法, 本文所提算法在决策效率方面提高约 1.2 个数量级. 概括来说, 本文主要贡献如下:

• 提出了一种高效的舰载机保障作业调度算法, 该算法同时考虑保障作业即时奖励和长期收益, 能够对作业调度的长期效用进行良好的优化;

• 将保障作业实时调度建模为一个集中控制的顺序决策问题, 并在 DQN 学习规划框架下实现求解, 这是 DQN 模型在大规模舰载机保障作业调度中的首次应用;

• 在仿真数据集上的实验验证结果表明, 本文所提出的基于 DQN 的调度算法能够有效满足保障作业实时调度场景的需要.

其余章节安排如下: 第 2 节概述了舰载机保障作业调度问题及场景设定; 第 3 节介绍了保障作业的策略学习; 第 4 节介绍了保障作业在线调配; 第 5 节对本文所提算法进行了验证与分析; 第 6 节讨论了相关研究工作; 第 7 节对论文进行了总结与展望.

## 2　问题概述

舰载机的保障作业主要在航母甲板上进行, 整个舰面可以分为若干个保障战位, 每个保障战位配置零个或多个保障资源, 其中具有零个保障资源的保障站点可以进行舰载机停靠或进行非资源占用的保障作业. 舰载机可以在甲板上进行起飞、降落、油气电补给、检修、系留等一系列保障作业, 这些作业以及作业间的依赖关系如图 2 所示. 本文聚焦于舰载机在动态场景下的作业调度, 以最小化舰载机保障作业周期为主要优化目标, 通过设计高效算法来解决动态实时场景下的舰载机保障作业调度问题. 本研究相关的若干概念如下所示.

**定义1** (保障资源)　一个保障资源可被表示为一个二元组 $r=(c,H)$, 其中 $c$ 是资源类型, $H$ 为资源等待序列.

在实际应用中, 保障资源对于不同舰载机的保障作业来说通常是互斥的, 一个保障资源一旦分配给某一舰载机进行保障作业, 其他舰载机仅能排队等候该资源直到释放. 一个保障资源可以被多个保障战位共享, 一种类型的保障资源可以分布在一个或多个保障战位上.

**定义2** (保障作业)　一个保障作业可被表示为一个二元组 $\tau=(w,c)$, 其中 $w$ 表示作业执行时长, $c$ 表示该保障作业需要的保障资源类型.

通常一架舰载机需要完成全部保障作业才可进入起飞就绪状态, 但实际中舰载机需进行多波次飞

行, 这意味着不一定每次都需要进行全部作业保障, 舰载机根据实际所需资源向系统提交相应的作业保障请求.

**定义3** (保障战位) 一个保障战位可被表示为一个三元组 $z=(W,l,k)$, 其中 $W$ 表示该战位拥有的资源类型, $l$ 表示该保障战位的位置, $k$ 表示该保障战位的容量 (即可容纳舰载机的数量).

通常保障战位具有容量约束, 一旦一个保障战位停靠的舰载机数量超出容量限制, 则该保障战位不能再接纳新的舰载机进行停靠.

**定义4** (作业环境) 航母甲板作业环境定义为一个加权图模型 $G=(Z,E,C)$, 其中 $z\in Z$ 表示一个保障战位, $e_{ij}\in E$ 表示连接保障战位的一条边, 边 $e_{ij}$ 的权重 $c_{ij}\in C$ 表示保障战位之间的移动代价.

**定义5** (作业调度) 给定一组保障作业集合 $\Gamma$ 和一组保障战位集合 $Z$, 系统根据每个保障作业 $\tau\in\Gamma$ 所需 $Z$ 中资源状态, 为每个舰载机安排合适的保障战位和时间片进行作业保障. 作业调度的最终目标为最小化所有舰载机保障作业完成时间.

需要注意的是, 航母舰载机调度通常按飞行波次进行, 每个飞行波次为一个飞行调度周期. 本文聚焦于一个调度周期中所有舰载机保障作业和保障战位及其资源的优化分配.

## 3 策略学习

本节主要介绍舰载机保障作业匹配策略的学习, 本文将保障作业实时匹配问题建模为一个部分可观测马尔可夫决策过程 (POMDP) 问题, 利用在历史作业匹配决策中获得的值函数为每个时空状态产生对应的状态值, 为后续保障作业匹配的在线决策提供支持.

### 3.1 模型定义

在 POMDP 模型中, 智能体根据当前系统状态和既定动作策略进行动作选择, 并从环境状态改变中获得相应奖励, 这些奖励随时间而被积累为序列决策回报. POMDP 模型通常被定义为一个七元组 $\Gamma=\langle S,A,P(s'|s,a),R(s,a),\Omega,O,\gamma\rangle$. 其中, $S$ 表示一组有限状态集, $A$ 表示一组有限动作集, $P(s'|s,a)$ 表示在状态下执行动作之后转移到状态的概率, $R(s,a)$ 表示在状态下执行动作的奖励, $\Omega$ 表示一组观察结果集 (即智能体感知到的环境数据), $O$ 表示条件观察概率 (即智能体在观察到环境数据时自身处于状态的概率), $\gamma$ 表示在 [0,1] 区间取值的衰减因子. 本文假定航母甲板上所有舰载机为一组同构智能体集合 $F=\{f_1,f_2,\ldots,f_n\}$, 舰面资源状态及舰面作业保障战位的状态为舰载机作业保障的全域环境, 舰载机所处保障战位及当前正在保障或待保障作业为舰载机作业保障的局域环境. POMDP 模型参数定义如下:

**(1) 状态 $S$.** 每个舰载机的状态可以用一个包含舰载机所处战位以及进行何种保障作业的二维向量表示. 具体来说, 舰载机状态可定义为一个二元组 $s=(z,\tau)\in S$, 其中 $z$ 和 $\tau$ 分别表示当前舰载机所处的战位和当前舰载机进行的保障作业. 给定一个保障战位集合 $Z$ 和保障作业集合 $\Gamma$, $|S|=|Z|\times|\Gamma|$.

**(2) 动作 $A$.** 本文中舰载机可执行的动作分为空闲保持和作业执行两类. 在作业状态下, 舰载机会被分配去进行一项待保障的作业直到该项保障作业完成, 然后获得执行该保障作业的奖励. 在空闲状态下, 舰载机因其所需资源被占用无法进行作业保障, 这就导致该舰载机在一定时间内无法前往任何一个保障战位. 在保持空闲状态下, 我们假定舰载机可以直接进行状态转移, 下一个状态与当前状态保持一致. 舰载机在每个时间片可选择的动作集合记为 $A_i$.

**(3) 状态转移概率 $P$.** 舰载机在每个时间片选择的动作记为 $a_i \in A_i$, 形成一组联合动作集 $a^t = A_1 \times \cdots \times A_N$, 根据状态转移函数在环境中产生转移 $P(s^{t+1}|s^t, a^t) : S \times A_1 \times \cdots \times A_N \to S$.

**(4) 奖励 $R$.** $R$ 为舰载机在时间步 $t$ 成功执行动作 $a_i$ 后, 状态由 $s_i$ 转变为 $s_{i+1}$ 获得的奖励.

**(5) 条件观察概率 $O$.** 舰载机在时间步 $t$ 根据观测函数 $S \times F \to O$ 产生与真实环境状态关联的观测值.

**(6) 折扣因子 $\gamma$.** 折扣因子主要用来设置 POMDP 能够向未来看多远. 在舰载机保障作业调度场景中, 我们使用较小的折扣因子来应对长视野导致值函数变化大的问题. 需要注意一个保障作业的执行要跨 $T$ 个时间步, 在折扣因子的影响下, 最终收到的奖励为 $R_\gamma = \sum_{t=0}^{T-1} \gamma^t \frac{R}{T}$.

### 3.2 模型学习

针对上述建立的 POMDP 模型, 本文基于 DQN 算法提出了适应于作业实时调度场景的 S-DQN (strategy-deep Q-network) 算法, 算法伪代码如算法 1 所示.

**Algorithm 1** Task scheduling strategy learning algorithm

**Input:** Current state $s_i$, action $a_i$, reward $R$, next state $s_{i+1}$.

**Output:** Next action $a_{i+1}$.

1: Initialize the environment and target network;
2: **while** training is not completed **do**
3: **#Step 1: collect experience**;
4: Randomly choose the next action $a_{i+1}$ or $a_{i+1} = \max_a Q^*(\emptyset(s_i, a_i, R, s_{i+1}))$;
5: Calculate the value of the target network and evaluation network, and store the transformation in model $M_Q$;
6: **#Step 2: update parameters**;
7: Sample a batch of experience $\{s_i, a_i, R, s_{i+1}\}$;
8: Update the target network and evaluation network;
9: **end while**

**Return:** $a_{i+1}$.

算法 1 的输入为一组状态值 $\Pi = \{s_i, a_i, R, s_{i+1}\}$, 其中 $s_i$ 为舰载机当前状态, $s_{i+1}$ 为舰载机下一状态, $a_i$ 为所选取的动作, $R$ 为执行 $a_i$ 后状态从 $s_i$ 转变为 $s_{i+1}$ 所获奖励, 输出为下一步动作 $a_{i+1}$. 该算法首先初始化全局环境和目标网络 (第 1 行), 接下来进行两阶段操作: 储存经验数据 (第 3~5 行) 和更新网络参数 (第 6~8 行). 算法在第 4 行动作选择中以 $1-\gamma$ 的概率随机选择一个动作值 $a_{i+1}$ 或根据记忆储存库和最大化公式 $\max_a Q^*$ 来决策可使总体收益最大化的下一个动作 $a_{i+1}$, 其中 $\emptyset$ 为归一化操作. 此外, 每隔一段时间它便将决策所获得的经验数据更新到内在记忆储存库. 这个迭代过程便是舰载机保障作业实时匹配策略学习过程.

### 3.3 策略评估

为了进一步优化 S-DQN 算法的策略学习效果, 本文设计了一个策略评估优化方法, 对每个保障作业分配的保障战位进行校正, 优化最终整个策略的学习效果. 该策略评估算法如算法 2 所示.

算法第 1~5 行主要通过遍历获得所有符合条件的战位集合 $Z$, 第 6 行根据从当前战位移动至所选战位的转移代价, 对所有战位按照转移代价从小到大进行排序, 得到保障战位集合 $Z'$. 最后, 第 7 行通过策略评估奖励函数 $\mathbb{R}$ 将 S-DQN 计算所得战位 $z$ 与 $Z'$ 进行匹配, 即观察 $z$ 在 $Z'$ 中的位置, 若位置靠前, 说明此战位较优, 会产生一个较大的奖励值; 若位置靠后, 则说明此战位略差, 会得到一个较小的奖励值; 通过这种策略, 来得到最终奖励值 $R_s$. 虽然多次遍历会导致算法执行效率略低, 但由于

**Algorithm 2** Strategy evaluation algorithm

**Input:** A task $\tau$ and station $Z$.

**Output:** The strategy evaluation reward $R_s$.

1: **if** $\tau$ need resource $r$ **then**
2: Find all stations containing resource $r$ and put them in set $Z$;
3: **else**
4: Find all free stations and put them in set $Z$;
5: **end if**
6: $Z' \leftarrow$ sort the stations in $Z$ by the shortest distance to the task;
7: $R_s \leftarrow$ reward function for strategy evaluation $\mathbb{R}(Z', z)$;

**Return:** $R_s$.

该算法仅在学习阶段使用, 故不影响后期在线实时调度效率.

## 4 在线决策

在线决策步骤以策略学习过程中获取的值函数为输入, 实时规划舰载机保障作业和保障战位之间的最佳匹配. 本节假设匹配过程符合第 3 节中的描述.

### 4.1 调配算法

基于第 3.2 小节的学习步骤, 最终会获得一个近似最优的状态 – 动作值函数, 即决策模型 $M_Q$. 给定一组状态值 $\Pi = \{s_i, a_i, R, s_{i+1}\}$, 我们可以计算出当前环境下的最佳动作 $a$. 调配算法伪代码如算法 3 所示. 此算法输入为一组带有保障作业的舰载机序列, 输出为保障作业调度策略. 第 1 步初始化了全局环境, 接着在第 2 步中将之前学习到的值函数 $M_Q$ 加载进算法, 随后在 3∼9 步描述了一个调度 + 学习方法的迭代过程. 在每个时间步 $t$ 中, 在线调配算法根据当前环境确定保障作业和保障战位之间的最佳匹配. 在本问题的设定中, 这个目标也可以被理解为针对每个舰载机都找到其接近最佳的动作, 以协调的方式优化未来的全局收益. 形式上, 作业调配的目标函数是

$$\arg\max \sum_{n=0}^{N} \sum_{i=0}^{I} R_\pi(n, i) \mathbb{A}_{ni}, \tag{1}$$

**Algorithm 3** Online task scheduling algorithm

**Input:** A set of tasks $T = \{\tau_1, \tau_2, \ldots, \tau_n\}$.

**Output:** The scheduling strategy $\mathbb{E}$.

1: Initialize the environment;
2: Load decision model $M_Q$;
3: **while** $|T| > 0$ **do**
4: Observe the current environment and get the status $s_i$;
5: Choose action $a_i$ based on decision model $M_Q$;
6: Execute action $a_i$ and observe reward $R$ and next state $s_{i+1}$;
7: S-DQN$(s_i, a_i, R, s_{i+1})$;
8: Update environment;
9: **end while**

**Return:** $\mathbb{E}$.

$$\begin{aligned} \text{s.t. } & \sum_{n=0}^{N} \mathbb{A}_{ni} = 1,\ i = 1, 2, \ldots, I, \\ & \sum_{i=0}^{I} \mathbb{A}_{ni} = 1,\ n = 1, 2, \ldots, N, \end{aligned} \tag{2}$$

其中, 若舰载机 $f_n$ 可分配至保障战位 $Z_i$, 则 $\mathbb{A}_{ni}$ 为 1; 否则, $\mathbb{A}_{ni}$ 为 0. 此处, $i \in \{1, 2, \ldots, I\}$ 对应于这个时间步上所有可用的保障战位编号, 而 $n \in \{1, 2, \ldots, N\}$ 对应于要服务的舰载机编号. $R_\pi(n, i)$ 是舰载机 $f_n$ 匹配保障战位 $Z_i$ 的奖励函数. 注意 $n = 0$ 和 $i = 0$ 的实例对应于一个特殊的默认操作, 该操作在此时间步中不提供任何调配. $\mathbb{A}_{ni}$ 用来表示舰载机是否可被分配至保障战位, 若 $\mathbb{A}_{ni}$ 为 0, 说明舰载机 $f_n$ 不可被分配至保障战位 $Z_i$; 反之, 若 $\mathbb{A}_{ni}$ 为 1, $f_n$ 可被分配至 $Z_i$. 值得注意的是, 此处舰载机的动作空间是受限的, 因为它们只能在可服务的保障战位中选择一个来进行作业, 或者什么也不做, 仍然停留在当前战位. 不同的是, 在学习阶段, 舰载机有一个不受限的动作空间. 式 (2) 中的约束可以保证每个舰载机都将选择一个可用的操作, 包括可服务的保障战位和不执行任何操作, 而每个保障战位最多可以分配给一架舰载机, 或者在此时间步未服务. 本文将式 (1) 表示为一个最大化全局收益函数, 其中保障战位和舰载机是两组节点; 舰载机 $f_n$ 和保障战位 $Z_i$ 之间的每次匹配的奖励为 $R_\pi(n, i)$. 关于 $R_\pi$ 的详细介绍将在 4.2 小节给出.

### 4.2 收益函数

据上文所述, 调配算法的目标是最大化 $R_\pi$, $R_\pi$ 表示在整个调配过程中所有时间步所产生的收益之和, 即 $R_\pi = \sum_{t=1}^{T} \sum_{n=1}^{N} R_n$. 其中, $R_n$ 为舰载机 $F_n$ 在时间步 $t$ 时刻所产生的全部收益. 具体为: $R_n = \alpha \cdot R_t + \beta \cdot R_p + \lambda \cdot R_s$. 其中, $\alpha$, $\beta$, $\lambda$ 为超参数, 用来控制每种奖励占比, 并且 $\alpha, \beta, \lambda \in [0, 1]$, $\alpha + \beta + \lambda = 1$.

在实际调配过程中, 每架舰载机必须要在一定时间段里完成指定的保障作业, 为了完成这一优化目标, 本文设计了时间奖励 $R_t$ 来约束学习过程. 其计算策略具体为: 若舰载机当前保障作业不需要保障资源 (即在任一空闲战位都可以完成), $R_t$ 默认为 1. 反之, 若需要保障资源 $r$, 则观察其抢占的保障资源时间是否已经超过预定时间段, 若超过, 则置为 $-1$; 未超过, 则置为 1; 若因等待所需保障资源而处于闲置状态, 则此时间步置为 0.

为了防止出现保障死锁情况的发生, 即某一架舰载机 $f_n$ 因等待某一保障资源 $r$ 而永久处于闲置状态, 本文设计了排队优先级奖励 $R_p$. 其计算策略具体为: 若舰载机 $f_n$ 在保障资源 $r$ 等待序列的第 1 位, $R_p$ 为 1; 若舰载机 $f_n$ 在保障资源 $r$ 等待序列的第 2 位或其后, $R_p$ 为 0. 在调配过程中, 若一舰载机 $f_1$ 需要保障资源 $r_1$, 但 $r_1$ 均处于被占用状态, 则需把 $f_1$ 加入到 $r_1$ 的等待序列 $\psi_1$ 中, 并给予其优先级标识 1. 此后, 若再有舰载机 $f_5$ 进入等待序列 $\psi_1$, 则给予其优先级标识加 1, 如此构建出一个形如 $\psi = \{f_1^1, f_5^2, f_2^3, f_7^4, \ldots, f_n^i\}$ 的队列. 此后, 若某一舰载机离开此队列, 则触发队列的更新策略, 即将队列中的优先级全部依次重新赋值. 如在队列 $\{f_1^1, f_5^2, f_2^3, f_7^4\}$ 中, $f_5$ 离去, 则队列自动更新为 $\{f_1^1, f_2^2, f_7^3\}$. 如在某一时间步, 调配算法为舰载机 $f_n$ 选定保障资源 $r_k$, 若 $f_n$ 在 $r_k$ 的等待序列 $\psi_k$ 中优先级标识为 1, 即 $f_n^1$, 则将 $R_p$ 置为 1; 若优先级为 2, 3, 4 等, 则将 $R_p$ 置为 $-1$. $f_n$ 若处于等待状态, 即当前时间步调配算法未为 $f_n$ 分配保障资源, 则此时间步 $R_p$ 置为 0.

通过全局收益函数的设置, 再结合 S-DQN 算法进行学习, 尽量保证可以获得一个较好状态 – 动作值函数. 使其可以针对大多数状态的输入都计算出一个接近最优的策略.

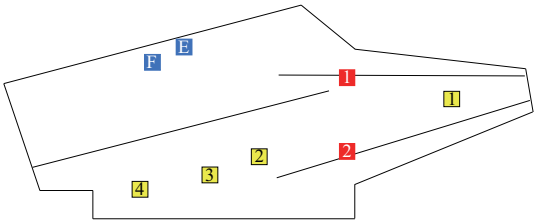


图 3 (网络版彩图) 4 机 6 战位场景示例图

**Figure 3** (Color online) A scenario of 4 machines and 6 stations

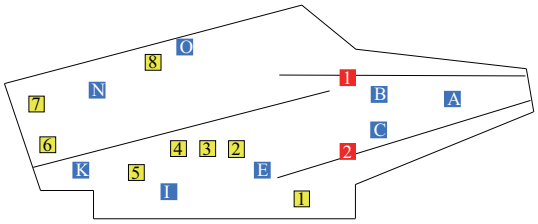


图 4 (网络版彩图) 8 机 16 战位场景示例图

**Figure 4** (Color online) A scenario of 8 machines and 16 stations

## 5 实验评估

本节主要评估本文所提算法的有效性, 在不同参数设置下将本文所提算法与遗传算法 (GA)、线性规划 (LP)、模拟退火算法 (SA) 和 Q-learning 等 4 种算法在两种调度仿真环境中进行对比.

### 5.1 仿真环境

根据舰载机保障作业的调度流程, 本文构建仿真环境进行实验验证. 此仿真环境的约束条件设置如下:

**时间约束.** 任一保障作业一旦开始, 中间不能间断, 其完工时间与开工时间之差为该作业的作业时长; 任一保障作业必须在其前序作业都完成后才能开始; 任一舰载机在不同的保障战位上连续进行的两种作业, 其转移时间由两保障战位之间的路径长度决定.

**空间约束.** 当存在空闲保障战位时, 舰载机才可以停放; 同一时刻同一个保障战位仅可停放一架舰载机.

**资源约束.** 一个保障战位仅可提供有限种保障资源; 同一时刻同一战位的一种资源仅可被一架舰载机使用.

实验主要构建了两个仿真场景, 第 1 个场景如图 3 所示, 其中 4 个黄色方块代表 4 架舰载机, 6 个蓝色方块代表 6 个保障战位, 2 个红色方块代表起飞位, 起飞位可临时停放舰载机, 但不可进行保障作业. 此时, 舰载机 1 停靠在保障战位 A 上, 舰载机 2 停靠在保障战位 B 上, 舰载机 3 停靠在保障战位 C 上, 舰载机 4 停靠在保障战位 D 上, 故图中只显示出 2 个蓝色的空闲保障战位.

为了测试扩大舰载机和保障战位规模对各种算法的影响, 故而构建了如图 4 所示的第 2 个场景, 其中包含 8 架舰载机和 16 个保障战位. 各方块含义跟第 1 个场景相同.

以下所有实验均在搭配 i7-9700k 处理器, GTX 2080Ti 显卡 (11 G 显存), 32 G 运行内存和 Ubuntu 18.04 操作系统的机器上进行, 编程语言采用了 Python, 深度学习框架采用了 Tensorflow 1.15.0.

### 5.2 对比方法与评价指标

本文主要与 GA, LP, SA 和 Q-learning 4 种算法进行了对比. 基于实际实时作业保障场景需求, 采用了决策总奖励值和作业响应时间两个指标对上述几种算法的效能进行评价. 其中, 总奖励值可以反映算法的求解质量, 作业响应时间可以反映算法的实时性.

### 5.3 实验设置

为了验证本文所提算法的求解效果, 根据实际调度环境设计了两个实例, 具体细节如下.

**表 1 舰载机保障作业详情**

**Table 1** Details of carrier-borne aircraft support tasks

| Support task | Support time (s) | Need resources |
|---|---|---|
| W1 | 1 | No |
| W2 | 3 | Yes |
| W3 | 3 | Yes |
| W4 | 3 | Yes |
| W5 | 3 | No |
| W6 | 4 | Yes |
| W7 | 4 | No |
| W8 | 4 | Yes |
| W9 | 2 | No |
| W10 | 2 | No |
| W11 | 5 | Yes |
| W12 | 6 | No |
| W13 | 6 | Yes |
| W14 | 4 | Yes |
| W15 | 1 | No |

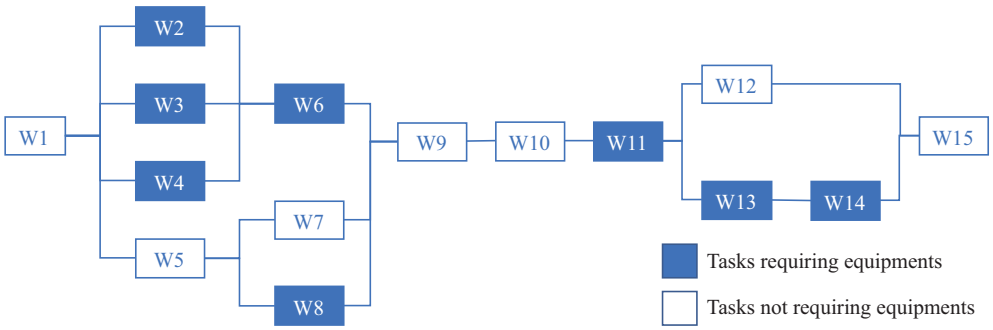


**图 5 (网络版彩图) 保障作业间序列图**

**Figure 5** (Color online) Task sequence

实例一: 假设一波次中一共有 4 架舰载机需要出动执行任务, 每架舰载机需完成表 1 中 15 种保障作业, 作业间顺序如图 5 所示, 每项保障作业有多个保障战位可以提供保障资源. 舰面上有 6 个保障战位可以正常提供保障服务, 每个战位可提供的保障资源如表 2 所示.

在此实例中, 根据表 2 所示作业顺序构建了 10 种可行作业执行序列 (如表 3), 舰载机的保障作业从这 10 种作业序列中随机生成. 然后, 利用上述提及的 5 种算法分别进行求解, 每种算法将对每种作业序列求解 100 次, 计算出 100 次的平均奖励值和平均作业响应时间.

实验中所设置的实验参数为: 奖励函数中, 分别设置为 0.3, 0.2 和 0.5; 遗传算法中, 染色体的种群规模取 100, 交叉概率为 0.8, 变异概率取 0.1, 迭代次数取 100; 线性规划中, 系数矩阵中的每个值为当前舰载机若在当前保障战位执行作业所获得的奖励, 求解目标为最大化所有舰载机的总奖励; 模拟退火算法中, 迭代次数为 50; Q-learning 中, q-table 中含 30000 条记录, 实验过程不启用学习步骤. 本文算法中, 选取训练 7000 步的模型.

表 2 航母保障战位详情 (实例 1)

**Table 2** Details of aircraft carrier support stations (example 1)

| Support station | Number of resources | Resources |
|---|---|---|
| P1 | 4 | W2, W3, W4, W11 |
| P2 | 5 | W2, W3, W6, W13, W14 |
| P3 | 5 | W4, W6, W8, W13, W14 |
| P4 | 8 | W2, W3, W4, W6, W8, W11, W13, W14 |
| P5 | 6 | W2, W3, W8, W11, W13, W14 |
| P6 | 6 | W2, W4, W6, W8, W13, W14 |

表 3 保障作业执行序列详情

**Table 3** Execution sequences of support tasks

| Sequence | Resources |
|---|---|
| S1 | W1, W2, W3, W4, W5, W6, W7, W8, W9, W10, W11, W12, W13, W14, W15 |
| S2 | W1, W2, W4, W3, W5, W6, W7, W8, W9, W10, W11, W12, W13, W14, W15 |
| S3 | W1, W3, W2, W4, W5, W6, W7, W8, W9, W10, W11, W12, W13, W14, W15 |
| S4 | W1, W2, W3, W4, W5, W6, W8, W7, W9, W10, W11, W12, W13, W14, W15 |
| S5 | W1, W2, W3, W4, W5, W6, W7, W8, W9, W10, W11, W12, W14, W13, W15 |
| S6 | W1, W2, W3, W4, W5, W6, W8, W7, W9, W10, W11, W12, W14, W13, W15 |
| S7 | W1, W4, W3, W2, W5, W6, W8, W7, W9, W10, W11, W12, W14, W13, W15 |
| S8 | W1, W3, W4, W2, W5, W6, W8, W7, W9, W10, W11, W12, W14, W13, W15 |
| S9 | W1, W3, W4, W2, W5, W6, W8, W7, W9, W10, W11, W12, W13, W14, W15 |
| S10 | W1, W3, W4, W2, W5, W6, W7, W8, W9, W10, W11, W12, W13, W14, W15 |

实例二: 假设一波次中一共有 8 架舰载机需要出动执行任务, 每架舰载机需完成如表 1 所示的 15 种保障作业, 其作业间的保障顺序如图 5 所示, 每项保障作业有多个保障战位可以提供保障资源. 此时, 甲板上 16 个保障战位可以正常提供保障服务, 其可提供的保障资源如表 4 所示. 在本实例中, 作业执行序列和实验参数的设置与实例一中一致.

### 5.4 结果分析

按照 5.3 小节的实验设置进行多组实验后, 得到了以下实验数据, 下文将依次分析这些实验结果.

**(1) 实例一结果分析.** 图 6 和表 5 分别展示了 4 机 6 战位场景的平均总奖励值对比和平均作业响应耗时对比.

从平均作业响应耗时对比图可以看出, 本文 S-DQN 方法在不同作业执行序列下曲线变化幅度平稳, 且总耗时和响应时间最少, 而 GA, LP, SA 和 Q-learning 算法曲线变化幅度较大, 说明本文 S-DQN 算法在实时性和鲁棒性方面表现优良. 从平均总奖励值对比图可以看出, 本文 S-DQN 算法所得到的解虽略低于 GA, LP 和 SA, 但却好于 Q-learning, 并且在后面实验中我们验证了随着作业数目的增多, S-DQN 的求解质量逼近 LP 最优解.

**(2) 实例二结果分析.** 图 7 和表 6 分别展示 8 机 16 战位场景的平均总奖励值对比和平均作业响应耗时对比.

从实验对比图可以看出, 扩大作业规模后, 本文所提的 S-DQN 算法仍可在实时性和鲁棒性方面

**表 4 航母保障战位详情 (实例 2)**

**Table 4** Details of aircraft carrier support stations (example 2)

| Support station | Number of resources | Resources |
|---|---|---|
| P1 | 3 | W2, W3, W11 |
| P2 | 5 | W2, W3, W6, W13, W14 |
| P3 | 3 | W6, W13, W14 |
| P4 | 8 | W2, W3, W4, W6, W8, W11, W13, W14 |
| P5 | 8 | W2, W3, W4, W6, W8, W11, W13, W14 |
| P6 | 6 | W2, W3, W6, W8, W13, W14 |
| P7 | 8 | W2, W3, W4, W6, W8, W11, W13, W14 |
| P8 | 8 | W2, W3, W4, W6, W8, W11, W13, W14 |
| P9 | 6 | W2, W3, W4, W6, W8, W11 |
| P10 | 8 | W2, W3, W4, W6, W8, W11, W13, W14 |
| P11 | 7 | W2, W3, W6, W8, W11, W13, W14 |
| P12 | 5 | W2, W3, W4, W6, W8 |
| P13 | 2 | W3, W11 |
| P14 | 4 | W8, W11, W13, W14 |
| P15 | 8 | W2, W3, W4, W6, W8, W11, W13, W14 |
| P16 | 6 | W2, W3, W4, W11, W13, W14 |

**表 5 平均作业响应耗时对比 (实例 1)**

**Table 5** Comparison of average task response time (example 1)

| Sequence | Ours | Q-learning | SA | LP | GA |
|---|---|---|---|---|---|
| S1 | 0.0055 | 0.0069 | 0.509 | 0.636 | 0.464 |
| S2 | 0.0048 | 0.0067 | 0.477 | 0.508 | 0.502 |
| S3 | 0.0017 | 0.0051 | 0.490 | 0.607 | 0.525 |
| S4 | 0.0083 | 0.0135 | 0.341 | 0.657 | 0.478 |
| S5 | 0.0014 | 0.0128 | 0.508 | 0.518 | 0.499 |
| S6 | 0.0055 | 0.0086 | 0.634 | 0.572 | 0.463 |
| S7 | 0.0013 | 0.0146 | 0.438 | 0.696 | 0.454 |
| S8 | 0.0014 | 0.0083 | 0.507 | 0.710 | 0.576 |
| S9 | 0.0084 | 0.0015 | 0.596 | 0.602 | 0.507 |
| S10 | 0.0061 | 0.0092 | 0.426 | 0.648 | 0.474 |

表现优良, 而 GA, LP, SA 和 Q-learning 算法, 在总耗时方面大幅升高, 在总奖励方面却呈下降趋势. 可以看出本文算法求解仍可满足实际场景需要.

为了观察作业规模对 5 种算法求解质量的影响, 新增了一组实验, 在保证实验设置与上述实例二中的设置一致的前提下, 选取了作业执行序列 S1∼S10 作为基本序列, 在其基础上分别扩大 2, 4, 6, 8 倍, 即针对每种序列重复执行 2, 4, 6, 8 遍, 并最终重复执行 10 个回合取平均值. 得到了如图 8 所示的结果.

从图 8 可以看出, 本文算法在作业规模扩大后, 解质量提升效果明显. 其中, 当作业数目超过 480

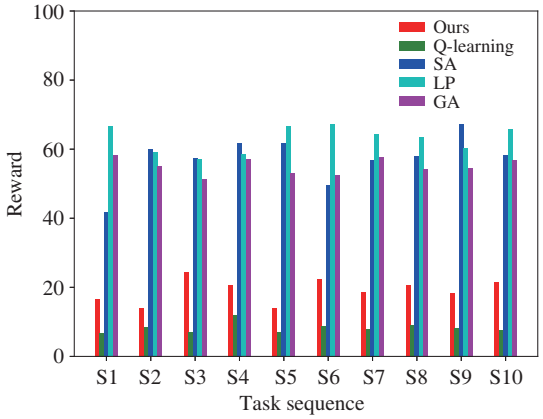


图 6 (网络版彩图) 平均总奖励值对比 (实例 1)

**Figure 6** (Color online) Comparison of average total reward value (example 1)

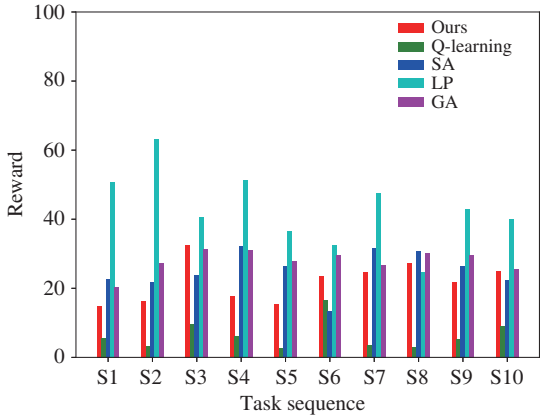


图 7 (网络版彩图) 平均总奖励值对比 (实例 2)

**Figure 7** (Color online) Comparison of average total reward value (example 2)

表 6 平均作业响应耗时对比 (实例 2)

**Table 6** Comparison of average task response time (example 2)

| Sequence | Ours | Q-learning | SA | LP | GA |
|---|---|---|---|---|---|
| S1 | 0.0275 | 0.3319 | 23.792 | 19.02 | 21.22 |
| S2 | 0.0156 | 0.3219 | 23.586 | 20.28 | 18.67 |
| S3 | 0.0152 | 0.3253 | 24.772 | 22.66 | 20.33 |
| S4 | 0.0352 | 0.2292 | 18.764 | 19.60 | 27.09 |
| S5 | 0.0121 | 0.3209 | 21.755 | 19.75 | 25.65 |
| S6 | 0.0234 | 0.3281 | 25.815 | 18.96 | 20.20 |
| S7 | 0.0128 | 0.2287 | 20.083 | 19.19 | 22.33 |
| S8 | 0.0195 | 0.3224 | 20.313 | 24.77 | 21.17 |
| S9 | 0.0115 | 0.2261 | 21.696 | 18.70 | 19.75 |
| S10 | 0.0129 | 0.3283 | 20.797 | 22.46 | 22.38 |

后, 本文算法的解质量将超过除 LP 外的其余几种算法, 逼近线性规划的最优解. 综上, 本文所提算法较其余几种算法更适合大规模保障作业实时调度场景.

## 6 相关工作

### 6.1 传统算法

国内关于舰载机作业调度的研究起步较晚, 主要集中在两方面, 即局部和整体作业流程优化. 其中局部优化主要目标是对一些瓶颈作业, 如弹药装配和起降作业等进行优化资源数量配置. Lin 等 [9] 利用 3 种启发式算法, 即先来先服务、滑动时间窗和最早到达, 针对舰载机的着舰顺序和时刻进行了研究. Lv 等 [10] 针对既定保障任务和升降机批次调度任务的场景, 利用遗传算法进行了舰载机弹药调运次序问题的研究. 在整体作业优化方面, Feng 等 [11] 利用多主体技术研究了不确定条件下舰载机动态调度优化, 使用敏感性方法分析故障扰动对动态调度性能的影响. Han 等 [12] 建立以最小

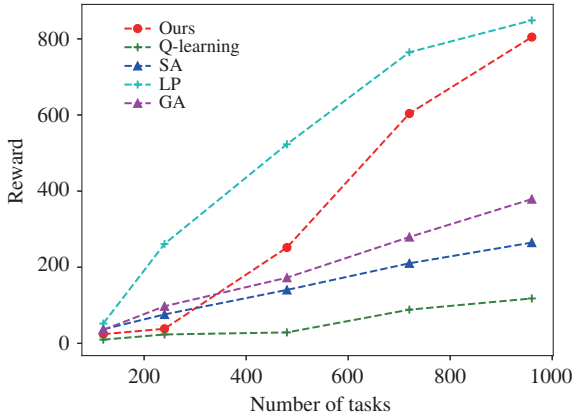


**图 8 (网络版彩图) 大规模作业总奖励对比图**

**Figure 8** (Color online) Comparison of total rewards for large-scale tasks

化闲忙比方差和保障完工时间为双目标的多机一体化机务保障调度模型, 设计了基于自适应多邻域搜索的混合差分进化算法. Fan 等 [13] 综合考虑了战位保障能力和转运路径可行性建立了舰载机甲板调度模型, 利用遗传算法来设计求解算法, 实现了搜索多样性和集中性的平衡. 以上传统算法多用于处理静态场景的规划问题, 规划质量较高, 但耗时巨大. 在航母甲板这个空间狭小、保障时间有着严格限制的场景中, 对作业响应的实时性要求十分高. 显然, 传统算法一般无法达到这个要求.

### 6.2 强化学习算法

近年来, 多智能体系统在资源管理、顺序决策和机器人等领域得到了广泛的研究 [14,15], 其基本思想便是将其建模成一个多智能体问题, 并使用强化学习算法进行求解. Xu 等 [16] 将司机和订单的匹配建模成一个大规模顺序决策问题, 并通过 Markov decision process (MDP) 算法对其进行了求解, 其成果被应用在滴滴出行平台上, 显著提高了平台的订单调度效率. Wang 等 [17] 提出了动态二部图匹配 (DBGM) 问题, 并采用了基于强化学习的算法 restricted Q-learning (RQL) 对该问题进行求解, 最终设计出了一套具有恒定竞争比的自适应批处理解决方案框架, 其针对批处理拆分可以做到近乎最优的决策. Lin 等 [18] 通过对复杂的动态供需环境下的大规模在线乘车共享平台的研究, 利用基于强化学习的 contextual deep Q-learning 和 contextual multi-agent actor-critic 两种算法对大规模车队管理问题进行求解, 实现了大量智能体在不同上下文间的自适应. Li 等 [19] 提出了上下文协作强化学习用于指导每个快递员在每个短时间内应在何处派件和收件, 该方法不仅注重于快递员之间的合作, 还兼顾系统上下文. Shan 等 [20] 提出了用于众包平台中任务安排的深度强化学习框架. Zhang 等 [21] 设计了一种端到端的深度强化学习网络用于在高维连续的数据库管理系统空间内找到最佳配置. 近年来, 一些研究 [22,23] 尝试采用 deep reinforcement learning (DRL) 来解决涉及时空数据的实际问题. 此外, 还有一些研究将 DRL 应用于推荐系统, 例如 Chen 等 [24] 提出了一种健壮的 DQN 方法, 以在动态电子商务平台中获得更好的推荐性能, Hu 等 [25] 采用 DRL 来学习每次搜索的最佳排名策略. 本文是 DQN 在舰载机作业调度领域首次应用, 通过与传统算法 GA, LP 和 SA 以及强化学习算法 Q-learning 进行实验对比后发现, 它较 GA, LP, SA 和 Q-learning 在综合效果 (效率和质量) 方面有较好的表现.

## 7 总结与展望

本文提出了一种新的舰载机保障作业实时调度算法, 旨在优化保障作业调度的长期收益和局部保障需求. 为此, 本文将保障作业调度建模为一个基于 POMDP 的顺序决策问题. 然后, 以集中协调的方式确定多个舰载机和保障战位之间的匹配. 通过实验对比发现, 本文设计算法具有良好的鲁棒性. 与传统的舰载机保障作业调度问题所采用的遗传算法、线性规划、模拟退火算法和 Q-Learing 等方法相比, 本文所提的实时调度算法在效率上和规模上具有良好的性能.

舰载机保障作业调度不仅是影响舰载机出动能力, 更是影响航母综合作战能力的关键因素, 目前关于舰载机保障作业调度问题仍有许多方面值得深入研究. 未来计划将会从人机协同方面对实时调度算法进行优化, 在机器决策的过程中加入人类经验, 提高算法的执行效率和准确性, 进而全面提升舰载机的保障能力.

## 参考文献


1 Zhou X G, Feng B S, Chi Z Y, et al. Analysis of carrier-borne aircraft movement rate based on closed line network. Ordn Ind Autom, 2014, 33: 79–83 [周晓光, 冯百胜, 迟志艳, 等. 基于闭排队网络的舰载机出动架次率分析. 兵工自动化, 2014, 33: 79–83]

2 Ryan J C, Banerjee A G, Cummings M L, et al. Comparing the performance of expert user heuristics and an integer linear program in aircraft carrier deck operations. IEEE Trans Cybern, 2014, 44: 761–773

3 Han W, Si W C, Ding D C, et al. Multi-routes dynamic planning on deck of carrier plane based on clustering PSO. J Beijing Univ Aeronaut Astronautics, 2013, 39: 610–614 [韩维, 司维超, 丁大春, 等. 基于聚类 PSO 算法的舰载机舰面多路径动态规划. 北京航空航天大学学报, 2013, 39: 610–614]

4 Bian D P, Luan T T, Song Y. A layout method of carrier-based aircraft based on simulated annealing. Appl Sci Technol, 2015, 42: 20–24 [卞大鹏, 栾添添, 宋晔. 基于模拟退火算法的舰载机布列方法研究. 应用科技, 2015, 42: 20–24]

5 Li J, Sun Z, Li M L, et al. Research on scheduling decision of carrier aircraft support operation. Ship Electron Eng, 2018, 38: 165–168 [李经, 孙哲, 李梦龙, 等. 舰载机保障作业调度决策研究. 舰船电子工程, 2018, 38: 165–168]

6 Abbeel P. Apprenticeship learning and reinforcement learning with application to robotic control. Dissertation for Ph.D. Degree. Palo Alto: Stanford University, 2008

7 Mnih V, Kavukcuoglu K, Silver D, et al. Playing Atari with deep reinforcement learning. 2013. ArXiv:1312.5602

8 Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning. Nature, 2015, 518: 529–533

9 Lin H, Zhan M F, Zhou F. Optimal scheduling algorithm and simulation of carrier-based aircraft recovery task. J Nav Univ Eng, 2008, 20: 50–54 [林华, 占明锋, 周丰. 舰载机回收任务的优化调度算法及仿真. 海军工程大学学报. 2008, 20: 50–54]

10 Lv X F, Guo X W, Wang Y F. Ammunition scheduling sequence of carrier-based aircraft based on genetic algorithm. Ordn Ind Autom, 2011, 30: 10–12 [吕晓峰, 郭小威, 王云飞. 基于遗传算法的舰载机弹药调度次序. 兵工自动化, 2011, 30: 10–12]

11 Feng Q, Zeng S K, Kang R. Simulation and optimization method for dynamic dispatch of carrier-based aircraft under uncertain conditions. J Syst Simul, 2011, 23: 1497–1501 [冯强, 曾声奎, 康锐. 不确定条件下舰载机动态调度仿真与优化方法. 系统仿真学报, 2011, 23: 1497–1501]

12 Han W, Su X C, Chen J F. Multi-aircraft integrated maintenance scheduling method for carrier-based aircraft. J Syst Eng Electron, 2015, 37: 809–816 [韩维, 苏析超, 陈俊锋. 舰载机多机一体化机务保障调度方法. 系统工程与电子技术, 2015, 37: 809–816]

13 Fan J L, Zhu X D, Gao W, et al. Shipboard aircraft dispatching again based on parallel genetic algorithm. J Ordnance Eq Eng, 2019, 40: 139–143 [范加利, 朱兴动, 高伟, 等. 基于并行遗传算法的舰载机再次出动作业调度. 兵器装备工程学报, 2019, 40: 139–143]

14 Michael J W. An Introduction to Multiagent Systems. 2nd ed. Hoboken: John Wiley & Sons, 2009

15 Bakker B, Whiteson S, Kester L, et al. Traffic light control by multiagent reinforcement learning systems. In: Interactive Collaborative Information Systems. Berlin: Springer, 2010. 475–510
16 Xu Z, Li Z X, Guan Q W, et al. Large-scale order dispatch in on-demand ride-hailing platforms: a learning and planning approach. In: Proceedings of the 24th ACM SIGKDD International Conference, 2018. 905–913
17 Wang Y S, Tong Y X, Long C, et al. Adaptive dynamic bipartite graph matching: a reinforcement learning approach. In: Proceedings of the 35th International Conference on Data Engineering (ICDE), 2019. 1478–1489
18 Lin K X, Zhao R Y, Xu Z, et al. Efficient large-scale fleet management via multi-agent deep reinforcement learning. In: Proceedings of the 24th ACM SIGKDD International Conference, 2018. 1774–1783
19 Li Y X, Zheng Y, Yang Q. Efficient and effective express via contextual cooperative reinforcement learning. In: Proceedings of the 25th ACM SIGKDD International Conference, 2019. 510–519
20 Shan C, Mamoulis N, Cheng R, et al. An end-to-end deep RL framework for task arrangement in crowdsourcing platforms. In: Proceedings of the 36th International Conference on Data Engineering, 2020. 49–60
21 Zhang J, Liu Y, Zhou K, et al. An end-to-end automatic cloud database tuning system using deep reinforcement learning. In: Proceedings of International Conference on Management of Data, 2019. 415–432
22 Li Y X, Zheng Y, Yang Q. Dynamic bike reposition: a spatio-transit reinforcement learning approach. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018. 1724–1733
23 Wei H, Zheng G J, Yao H X. IntelliLight: a reinforcement learning approach for intelligent traffic light control. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018. 2496–2505
24 Chen S Y, Yu Y, Da Q, et al. Stabilizing reinforcement learning in dynamic environment with application to online recommendation. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018. 1187–1196
25 Hu Y J, Da Q, Zeng A X, et al. Reinforcement learning to rank in e-commerce search engine: formalization, analysis, and application. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018. 368–377

# Real-time scheduling for carrier-borne aircraft support operations: a reinforcement learning approach

Yafei LI, Qingshun WU, Mingliang XU*, Pei LV, Xiaoheng JIANG, Ruijie ZHU & Bing ZHOU

*School of Information Engineering, Zhengzhou University, Zhengzhou* 450001*, China*
* Corresponding author. E-mail: iexumingliang@zzu.edu.cn

**Abstract** The carrier-borne aircraft dispatch rate is an important indicator to measure the combat performance of an aircraft carrier. The key factor affecting the carrier-borne aircraft dispatch rate is the efficiency of the carrier aircraft support operation scheduling. Shipboard aircraft support operation scheduling refers to a rational arrangement of the order of support operations required by the carrier aircraft as well as to an efficient completion of the support operations for the carrier-borne aircraft under constraints of limited time, space, and resources. The existing solution strategies based on optimization methods (dynamic programming, linear programming, etc.) and heuristic methods (genetic algorithm, particle swarm optimization, etc.) are only suitable for operation scheduling in the case of predictable operations, and it is challenging to meet the real-time support operation scheduling requirements in highly dynamic combat scenarios. For this reason, a new real-time scheduling method for carrier-borne aircraft support operations based on deep Q-networks (DQNs) is proposed. This method consists of modeling the scheduling problem of aircraft support operations as a partially observable Markov decision process (POMDP) problem. The global and long-term benefits are used to optimize the scheduling process, and the decision-making framework of offline learning and online deployment is used to solve the scheduling problem of aircraft support operations. The simulation results demonstrate that this new method can significantly improve the efficiency of shipboard aircraft support operation scheduling, thus enabling to meet the needs of real-time decision-making environments.

**Keywords** carrier-borne aircraft, support operations, real-time scheduling, reinforcement learning, simulations

**Yafei LI** was born in 1983. He received his Ph.D. degree in computer science from Hong Kong Baptist University in 2015. He is currently an associate professor in the School of Information Engineering, Zhengzhou University, China. His research interests span mobile and spatial data management, location-based services, and urban computing.

**Qingshun WU** was born in 1996. He received his B.S. degree in computer science and technology from Zhengzhou University, China, in 2019. He is currently studying for an M.S. degree at the School of Information Engineering, Zhengzhou University. His research interests include multiagent computing, deep learning, and spatiotemporal data processing

**Mingliang XU** was born in 1981. He received his Ph.D. degree in computer science and technology from the State Key Lab of CAD&CG at Zhejiang University, Hangzhou, China. He is currently a full professor in the School of Information Engineering of Zhengzhou University, China. Furthermore, he is the director of the Center for Interdisciplinary Information Science Research (CIISR) and the vice general secretary of ACM SIGAI China. He has worked at the Department of Information Science of the National Natural Science Foundation of China (NSFC), from Mar. 2015 to Feb. 2016. His current research interests include computer graphics, multimedia, and artificial intelligence.

**Pei LV** was born in 1986. He received his Ph.D. degree from the State Key Laboratory of CAD&CG, Zhejiang University, China, in 2013. He is currently an associate professor in the School of Information Engineering, Zhengzhou University, China. His research interests include video analysis and crowd simulation.