

控制与决策

Control and Decision

基于突发事件的任务计划动态调整模型及算法

孙鹏, 陈冠宇, 张杰勇, 武君胜

引用本文:

孙鹏, 陈冠宇, 张杰勇, 等. 基于突发事件的任务计划动态调整模型及算法[J]. *控制与决策*, 2020, 35(5): 1052–1062.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2018.1096>

您可能感兴趣的其他文章

Articles you may be interested in

带有专家信度的无人机任务分配最小风险问题

minimum-risk problem of unmanned aerial vehicle task allocation with expert belief degree
控制与决策. 2019, 34(9): 2036–2040 <https://doi.org/10.13195/j.kzyjc.2018.0413>

基于分众模式的多部门大群体应急决策方法

Multi-department large group emergency decision making method based on crowd sourcing
控制与决策. 2019, 34(4): 871–879 <https://doi.org/10.13195/j.kzyjc.2017.1226>

灾后动态环境下基于MPC的应急运输实时调度

Real-time relief transportation planning based on MPC in post-disaster dynamic environment
控制与决策. 2018, 33(12): 2131–2141 <https://doi.org/10.13195/j.kzyjc.2017.0928>

基于SEAD任务特性约束的协同任务分配方法

Research on cooperative task assignment method used to the mission SEAD with real constraints
控制与决策. 2017, 32(9): 1574–1582 <https://doi.org/10.13195/j.kzyjc.2016.0858>

带返工的定点装配车间生产计划与调度集成优化

Integrated optimization of production planning and scheduling for fixed-position assembly workshop with rework
控制与决策. 2017, 32(11): 2045–2055 <https://doi.org/10.13195/j.kzyjc.2016.1151>

炼钢连铸生产调度的优先级策略混合遗传算法

HGA combined with priority strategy for production planning of steelmaking-continuous casting
控制与决策. 2016, 31(8): 1394–1400 <https://doi.org/10.13195/j.kzyjc.2015.0856>

基于鲁棒多目标优化的智能体群组协同任务规划

Cooperative task scheduling method for agent group using robust multi-objective optimization approach
控制与决策. 2016, 31(11): 2045–2052 <https://doi.org/10.13195/j.kzyjc.2015.1573>

考虑时序约束的多智能体协同任务分配

Multi-agents cooperative task allocation with precedence constraints
控制与决策. 2015, 30(11): 1999–2003 <https://doi.org/10.13195/j.kzyjc.2014.1114>

基于突发事件的任务计划动态调整模型及算法

孙 鹏^{1,2†}, 陈冠宇¹, 张杰勇¹, 武君胜³

(1. 空军工程大学 信息与导航学院, 西安 710077; 2. 西北工业大学 计算机学院, 西安 710072;
3. 西北工业大学 软件学院, 西安 710072)

摘要: 由于战场环境的复杂多变,在作战过程中会产生一些突发事件,这些事件主要包括新任务出现和平台实体失效.为应对作战中的突发事件,战前制定的任务计划在作战过程中需作出适应性调整,战时任务计划的调整是作战指挥控制领域的难点问题.首先,描述指挥控制组织的组成要素,分析在突发事件下任务计划调整的约束条件,建立以得到最小使命完成时间为目标函数,以任务完成质量、任务插入顺序和调整稳定性为约束的数学模型;其次,提出一种基于可行任务执行序列和贪婪算法(greedy strategy, GS)的任务计划调整方法,给出任务计划调整方法的设计思路和详细步骤;最后,结合联合登陆作战的案例进行仿真,仿真实验验证了所提出方法应对突发事件的可行性和有效性.

关键词: 指挥控制组织; 突发事件; 任务计划; 适应性优化; 调整模型; 贪婪算法

中图分类号: TP391.9

文献标志码: A

Dynamic task plan adjustment model and algorithm based on battlefield emergencies

SUN Peng^{1,2†}, CHEN Guan-yu¹, ZHANG Jie-yong¹, WU Jun-sheng³

(1. College of Information and Navigation, Air Force Engineering University, Xi'an 710077, China; 2. School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China; 3. School of Software, Northwestern Polytechnical University, Xi'an 710072, China)

Abstract: The high uncertainty of battlefield environment leads to some emergencies in the course of a battle. These events mainly include the emergence of new tasks and the failure of platform entities. In order to cope with emergencies in a battle, in the face of changes in the battlefield environment, the task plan formulated before the operation needs to be adjusted in the course of operations, and the adjustment of wartime task plan is a difficult issue in the field of command and control. Firstly, the components of the command and control organization are described, the constraints in the task plan adjustment are analyzed, a mathematical model is established, which takes the minimum mission completion time as the objective function, and task completion quality, task insertion order and adjustment stability as the constraints. Then, a mission planning adjustment algorithm based on the feasible task execution sequence and the greedy algorithm is proposed. The design ideas of adjustment method and detailed steps of the task plan adjustment method are given. Finally, combined with the joint landing operations case simulation, the simulation experiment verifies the feasibility and effectiveness of the method in dealing with emergencies.

Keywords: command and control organization; battlefield emergencies; mission planning; adaptive adjustment optimization; adjustment model; greedy algorithm

0 引言

作战任务由作战使命分解得到,任务完成的好坏直接影响作战使命能否达成^[1],因此,如何合理高效地为任务分配平台资源,保证每个任务的完成质量,已经成为现阶段指挥控制领域内的热点问题^[2-4].

目前,国内外诸多学者对任务计划的制定问

题进行研究,并取得了一定的成果.文献[5]提出了多维动态列表规划(multidimensional dynamic list scheduling, MDLS)与成对交换(pair wise exchange, PWE)算法相结合的方法,以解决任务与资源之间的分配问题;文献[6]在MDLS算法的基础上进行改进,提出了多优先级列表动态规划(multiPRI list dynamic

收稿日期: 2018-08-09; 修回日期: 2018-12-26.

基金项目: 国家自然科学基金项目(61573017, 61773396); 陕西省自然科学基金项目(2017JM6062); 装备预研重点实验室基金项目(61421010201).

†通讯作者. E-mail: Sunypt@163.com.

scheduling, MPLDS)算法;文献[7]提出了基于动态列表调度和遗传算法的任务-平台资源匹配方法. 在任务计划的调整方面,文献[8]提出了以方案改造代价为约束条件的任务计划调整方法.

随着战场环境复杂多变,战场上可能出现各种不确定性因素^[9-12]. 作战任务会根据变化的战场态势进行改变,例如,出现任务增加、任务位置改变、任务处理时间增加、平台损毁等情况^[12]. 突发事件发生后,战前拟定的任务计划无法适用于变化后的作战环境,因此任务和平台发生改变后,及时地调整任务-平台分配方案,修改战前作战计划,以使作战使命顺利达成.

目前,关于战前平台资源调度方案的制定已经取得了许多成果^[13-16],但在任务计划的适应性调整上研究较少. 鉴于此,本文建立了任务计划调整的数学模型,基于可行任务执行序列与贪婪算法进行模型求解,最后结合具体的作战案例进行仿真实验,通过数值仿真验证所提出算法的可行性和有效性.

1 任务计划调整问题建模

为了保持指控组织的稳定性,任务计划的调整不能依赖于重新制定新的任务计划,而是对某些任务执行精度低于阈值的任务进行平台资源的重新分配,未受突发事件影响的任务则保持原有的任务-平台关系,进而保证组织不会因复杂的关系调整造成不稳定.

1.1 基本概念

文中涉及到的基本概念定义如下.

定义1 (任务 T) 使命是战役要达到的目标,任务是将使命分解后得到的军事行动. 任务的完成需要调用一个或多个平台. 在作战使命MP中,每个任务 $T_i (i = 1, 2, \dots, NT)$ 都有以下属性,其中NT为任务的总数:

- 1) 任务 T_i 的开始时间 $t_{s,i}$;
- 2) 任务 T_i 的处理时间 $t_{p,i}$;
- 3) 任务 T_i 的平面坐标 $\text{Loc } T_i = (x_i, y_i)$;
- 4) 任务 T_i 的优先权系数 Pr_i ;
- 5) 任务 T_i 的资源需求矢量 $R_i, R_i = [R_{i1}, \dots, R_{iL}, \dots, R_{iL}]$,其中 R_{iL} 是任务 T_i 需要 L 类型资源的能力数值.

定义2 (资源 R) 资源是兵力组织在作战活动中不可再分割的基本单元,不同类型的资源具有不同的能力. 任务的执行需要一定数量的平台,每个平台具有一种或多种类型资源. 例如火力打击资源、情报侦察资源,这些资源不能单独存在,而是寄托在物理平台上. 描述 C_2 组织基本的资源能力矢量为

$F = [f_1, f_2, \dots, f_L], L$ 为资源的类型数.

定义3 (平台 P) 平台在指控组织中承载着资源,是直接执行和完成作战任务的实体. 在作战使命MP中平台实体 $P_j (j = 1, 2, \dots, NP)$ 具有以下属性,其中NP为平台实体的数目:

- 1) 平台 P_j 的移动速度 v_j ;
- 2) 平台 P_j 的平面坐标 $\text{Loc } P_j = (x_j, y_j)$,平台实体位置是可变的;
- 3) 平台 P_j 的资源能力矢量 $r_j, r_j = [r_{j1}, \dots, r_{jL}, \dots, r_{jL}]$,其中 r_{jl} 是平台 P_j 拥有的 l 类型资源的能力数值.

任务、资源与平台的关系如图1所示.

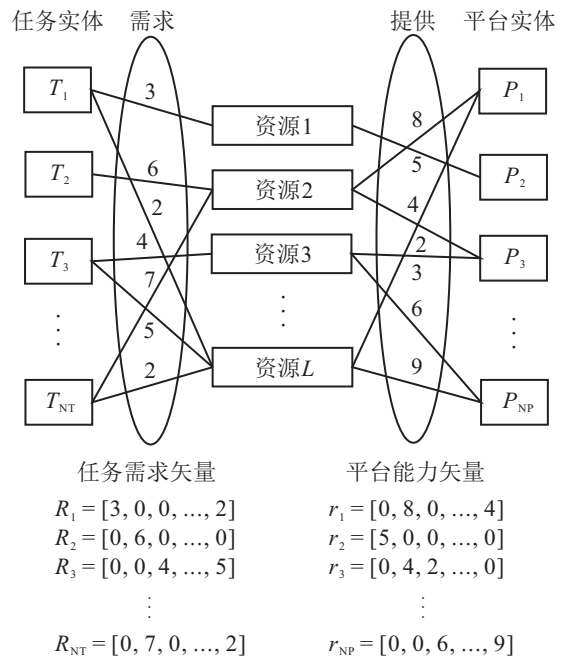


图1 任务、资源与平台关系

在复杂多变的战场环境中,影响作战的突发事件随时会出现. 这些突发事件主要包括任务新增和平台失效,当出现新的作战任务时,需要为新的任务分配平台资源;当出现平台损毁时,需要为受到影响的任务重新分配平台资源.

1.2 相关变量定义

按照战前制定的任务计划执行各个作战任务, t 时刻,部分任务的状态如图2所示.

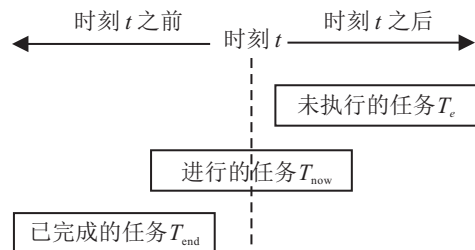


图2 任务分类

任务执行完成时间在 t 时刻前的为已完成任务

T_{end} : 任务开始执行时间在 t 时刻后的为未执行的任
务 T_e ; 时刻 t 介于任务开始时间与结束时间之间的,
为正在进行的任务 T_{now} :

1) t 时刻, 未执行任务 T_e 组成的集合为 $S_T^{\text{wait}}(t) = \{T_{e_1}, T_{e_2}, \dots, T_{e_{N(t)}}\}$, 其中 $N(t)$ 为 t 时刻使命中待
执行的任任务总数. 战场上所有平台集合为 $S_P(t) = \{P_{f_1}, P_{f_2}, \dots, P_{f_{PN(t)}}\}$, 其中 $PN(t)$ 为 t 时刻平台的总
数, 若在使命的执行过程中未发生平台损毁的事件,
则 $PN(t)$ 一直等于 NP .

2) 分配变量 m_{ij} , 若平台 P_j 分配给了任务 T_i , 则
 m_{ij} 为 1, 否则 m_{ij} 为 0.

3) 使命的完成时间 MIT, 即使命中最后一个任务的
结束时间, 有

$$\text{MIT} = \max_{i=1,2,\dots,NT} (t_{s,i} + t_{p,i}). \quad (1)$$

4) 使命执行中遇到突发事件后, 未执行任务组成
的集合为 $S_T^{\text{wait}}(t') = \{T_{g_1}, T_{g_2}, \dots, T_{g_{N(t')}}\}$, 平台的
集合为 $S_P(t') = \{P_{h_1}, P_{h_2}, \dots, P_{h_{M(t')}}\}$.

5) 分配变量 $m(t)_{ij}^{T-P}$, 若 $m(t)_{ij}^{T-P} = 1$, 则平台 P_{f_j}
被分配处理任务 T_{e_i} , 否则 $m(t)_{ij}^{T-P} = 0$. 任务-平台实
体调度方案为 $M(t)^{T-P} = (m(t)_{ij}^{T-P})_{N(t) \times PN(t)}$, 调整
后的分配方案为 $M(t')^{T-P} = (m(t')_{ij}^{T-P})_{N(t') \times PN(t')}$.

6) 初始任务计划调整前的任务执行序列为 TS,
 $\text{TS} = [T_{S_1}, T_{S_2}, \dots, T_{S_{NT}}]$. TS 内任务的顺序即为任
务的执行顺序, TS 在战前准备阶段由任务规划算法
得到, 常用的算法有 MDLS 算法、MPLDS 算法、DLS
算法等^[5].

7) BEFORE(i) 为任务 T_i 的直接前导任务集.

8) BACK(i) 为任务 T_i 的直接后续任务集.

9) 任务 T_i 分配到的平台情况可以用矢量 y_i 表示

$$y_i = [y_{i1}, \dots, y_{il}, \dots, y_{iL}]^T,$$

y_{il} 是 0-1 变量, 表示执行任务 T_i 的平台 P_j 是否具有
任务 T_i 所需的 l 类型资源, 若具有, 则 $y_{il} = 1$, 否则
 $y_{il} = 0$.

1.3 目标函数

调整后的方案有多种, 选择在新的使命环境 MP'
下, 使命完成时间最短的平台-任务分配方案作为调
整后的方案, 将最小化使命完成时间作为调整的目标
函数, 有

$$\min \text{MIT}. \quad (2)$$

1.4 约束条件

1.4.1 任务精度约束

一个任务的完成需要多种类型的资源, 某类资
源需求的匹配情况由资源满足度 Z_{il} 衡量, 资源满足

度是任务中某类资源的实际数量与需求数目的比
值. 结果大于等于 1, 表明该资源的需求完全满足; 结
果小于 1, 则该结果即为满足程度. 因此, 任务 T_i 所需
的 l 类型资源的满足度 Z_{il} 为

$$Z_{il} = \min \left(1, \frac{\sum_{j=1}^J r_{jl} y_{ij}}{R_{il}} \right). \quad (3)$$

任务 T_i 的精度 $\text{Acc}(i)$ 反映了任务内各类资源满
足度的总体情况, 采用求几何平均的方法可得

$$\text{Acc}(i) = \left(\prod_{l \in \gamma(i)} \min \left(1, \frac{\sum_{j=1}^J r_{jl} y_{ij}}{R_{il}} \right) \right)^{\frac{1}{|\gamma(i)|}}. \quad (4)$$

其中: $\gamma(i)$ 为当前处理任务所需资源类型的集合,
 $|\gamma(i)|$ 为集合中类型的数目.

若任务 T_i 中出现某类型资源满足度为 0 的情况,
则表示该任务不能被完成符合实际情况. 每个任务
的完成精度要高于阈值 γ , 由下式可得整个使命的完
成精度 Accuracy:

$$\text{Accuracy} = \sum_{i=1}^{NT} w_i \text{Acc}(i). \quad (5)$$

调整后的任务-平台方案应满足使命完成精度高于阈
值 δ 的要求, 阈值结合具体使命确定.

式(5)中 w_i 代表任务的权重, 反应了任务 T_i 在整
个作战使命内的重要程度, 本文 w_i 取任务总数的倒
数, 即每个任务重要程度相同. 单个任务和使命都有
完成的精度要求, 即

$$\text{Acc}(i) \geq \gamma, \text{Accuracy} \geq \delta. \quad (6)$$

1.4.2 新增任务分配顺序约束

T 时刻战场环境发生变化, 记录下任务计划调
整前序列 TS 中还未执行任务序列, 记为 $\text{TS}_{\text{wait}} =$
 $[T_{S_1}, T_{S_2}, \dots, T_{S_k}, \dots, T_{S_K}]$, $T_{S_k} \in S_T^{\text{wait}}(t)$, K 为序
列中任务的总数, TS_{wait} 内任务的排列顺序参照 TS
中任务的顺序.

根据战场情况添加作战任务, t 时刻新增的任务
为 T_{add} . 将新增的任务 T_{add} 插入 TS_{wait} 后, 得到新的
任务分配序列

$$\text{TS}' = [T_{S_1}, \dots, T_{\text{add}}, \dots, T_{S_{K+1}}].$$

设 T_{add} 插入的位置为 b , 则该位置应当满足 T_{add} 的直
接前续任务都在该位置之前, T_{add} 的后续任务都在
该位置之后, 即

$$\begin{aligned} \text{Before}(T_{\text{add}}) &\subseteq \{T_{S_1}, \dots, T_{S_{b-1}}, T_{\text{add}}\}, \\ \text{Back}(T_{\text{add}}) &\subseteq \{T_{\text{add}}, T_{S_{b+1}}, \dots, T_{S_{K+1}}\}. \end{aligned} \quad (7)$$

1.4.3 调整的稳定性约束

在方案的调整中,要符合最小变更原则,以保持组织结构的稳定.因此,方案调整只为任务精度 $Acc(i) < \gamma$ 的任务重新分配平台,即新增加的任务集合和受平台损毁而导致任务精度低于阈值的任务.

平台损毁影响到的任务为 $S_T^{impact} = \{T_{gi} | P_{h_j} \in S_P^{broken}, m(t)_{ij}^{T-P} = 1\} \cap S_T(t')$, 损毁平台为 $S_P^{broken} = S_P(t) - S_P(t) \cap S_P(t')$. 任务精度 $Acc(i) \geq \gamma$ 的任务集合为 $S_T^{keep}(t') = S_T^{wait}(t') - S_T^{add} - S_T^{impact}$. 为符合最小变更原则,调整的约束为

$$\begin{aligned} m(t')_{ij}^{T-P} &= m(t)_{kl}^{T-P}, \\ (T_{gi} \in S_T^{keep}(t'), T_{gi} &= T_{ek}, \\ P_{h_j} \in S_P(t') \cap S_P(t), P_{h_j} &= P_{fi}). \end{aligned} \quad (8)$$

此约束表示任务精度 $Acc(i) \geq \gamma$ 的任务调整前后调用的平台一致.

1.5 问题模型

结合目标函数和各类约束条件,构建的任务计划调整模型如下所示:

$$\begin{aligned} \min \quad & MIT. \\ \text{s.t.} \quad & Acc(i) \geq \gamma, i = 1, 2, \dots, NT; \\ & Accuracy \geq \delta; \\ & Before(T_{add}) \subseteq \{T_{s_1}, \dots, T_{s_{b-1}}, T_{add}\}; \\ & Back(T_{add}) \subseteq \{T_{add}, T_{s_{b+1}}, \dots, T_{s_{K+1}}\}; \\ & m(t')_{ij}^{T-P} = m(t)_{kl}^{T-P}; \\ & (T_{gi} \in S_T^{keep}(t'), T_{gi} = T_{ek}, \\ & P_{h_j} \in S_P(t') \cap S_P(t), P_{h_j} = P_{fi}). \end{aligned} \quad (9)$$

其中 $m(t)_{ij}^{T-P}$ 为模型的决策变量.

模型(9)的突出特点是,只为新增加的任务或平台损毁后任务执行精度低于阈值的任务重新分配平台,其他任务精度高于阈值的任务,继续按照战前制定的任务计划执行作战任务.传统的方法是对任务计划进行重新制定,本文模型既避免了对任务计划的大幅度调整,也降低了求解问题的复杂度.

2 基于可行任务执行序列和贪婪算法的求解方法

2.1 算法整体流程

按照战前制定的任务计划执行任务, t 时刻突发事件产生,战场态势发生改变,可能会出现任务增加或平台损毁等突发事件.当出现任务新增的事件时,任务执行序列中要插入新增的任务,任务执行序列需要发生改变,平台损毁后因为不涉及任务数目的变

化,可采用原来的任务分配序列.因此对任务计划的调整可根据是否有任务增加而划分为两种情况,同时出现任务增加和平台损毁的情况可以归为任务增加进行处理.算法整体流程如图3所示.

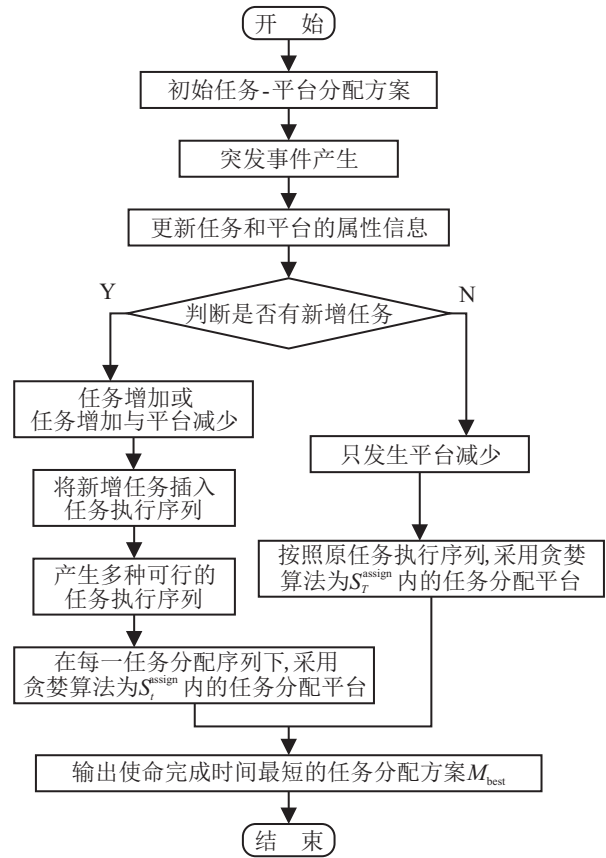


图3 算法整体流程

step 1: t 时刻突发事件产生,判断变化是否有增加的任务,若只发生平台损毁则转至 step 4.

step 2: 若有新增任务,则需将新增任务添加至任务执行序列,若新增任务有多种可行的插入位置,则每个插入位置都能够对应一个任务执行序列.

step 3: 在每种任务执行序列下使用贪婪策略进行求解,求解后转至 step 5.

step 4: 若只发生平台损毁,则结合调整之前的任务执行序列,使用贪婪算法为相关任务重新分配平台.

step 5: 选择执行时间最短的任务分配方案作为调整后的最佳方案.

新增的任务如何插入到任务执行序列,以及贪婪策略的运用是否得当,都会影响最终输出的调整方案是否切实可行,因此下文介绍这两个核心环节的详细流程.

2.2 将新增任务插入任务执行序列

若 t 时刻增加了新的作战任务 T_{add} ,则将新的任务 T_{add} 按照约束公式(7)插入到 TS_{wait} 中,得到

TS'. 每个新增任务符合约束的插入位置可能是1个或多个,每个满足约束条件的插入位置都会对应一个TS',因此对应的TS'也可能不止1个.找出所有TS'并存入分配序列集合Sequence中,Sequence = {TS'_1, TS'_2, ..., TS'_SQ},SQ为集合中任务执行序列的总数.具体插入流程如下.

为了找出所有可行的任务执行序列(如图4所示),采用遍历方法,将T_{add}依次插入TS_{wait}中.

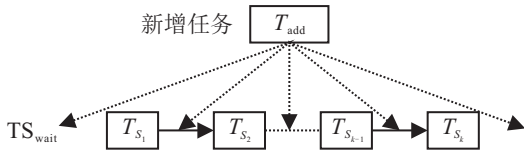


图4 新增任务插入示意图

step 1: 第1个插入位置在T_{S₁}之前,判断T_{add}插入此位置能否满足约束公式(7),即前导任务集合是否为空,若为空,则能够插入在此位置,在Sequence中记录T_{add}插入在T_{S₁}之前所形成的任务执行序列.

step 2: 令k依次取2~K,判断T_{add}插入在T_{S_{k-1}}与T_{S_k}之间时能否满足约束公式(7),若能满足则在Sequence中记录T_{add}插入在T_{S_{k-1}}与T_{S_k}之间位置时得到的TS'.

step 3: 最后的插入位置在T_{S_k}之后,判断T_{add}能否满足约束条件,只需判断T_{add}的后续任务集合是否为空集,若为空集,则能够插入在T_{S_k}之后.

step 4: 在集合Sequence中记录step 1~step 3所得到的任务执行序列,由此可得到所有满足新增任务插入约束的任务执行序列.

2.3 基于贪婪策略的平台资源分配

任务-平台的分配问题,本质上是组合优化寻找最优解的问题,因此可以采用贪婪算法的思想对模型进行求解.

需要重新分配平台的任务集合 $S_T^{assign} = T_{add} \cup S_T^{impact} = \{T_{assign_1}, T_{assign_2}, \dots, T_{assign_{as}}, \dots, T_{assign_{AS}}\}$. 其中:AS为需要重新分配的任务总数,T_{assign_{AS}}分配到的平台集用S_{P^{as}}表示.

若没有出现任务新增,则任务的执行序列为TS_{wait},序列集合Sequence内只有TS_{wait};若出现新增任务,则任务执行序列集合Sequence内有一个或多个TS'.

定义分配方案组成的集合为O, $O = \{M_{Best_1}, M_{Best_2}, \dots, M_{Best_v}\}$,其中V由Sequence中元素的数目确定.

step 1: 令v = 1.

step 2: 从Sequence内选择任务序列TS'_v.

step 3: 给S_{T^{assign}}中的每一任务都分配战场上全部平台资源,即 $\forall T_{assign_{as}} \in S_T^{assign}, S_P^{T_{assign_{as}}} = S_P(t'), S_T^{keep}$ 中的任务遵照约束公式(8)按原任务计划分配平台,以此作为初始分配方案M₀,计算M₀下的使命完成时间MIT₀,令as = 0.

step 4: 若as ≤ AS,则从S_{T^{assign}}中选择任务T_{assign_{as}},为此任务进行平台删减,令as = as + 1,否则转至step 7.

step 5: 当as = 1时,将M₀、MIT₀作为M_{best}、MIT_{best};令M_{temp} = M_{best},MIT_{temp} = MIT_{best},从S_{P^{as}}中选择一个平台进行删减,删除平台后,任务T_{assign_{as}}的精度应满足Acc ≥ γ,删除的平台是本轮删减中可使任务执行时间MIT减少最多的平台,此分配方案记为M_{best},使命完成时间记为MIT_{best},MIT_{best}即为本轮删减中的最优解.

step 6: 若删除S_{P^{as}}中的任意平台都会造成Acc <

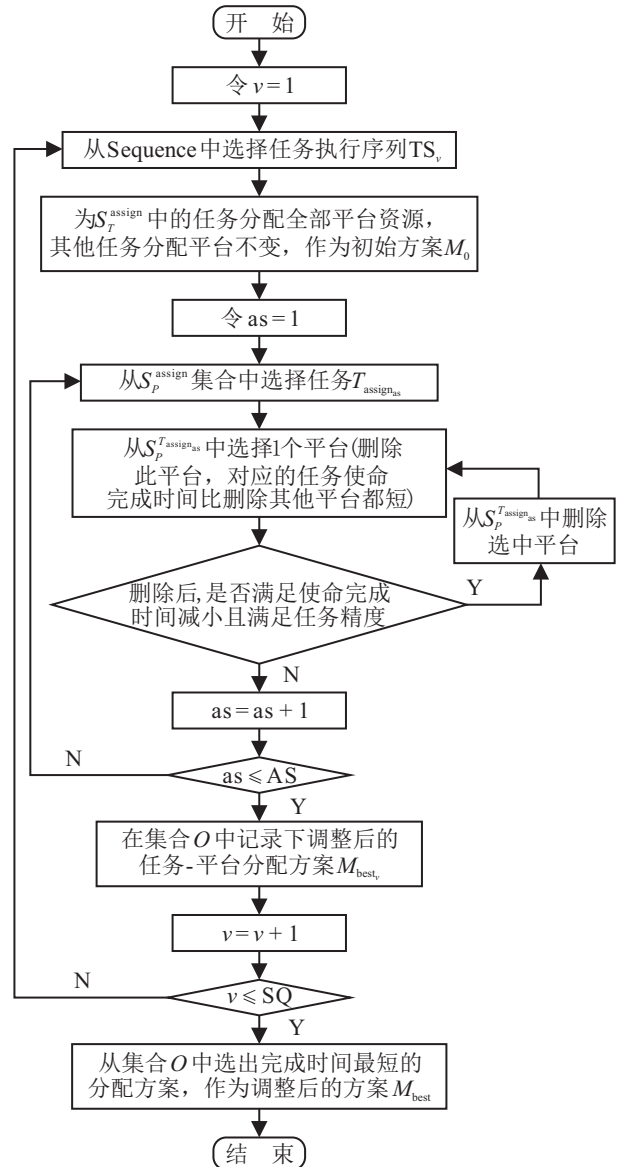


图5 采用贪婪策略求解的具体流程

γ , 或者使命执行时间不满足小于等于上一轮执行时间, 则停止删减, 返回 step 4, 若未满足任务删减的停止条件, 则返回 step 5.

step 7: 在分配方案集合 O 中记录 TS'_v 下的最优方案 M_{best} . 令 $v = v + 1$, 若 $v \geq SQ$, 则从集合 O 中输出使命执行时间最短的 M_{best} 作为调整过后的分配方案, 否则返回 step 2.

结合上文所述的具体步骤, 采用贪婪策略进行求解的流程如图 5 所示.

3 案例分析

算例 1^[17] 我方部队计划进行一次抢滩登陆的作战行动, 行动包括占领敌方的港口和机场, 仿真算例的作战态势如图 6 所示.

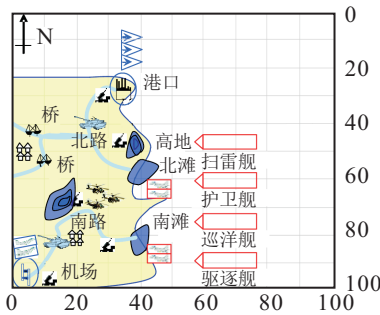


图 6 仿真算例的作战态势图

该作战态势图需要执行的作战任务如下: 北区防御 T_1 、南区防御 T_2 、北区补给 T_3 、南区补给 T_4 、清除海区障碍 T_5 、压制高地 T_6 、抢占北滩 T_7 等 18 个作战任务. 任务之间的序列关系如图 7 所示. 该仿真算例可使用的平台资源包括: 驱逐舰、护卫舰、两栖工兵分队、步兵分队、近距离空中支援作战飞机、舰

载战机、侦察卫星等, 共 $NP = 20$ 个平台资源.

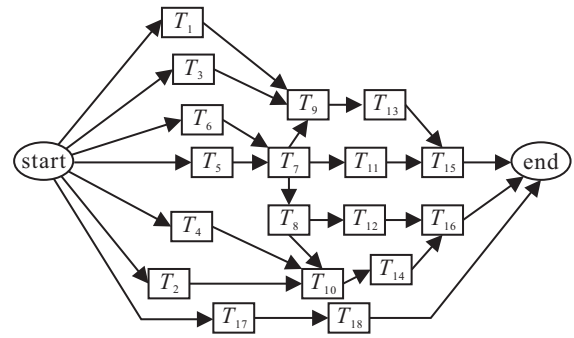


图 7 任务关系

作战任务的属性信息和平台资源的属性信息如表 1 和表 2 所示, 其中任务处理时间的单位为小时. 在表 1 和表 2 中, 任务资源能力需求和平台资源能力共有 8 种, 分别对应如下: R_1 、 r_1 表示对空防御能力; R_2 、 r_2 表示对海防御能力(反舰); R_3 、 r_3 表示反潜能力; R_4 、 r_4 表示炮火压制能力; R_5 、 r_5 表示地面部队进攻能力; R_6 、 r_6 表示装甲兵进攻能力; R_7 、 r_7 表示排雷能力; R_8 、 r_8 表示侦查能力.

考虑到实际情况, 所有平台的初始位置设置为 (82, 41), 分配任务后, 平台由初始位置移动到第 1 个需要执行的任务所在位置.

根据图 7 所示的任务关系以及表 1、表 2 任务和平台的属性信息, 采用 MDLS 算法进行战前任务规划, 采用长度加权算法 (weighted length algorithm, WL) 计算任务优先级, 函数选择 WL 算法^[5], 得到的任务分配方案如图 8 所示, 每个任务的执行精度均为 1, 满足每个任务的资源需求.

表 1 任务属性

任务	资源能力需求向量								处理时间	位置	
	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8		x	y
T_1	5	3	10	0	0	8	0	6	30	70	15
T_2	5	3	10	0	0	8	0	6	30	64	75
T_3	0	3	0	0	0	0	0	0	10	15	40
T_4	0	3	0	0	0	0	0	0	10	30	95
T_5	0	3	0	0	0	0	10	0	10	28	73
T_6	0	0	0	10	14	12	0	0	10	24	60
T_7	0	0	0	10	14	12	0	0	10	28	73
T_8	0	0	0	10	14	12	0	0	10	28	83
T_9	5	0	0	0	0	5	0	0	10	28	73
T_{10}	5	0	0	0	0	5	0	0	10	28	83
T_{11}	0	0	0	0	0	10	5	0	10	25	45
T_{12}	0	0	0	0	0	10	5	0	10	5	95
T_{13}	0	0	0	0	0	8	0	6	20	25	45
T_{14}	0	0	0	0	0	8	0	6	20	5	95
T_{15}	0	0	0	20	10	4	0	0	15	25	45
T_{16}	0	0	0	20	10	4	0	0	15	5	95
T_{17}	0	0	0	0	0	8	0	4	10	5	60
T_{18}	0	0	0	8	6	0	4	10	20	5	60

表2 平台属性

平台	平台初始拥有的资源能力向量								速度
	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	
P_1	10	10	1	0	9	5	0	0	2
P_2	1	4	10	0	4	3	0	0	2
P_3	10	10	1	0	9	2	0	0	2
P_4	0	0	0	2	0	0	5	0	4
P_5	1	0	0	10	2	2	1	0	1.35
P_6	5	0	0	0	0	0	0	0	4
P_7	3	4	0	0	6	10	1	0	4
P_8	1	3	0	0	10	8	1	0	4
P_9	1	3	0	0	10	8	1	0	4
P_{10}	1	3	0	0	10	8	1	0	4
P_{11}	6	1	0	0	1	1	0	0	4.5
P_{12}	6	1	0	0	1	1	0	0	4.5
P_{13}	6	1	0	0	1	1	0	0	4.5
P_{14}	0	0	0	0	0	0	10	0	2
P_{15}	0	0	0	0	0	0	0	6	5
P_{16}	0	0	0	0	0	0	0	6	7
P_{17}	0	0	0	6	6	0	1	10	2.5
P_{18}	1	0	0	10	2	2	1	0	1.35
P_{19}	1	0	0	10	2	2	1	0	1.35
P_{20}	1	0	0	10	2	2	1	0	1.35

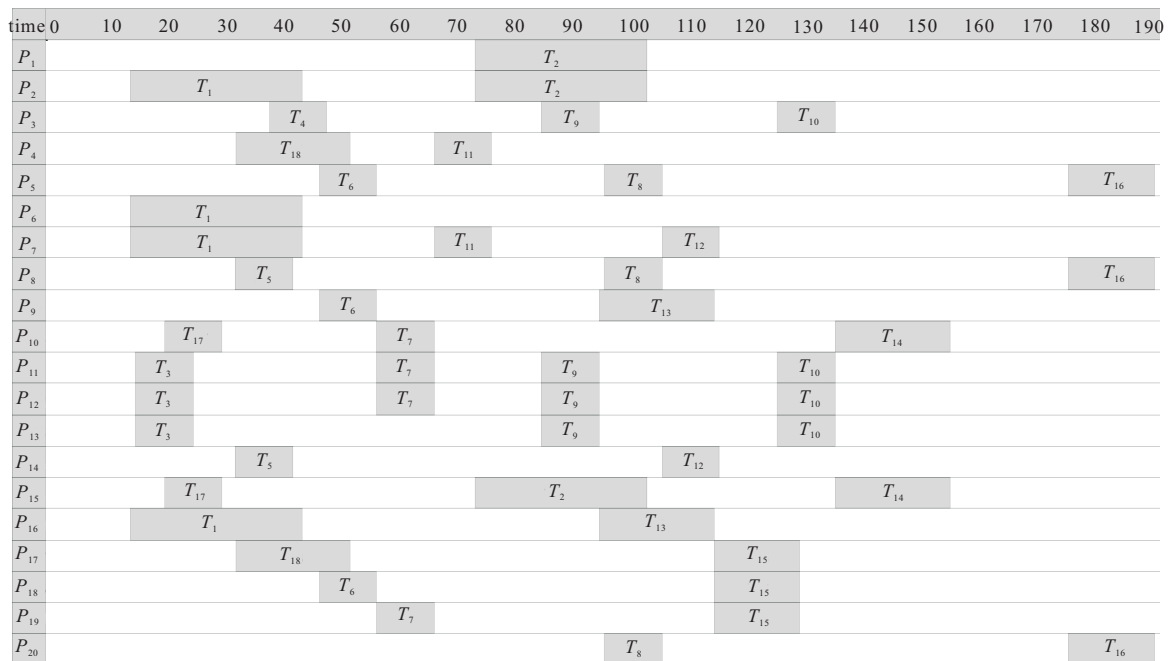


图8 调整前的任务计划甘特图

多维动态列表规划算法 MDLS 是进行任务-平台资源分配的常用方法,该算法是一种快速启发式算法,实现简单^[5].复杂的军事资源调度问题大多采用 MDLS 算法^[15-16],因此采用 MDLS 算法求解初始使命下的任务平台分配方案.在最初的作战条件下,任务执行的序列为 $TS = [T_5, T_6, T_1, T_3, T_4, T_{17}, T_{18}, T_2, T_7, T_8, T_9, T_{11}, T_{13}, T_{10}, T_{12}, T_{15}, T_{14}, T_{16}]$,使命的完成时间为 192.9529 h.

3.1 任务增加事件的仿真分析

3.1.1 任务增加发生在前期

使命执行到 $t = 50$ 时,根据战场态势的变化对作战使命进行调整,需要新增加任务 T_{19} .任务 T_{19} 插入图 8 的 T_5 与 T_7 之间.任务 T_{19} 的位置为 (28,78),资源需求矢量为 $[0, 0, 8, 0, 0, 6, 0, 10]$,任务执行时间为 10 h.

由图 8 可知,当 $t = 50$ 时,任务 $T_1, T_3, T_4, T_5, T_{17}$

已经执行完毕,任务 T_6, T_{18} 正在执行还未结束,执行完毕和正在执行的任务不需要进行任务分配. 当 $t = 50$ 时,待执行任务集合 $S_T^{\text{wait}} = \{T_7, T_2, T_8, T_9, T_{10}, T_{11}, T_{12}, T_{13}, T_{14}, T_{15}, T_{16}\}$, 任务执行顺序 $TS_{\text{wait}} = [T_7, T_8, T_9, T_{11}, T_{13}, T_{10}, T_{12}, T_{15}, T_{14}, T_{16}]$, 新增的任务 T_{19} 根据约束条件添加至 TS_{wait} 后得到 TS' , $TS' = [T_{19}, T_7, T_8, T_9, T_{11}, T_{13}, T_{10}, T_{12}, T_{15}, T_{14}, T_{16}]$.

采用第2节所述的调整方法,求解出增加任务后的最优任务-平台方案,以及任务增加后所有任务的执行情况,阈值 γ 和 δ 均设置为1.

任务增加后的甘特图如图9所示,新增的任务在图中以绿色框显示. 新增任务 T_{19} , 为其分配的平台为 $P_2 P_{17} P_{19} P_{20}$. 任务增加后,使命的完成时间由 192.9529 h 变为 199.5240 h.

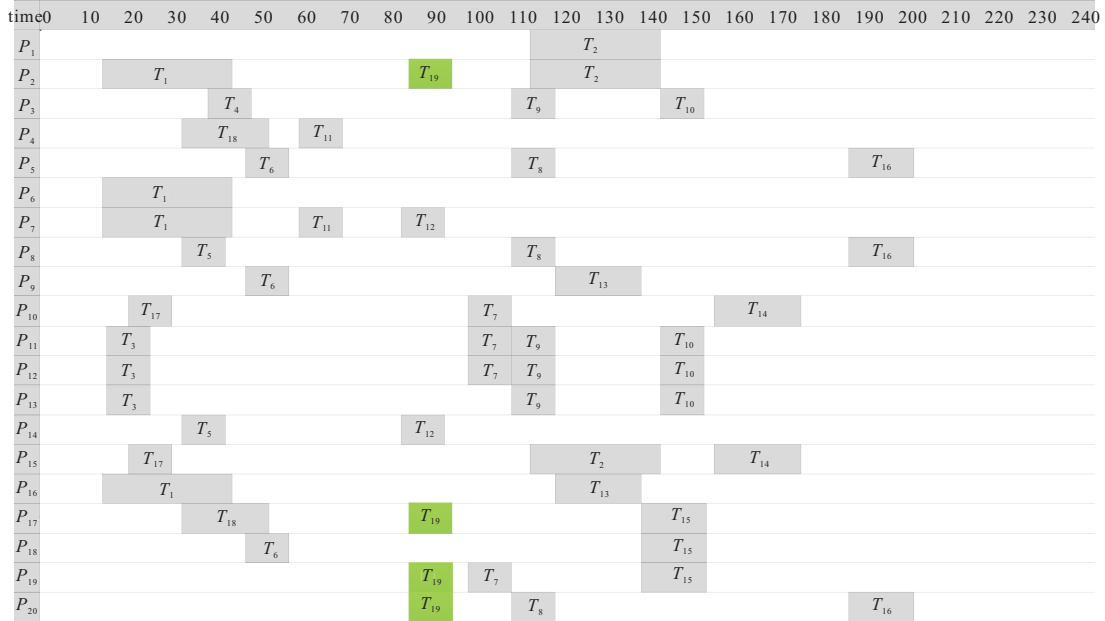


图9 调整后的任务计划甘特图(任务增加发生在前期)

3.1.2 任务增加发生在后期

当 $t = 110$ 时,新增加任务 T_{19} . 任务 T_{19} 插入图8的 T_{14} 与 T_{16} 之间. 任务 T_{19} 的位置为 $(5, 95)$, 资源需求矢量为 $[0, 0, 0, 0, 0, 8, 0, 6]$, 任务处理时间为 20h.

执行待执行任务集合 $S_T^{\text{wait}} = \{T_{10}, T_{14}, T_{15}, T_{16}\}$, 任务执行顺序 $TS_{\text{wait}} = [T_{10}, T_{15}, T_{14}, T_{16}]$, 新增的任务 T_{19} 根据约束条件添加至 TS_{wait} 中, 得到 TS' , $TS' = [T_{10}, T_{15}, T_{14}, T_{19}, T_{16}]$. 任务增加后的甘特图如图10所示,新增的任务在图中以绿色框显示.

由图8可见,当 $t = 110$ 时,任务 T_{12} 、 T_{13} 正在

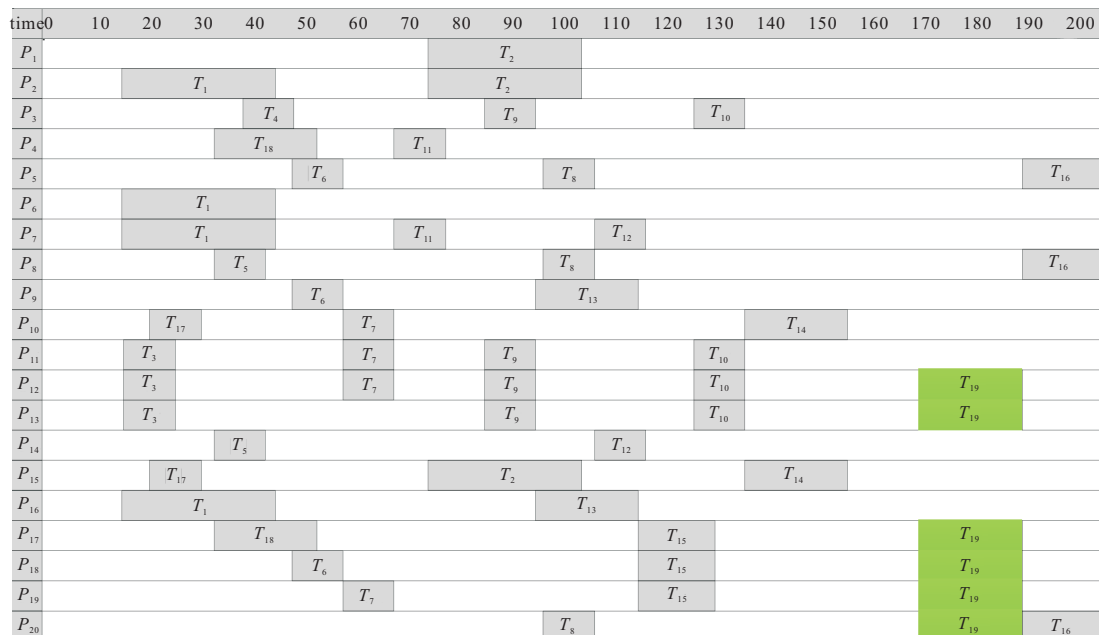


图10 调整后的任务计划甘特图(任务增加发生在后期)

由图10可见, T_{19} 分配到的平台为 $P_{12}P_{13}P_{17}P_{18}P_{19}P_{20}$, 分配的平台资源满足任务 T_{19} 的需求, 其余任务分配到的平台与原分配方案相同.

任务增加后, 使命的完成时间由 192.9529 h 增加至 206.1240 h, 执行时间多出约 14 h, 新增任务的执行时间为 20 h, 该调整算法较好地解决了任务增加的问题.

3.2 平台损毁事件的仿真分析

使命执行到 $t = 60$ 时发生平台损毁, 受损的平台为 P_7 . 由图8可见, 原分配方案中平台 P_7 负责的任务 T_{11} 和 T_{12} 精度低于阈值, 需对原任务分配方案进行调整, 任务 T_{11} 和 T_{12} 重新分配平台, 阈值 γ 和 δ 均设置为 1.

平台 P_7 损毁前任务 T_{11} 和 T_{12} 的平台分配情况

如表3所示. 平台 P_7 损毁后, 进行任务计划调整, 为任务 T_{11} 和 T_{12} 重新分配平台, 分配平台如表4所示.

表3 平台损毁前

任务	分配平台
T_{11}	$P_7 P_4$
T_{12}	$P_{14} P_7$

表4 平台损毁后

任务	分配平台
T_{11}	$P_1 P_2 P_3 P_4$
T_{12}	$P_1 P_2 P_3 P_4$

调整中, 只为任务 T_{11} 和 T_{12} 重新分配平台资源, 其他任务分配的平台不变, 在满足平台相关约束的条件下, 调整后的甘特图如图11所示.

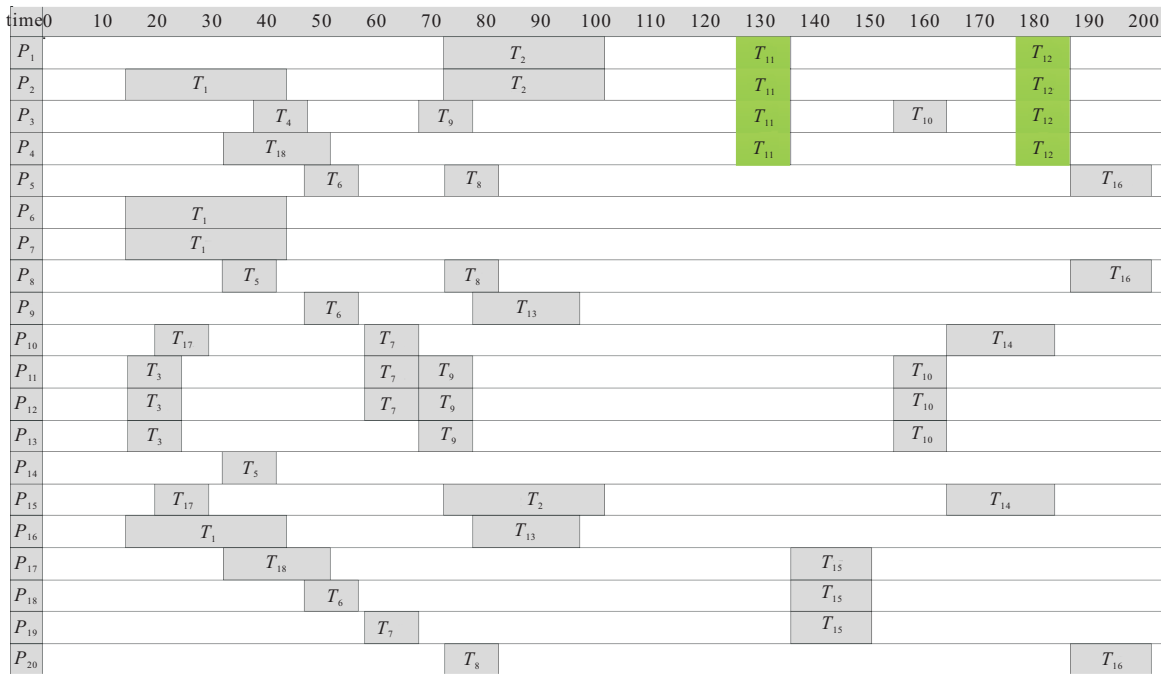


图11 调整后的任务计划甘特图(平台损毁)

调整后使命的结束时间变为 206.2094 h, 相比初始的使命完成时间, 平台 P_7 损毁后使命完成时间向后推移了 14, 满足要求, 解决了平台损耗后的任务计划调整问题.

3.3 算法对比分析

为验证本文所采用的贪婪算法 (greedy strategy, GS) 在解决任务计划调整问题上的优越性, 将贪婪算法与双重贪婪算法 (dual greedy strategy, DGS) 和一致贪婪算法 (consistent greedy strategy, CGS) 进行对比. 3 种方法的差异体现在对冗余平台的删减方法中. 删减前, 为每个需要重新分配平台的任务分配组织内的所有平台, 如图12所示.

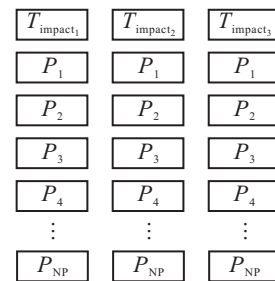


图12 要调整的任务分配所有平台资源

GS 算法在每一步选择要删除的平台时, 判断 $T_{\text{impact}_1} \sim T_{\text{impact}_3}$ 所分配的平台中, 删除哪个平台可使作战使命的完成时间缩短最多. DGS 算法由 $T_{\text{impact}_1} \sim T_{\text{impact}_3}$ 依次进行平台删减操作, 当从

T_{impact_1} 平台集内任意删减一个平台后, T_{impact_1} 的任务需求便不能得到满足, 开始对 T_{impact_2} 的平台集进行任务删减. CGS 算法考虑受影响任务 T_{impact_1} 、 T_{impact_2} 和 T_{impact_3} 的优先级, 先执行优先级高的任务, 进行冗余平台删减时, 依次从任务优先级高的任务开始进行删减.

遗传算法 (genetic algorithm, GA) 和模拟退火算法 (simulated annealing, SA) 也被用于求解任务计划的规划问题^[7, 14, 18-19], 因此也将本文算法与遗传算法和模拟退火算法进行对比, 设置任务的完成质量不低于 1, 进行 10 组仿真实验, 突发事件随机产生, 实验结果如图 13 和图 14 所示. 图中: 菱形为 GS 算法, 六角星为 DGS 算法, 圆形为 CGS 算法, 五角星为 GA 算法, 方形为 SA 算法.

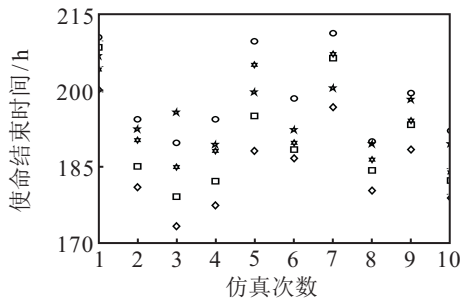


图 13 使命完成时间比较

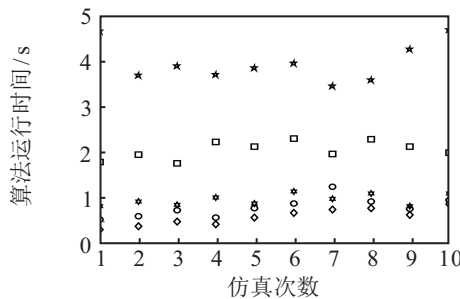


图 14 算法耗时比较

由图 13 可见, GS 算法、DGS 算法、CGS 算法、GA 算法和 SA 算法中, 采用 GS 算法进行任务计划调整得到的使命完成时间最低, 进行调整的效果最好, CGS 算法因为严格按照任务的优先级执行任务, 降低了平台资源的使用效率, 导致整体的使命完成时间较长, 该算法适用于紧急任务需要优先保障情况下的计划调整.

由图 14 可见, 在算法耗时方面, GA 算法耗时较高, SA 算法的用时也高于 3 种贪婪算法, 3 种贪婪算法的耗时相差不多, 算法运行时间较低. 采用 GS 算法在解决本文问题的耗时上明显优于智能算法, 在解决实时性要求较高的任务计划调整问题上更具有优越性.

4 结 论

本文研究了任务新增和平台损毁这两类突发事件下的任务计划调整问题, 建立了以最小化使命完成时间为目标的任务计划调整模型, 结合可行任务执行序列和贪婪算法提出了模型求解的方法. 最后以登陆作战案例进行实验仿真, 初始任务计划由 MDLS 算法对使命求解产生, 分别在两类突发事件下对任务计划进行适应性调整, 并将本文算法与双重贪婪算法、一致贪婪算法、遗传算法和模拟退火算法进行对比, 实验结果表明了所提出方法用于任务计划调整的有效性和优越性.

参考文献 (References)

- [1] Han X, Bui H, Mandal S. Optimization-based decision support software for a team-in-the-loop experiment: Asset package selection and planning[J]. IEEE Transactions on Systems, Man, and Cybernetics, 2013, 43(2): 237-251.
- [2] Bui H, Han X, Mandal S, et al. Optimization-based decision support algorithms for a team-in-the-loop planning experiment[C]. Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics. San Antonio: IEEE, 2009: 4684-4689.
- [3] Yu F, Tu F, Pattipati K R. Integration of a holonic organizational control architecture and multi-objective evolutionary algorithm for flexible distributed scheduling[J]. IEEE Transaction on System, Man, and Cybernetics, 2008, 38(5): 1001-1017.
- [4] Han X, Mandai S, Pattipati K R, et al. An optimization-based distributed planning algorithm: A blackboard-based collaborative framework[J]. IEEE Transactions on Systems, Man, and Cybernetics, 2014, 44(6): 673-686.
- [5] Levchuk G M, Levchuk Y N, Luo J, et al. Normative design of organizations, part I: Mission planning[J]. IEEE Transactions on Systems, Man, and Cybernetics, 2002, 32(3): 346-359.
- [6] 阳东升, 张维明, 刘忠, 等. 战役计划的数学描述与求解算法研究[J]. 系统工程理论与实践, 2006, 26(1): 26-34.
(Yang D S, Zhang W M, Liu Z, et al. Research on mathematical description and solving algorithms of tasks scheduling for campaign[J]. Systems Engineering—Theory & Practice, 2006, 26(1): 26-34.)
- [7] 张杰勇, 姚佩阳, 周翔翔, 等. 基于 DLS 和 GA 的作战任务——平台资源匹配方法[J]. 系统工程与电子技术, 2012, 34(5): 947-954.
(Zhang J Y, Yao P Y, Zhou X X, et al. Approach to

- operation task and platform resource matching based on DLS and GA[J]. *Systems Engineering and Electronics*, 2012, 34(5): 947-954.)
- [8] 孙鹏, 李锴, 姚佩阳, 等. 任务计划适应性改造优化建模及方法[J]. *空军工程大学学报: 自然科学版*, 2016, 17(1): 90-95.
(Sun P, Li K, Yao P Y, et al. Modeling and method of adaptive reform and optimization for mission planning[J]. *Journal of Air Force Engineering University: Natural Science Edition*, 2016, 17(1): 90-95.)
- [9] 张迎新, 陈超, 刘忠, 等. 资源不确定军事任务计划预测调度模型与算法[J]. *国防科技大学学报*, 2013, 35(3): 30-35.
(Zhang Y X, Chen C, Liu Z, et al. Method for modeling and solving military mission planning with uncertain resource availability[J]. *Journal of National University of Defense Technology*, 2015, 35(3): 30-35.)
- [10] 刘海啸. 敏捷C2组织结构设计与调整方法研究[D]. 长沙: 国防科学技术大学研究生院, 2010.
(Liu H X. Design and adjust methodology of command and control organization[D]. Changsha: Graduate School, National University of Defense Technology, 2010.)
- [11] 修保新. C2组织结构方法及其鲁棒性、适应性分析[D]. 长沙: 国防科学技术大学研究生院, 2006.
(Xiu B X. Design methodology of C2 organizational structure and its analysis of robustness and adaptivity[D]. Changsha: Graduate School, National University of Defense Technology, 2006.)
- [12] 吴瑞杰, 孙鹏, 孙昱. 分布式任务计划动态调整模型及算法[J]. *系统工程与电子技术*, 2017, 39(2): 322-328.
(Wu R J, Sun P, Sun Y. Distributed dynamic task plan adjustment model and algorithm [J]. *Systems Engineering and Electronics*, 2017, 39(2): 322-328.)
- [13] 万路军, 姚佩阳, 周翔翔, 等. 多编组协同任务分配模型及DLS-QGA算法求解[J]. *控制与决策*, 2014, 29(9): 1562-1568.
(Wang L J, Yao P Y, Zhou X X, et al. Cooperative task allocation methods in multiple groups using DLS-QGA[J]. *Control and Decision*, 2014, 29(9): 1562-1568.)
- [14] 陈志旺, 陈林, 白铎, 等. 求解约束多目标区间优化的交互多属性决策NSGA-II算法[J]. *控制与决策*, 2015, 30(5): 865-870.
(Chen Z W, Chen L, Bai X, et al. Interactive multi-attribute decision-making NSGA-II for constrained multiobjective optimization with interval numbers [J]. *Control and Decision*, 2015, 30(5): 865-870.)
- [15] 张杰勇, 姚佩阳, 李凡. 完成时间限制下的任务——平台关系设计模型及算法[J]. *系统工程与电子技术*, 2012, 34(8): 1621-1629.
(Zhang J Y, Yao P Y, Li F. Task-platform relation design model and its algorithm under completion time constraint[J]. *Systems Engineering and Electronics*, 2012, 34(8): 1621-1629.)
- [16] 陈行军, 齐欢, 阳东升. 含时间窗联合作战计划问题的建模与求解[J]. *系统工程理论与实践*, 2012, 32(9): 1980-1984.
(Chen X J, Qi H, Yang D S. Modeling and solution of joint operational plans with time window[J]. *Systems Engineering — Theory & Practice*, 2012, 32(9): 1980-1984.)
- [17] Yu F, Tu F, Pappipati K R. Integration of a holonic organizational control architecture and multiobjective evolutionary algorithm for flexible distributed scheduling[J]. *IEEE Transactions on Systems, Man, and Cybernetics*, 2008, 38(5): 1001-1017.
- [18] 包卫东, 王江峰, 张茂军. 一种改进的基于MDLS与GA的作战资源分配算法[J]. *火力与指挥控制*, 2008, 33(9): 18-21.
(Bao W D, Wang J F, Zhang M J. A novel algorithm of task resource distribution based on MDLS and GA[J]. *Fire Control and Command Control*, 2008, 33(9): 18-21.)
- [19] Mu L, Feng Y H, Zhang W M, et al. The adaptive optimization of C2 organization decision layer structure based on nested improved simulated annealing algorithm[C]. *Proceeding of the IEEE International Conference on Intelligent Computing and Intelligent Systems*. Xiamen: IEEE, 2010: 682-687.

作者简介

孙鹏(1972—), 男, 教授, 博士, 从事指挥信息系统、指挥控制组织建模、体系建模等研究; E-mail: Sunypt@163.com;

陈冠宇(1994—), 男, 硕士生, 从事任务规划、敏捷指挥控制组织的研究; E-mail: 821932689@qq.com;

张杰勇(1983—), 男, 讲师, 博士, 从事指挥控制组织建模、体系建模等研究; E-mail: dumu3110728@126.com;

武君胜(1962—), 男, 教授, 博士, 从事指挥信息系统、软件工程等研究; E-mail: wujunsheng@nwpu.edu.cn.

(责任编辑: 郑晓蕾)