一种基于DLS和ACO的平台资源规划方法

周翔翔 姚佩阳 张杰勇 王 欣
(空军工程大学电讯工程学院 西安 710077)

摘 要 平台资源规划方法是作战任务规划的重要组成部分,为作战提供资源分配方案。描述了作战任务、平台以及它们之间的关系,建立了以最小化全部任务完成的截止时间和最大化平台资源的利用率为目标的数学模型。设计了用于求解此模型的动态列表规划(Dynamic List Scheduling,DLS)与蚁群算法(Ant Colony Optimization Algorithm,ACO)相结合的算法,其描述了任务选择方法、ACO的二进制编码方案及候选解构造策略,设计了不可行候选解的修正策略和信息素更新方法,构造了包含任务选择平台的时间优先系数、平台功能能力优先系数和后续任务对平台需求程度3个因素的适应度函数。针对作战想定进行了仿真计算,结果表明,基于DLS和ACO的平台资源规划具有良好的规划效果,相比于他人算法,其具有更少的全部任务完成截止时间和更高的平台资源利用率。

关键词 平台资源规划,动态列表规划,蚁群算法,任务优先权系数,修正策略
中图法分类号 TP391,E919 **文献标识码** A

Platform Resource Scheduling Method Based on DLS and ACO

ZHOU Xiang-xiang YAO Pei-yang ZHANG Jie-yong WANG Xin
(The Telecommunication Engineering Institute, Air Force Engineering University, Xi'an 710077, China)

Abstract Platform resource scheduling method is an important part of operational mission planning and provides operational resource allocation scheme for campaign. Operational task, platform and the relationship between them were described. A mathematics model was set up for platform resource scheduling. The objectives are the mission's finish time minimization and the platform resource's utilization rate maximization. The algorithm, which is composed of dynamic list scheduling(DLS) and ant colony optimization algorithm(ACO) to solve this model was designed. The task selection method, the binary coding scheme, and the candidate solution formation strategy were described. The repaire strategy for infeasible candidate solutions and pheromone updating method were designed. The fitness function was designed with three factors: the time priority coefficient, the platform function capabilitary priority coefficient, and the requirement degree to follow-up tasks. Simulation results based on operational scenario indicate the platform resource scheduling method based on DLS and ACO behaves well. Compared with other algorithms, the proposed algorithm has less mission's finish time and higher platform resource's utilization rate.

Keywords Platform resource scheduling, Dynamic list scheduling(DLS), Ant colony optimization algorithm(ACO), Task priority coefficient, Repair strategy

1 引言

灵活运用作战资源是现代战争取得胜利的重要因素。应当建立合理模型和设计优化方法,使得作战资源能够得到最大效率的使用。平台资源规划是实现作战任务规划的重要内容之一,一般是在作战任务确定之后,根据所有获取到的情报信息,结合己方可用平台资源,再为作战任务分配平台资源[1]。

平台资源规划问题是NP完全问题。解决该类问题的有效途径是启发式方法,而最为常用的启发式方法是动态列表规划(Dynamic List Scheduling,DLS)方法[2]。文献[3]提出了多维动态列表规划(Multidimensional Dynamic List Scheduling,MDLS)算法;文献[4]提出了多优先级列表动态规划(Multipriority List Dynamic Scheduling,MPLDS)算法;文献[5]对MPLDS算法进行了改进。MDLS算法存在局部搜索和优先权函数的合理性问题;MDLS算法和MPLDS算法采用的局部搜索策略都是基于贪心策略的,这无法保证为任务分配的平台结果是最优解,甚至无法保证得到的是次优解。

针对上述不足,本文建立了平台资源规划问题的数学模型,研究了基于DLS和ACO的求解算法,使用DLS作为平

到稿日期:2011-07-09 返修日期:2011-09-03 本文受国家高技术研究发展计划(2010AAJ146),空军工程大学研究生科技创新专项计划(Dx2010307)资助。

周翔翔(1982—),男,博士生,主要研究方向为指控组织设计、任务规划,E-mail:zhouxiangxiang1982@163.com;姚佩阳(1960—),男,教授,博士生导师,主要研究方向为指挥控制系统、数据链;张杰勇(1983—),男,博士生,主要研究方向为指控组织设计;王 欣(1983—),女,博士生,主要研究方向为指挥信息系统效能评估。

台资源规划的基本求解框架，改进 ACO 为选定的任务优化选择平台资源。

2 平台资源规划问题的描述

2.1 基本概念及分析

(1)功能(function,F)：功能是作战过程中不可再分割的基本资源类型，是对作战平台资源类型的划分。能力是功能大小的定量描述，通常平台具有一项或多项功能，任务的执行需要一项或多项功能。记资源类型划分得到的功能集合为 $F=\{f_1,f_2,\cdots,f_L\}$，其中 L 为功能的数量。

(2)作战任务(operational task,T)：作战任务是需要特定的作战功能才能完成的活动，由使命分解而得。记使命分解得到的任务集为 $T=\{T_1,T_2,\cdots,T_I\}$，I 为任务的数量。$\forall T_i\in T$ 均具有下列基本属性：任务的处理时间 $t_{Process}^{(i)}$、任务发生的地理位置 $(x_T^{(i)},y_T^{(i)})$ 和任务的功能能力需求向量 $\boldsymbol{R}_T^{(i)}=[r_T^{(i,1)},r_T^{(i,2)},\cdots,r_T^{(i,L)}]$。其中，$r_T^{(i,l)}$ 表示任务 T_i 需要第 $l(l=1,2,\cdots,L)$ 类功能的大小。一个任务执行的开始需要某些任务均已完成，这便构成了任务集合中任务间的顺序逻辑关系[6]。任务 T_i 的开始执行必须在其所有前导任务执行完成之后。在任务图中通常设置虚拟的任务为 T_0，将之作为所有任务的起点。

(3)平台(platform,P)：平台是处理作战任务所需的物理资源实体，不仅包括火力单元和探测单元，如坦克、飞机、舰艇、雷达、卫星等，还包括成建制的作战部队，如步兵营、导弹营、工兵营等[5]。平台构成的集合为 $P=\{P_1,P_2,\cdots,P_J\}$，J 为平台的数量。$\forall P_j\in P$ 具有下列属性：平台类型 $P_{Type}^{(j)}$、平台的最大速度 v_j、平台的初始地理位置 $(x_P^{(j)}(0),y_P^{(j)}(0))$ 和平台的功能能力向量 $\boldsymbol{R}_P^{(j)}=[r_P^{(j,1)},r_P^{(j,2)},\cdots,r_P^{(j,L)}]$。其中，$r_P^{(j,l)}$ 表示平台 P_j 在第 l 类功能上具备的能力。

2.2 变量定义

本文进行平台资源规划建模与求解过程中主要用到以下变量：

(1)平台-任务分配变量 $m_{TP}^{(i,j)}$：若平台 P_j 分配给任务 T_i，则 $m_{TP}^{(i,j)}=1$；否则，$m_{TP}^{(i,j)}=0$。

(2)任务顺序变量 $o_{i_1i_2}$：若任务 T_{i_1} 是任务 T_{i_2} 的前导任务，即 T_{i_2} 在 T_{i_1} 完成之后才有可能开始，则 $o_{i_1i_2}=1$；否则，$o_{i_1i_2}=0$。

(3)平台的地理位置变量 $(x_P^{(j)},y_P^{(j)})$：该变量表示在战场各作战任务区域进行迁移并执行任务的平台在整个任务过程中不断变化的地理位置。

(4)平台在任务间的转移变量 $d_{i_1i_2}^{(j)}$：若平台 P_j 处理任务 T_{i_1} 后分配给任务 T_{i_2}，则 $d_{i_1i_2}^{(j)}=1$；否则，$d_{i_1i_2}^{(j)}=0$，并设置 $d_{ii}^{(j)}=0$。

(5)任务的开始时间变量 $t_{Start}^{(i)}$：该变量表示任务 T_i 在规划结果中的实际开始执行时刻。

2.3 数学模型的分析与建立

2.3.1 约束分析

(1)任务间顺序关系约束

每个任务的处理必须在所有前导任务都处理完成之后。若任务顺序变量 $o_{i_1i_2}=1$，则任务 T_{i_1} 是任务 T_{i_2} 的前导任务，任务 T_{i_2} 的处理开始必须在任务 T_{i_1} 完成之后，即

$$\begin{cases} t_{Start}^{(i_2)}\geqslant t_{Start}^{(i_1)}+t_{Process}^{(i_1)} \\ o_{i_1i_2}=1 \end{cases} \tag{1}$$

式中，$i_1,i_2=1,2,\cdots,I$。

(2)处理同一任务的平台相互等待约束

由于任务的处理需要分配的所有平台都到达任务发生区域后才开始，其中先到达的平台需要等待，因此，任务的处理开始时间不小于分配的所有平台的到达时间，即

$$t_{Start}^{(i)}\geqslant \max_{j\in\{k\mid m_{TP}^{(i,k)}=1,l(k)\neq 0,k=1,2,\cdots,J\}}\left(t_{Start}^{(l(j))}+t_{Process}^{(j)}+d_{i,l(j)}^{(j)}\times\frac{D_{l(j),i}}{v_j}\right) \tag{2}$$

式中，$i=1,2,\cdots,I;j=1,2,\cdots,J$；$l_j$ 表示平台 P_i 之前处理任务中的最后一个任务的序号，若 T_i 为平台 P_i 处理的第一个任务，则 $l_j=0$；$D_{l(j),i}$ 为任务 T_{l_j} 和任务 T_i 之间的距离，表示为 $D_{l(j),i}=\sqrt{(x_T^{(l(j))}-x_T^{(i)})^2+(y_T^{(l(j))}-y_T^{(i)})^2}$。

(3)任务的平台功能能力约束

在所有的功能类型上，分配到任务的平台的功能能力不小于任务的功能能力需求，即

$$\sum_{j=1}^{J}r_P^{(j,l)}m_{TP}^{(i,j)}\geqslant r_T^{(i,l)} \tag{3}$$

式中，$i=1,2,\cdots,I;l=1,2,\cdots,L$。

(4)平台的独占性约束

平台 P_j 在任意时刻只能处理一个任务，即 P_j 被分配处理任务 T_{i_1} 后，要么使之空闲不分配给其它任务，要么只能分配给一个任务，即

$$\sum_{i_2=0}^{I}d_{i_1i_2}^{(j)}\leqslant 1 \tag{4}$$

式中，$i_1,i_2=1,2,\cdots,I;j=1,2,\cdots,J$。

除上述 4 个约束条件外，平台-任务分配变量 $m_{TP}^{(i,j)}$ 与平台在任务间的转移变量 $d_{i_1i_2}^{(j)}$ 存在如下等价关系，即

$$\sum_{i_1=1}^{I}d_{i_1i_2}^{(j)}=m_{TP}^{(i_2,j)} \tag{5}$$

式中，$i_1,i_2=1,2,\cdots,I;j=1,2,\cdots,J$。

2.3.2 目标选取

平台资源规划问题关注多个设计指标，这些优化指标包括整个任务总的完成时间、平台的平均运动距离和平台资源利用率。平台的平均运动距离较小，则处理同一任务的平台间相互等待的耗时就小，因此该指标已包含在整个任务总的完成时间指标中。本文认为平台资源规划需要在满足作战任务的功能能力需求下，以较短的任务完成时间、较高的平台资源利用率实现平台到任务的最佳分配。

指标 1　全部任务完成的截止时间 $t_{makespan}$

全部任务完成的截止时间是所有任务中最晚完成任务的完成时间，即

$$t_{makespan}=\max_{i=1}^{I}(t_{Start}^{(i)}+t_{Process}^{(i)}) \tag{6}$$

指标 2　平台资源利用率 U

$\forall P_j$，其资源利用率定义为

$$U_j=\frac{1}{\hat{L}_j}\sum_{l=1}^{L}\bar{r}_P^{(j,l)} \tag{7}$$

式中，$\hat{L}_j$ 是平台 P_j 中功能能力值不为零的属性数目，即 $\hat{L}_j=\sum_{l=1}^{L}\delta_P^{(j,l)}$。若 $r_P^{(j,l)}>0$，则 $\delta_P^{(j,l)}=1$；否则 $\delta_P^{(j,l)}=0$。$\bar{r}_P^{(j,l)}$ 为平台 P_j 的第 l 类功能的平均利用率。$\bar{r}_P^{(j,l)}$ 的取值需要分两种情况：

① 若 $r_P^{(j,l)}=0$，则平台 P_j 不具备第 l 类功能能力，故 $\bar{r}_P^{(j,l)}=0$；

② 若 $r_P^{(j,l)}\neq 0$，则 $\overline{r}_P^{(j,l)}$ 的值为平台 P_j 处理的所有任务在第 l 类功能上的实际功能能力使用值的均值，即 $\overline{r}_P^{(j,l)}=\frac{1}{n_{TP}^{(i,j)}}\sum_{i=1}^{I}\frac{\min(r_P^{(j,l)},r_T^{(i,l)})}{r_P^{(j,l)}}\times m_{TP}^{(i,j)}$，$n_{TP}^{(i,j)}=\sum_{i=1}^{I}m_{TP}^{(i,j)}$。

所有平台的资源利用率为各个平台的资源利用率的均值，即

$$U=\frac{1}{J}\sum_{j=1}^{J}U_j \tag{8}$$

2.3.3 数学模型建立

综上所述，平台资源规划问题的数学模型描述为

$$\begin{cases}\min t_{\text{makespan}}=\max\limits_{i=1}^{I}(t_{\text{Start}}^{(i)}+t_{\text{Process}}^{(i)})\\ \max U=\frac{1}{J}\sum\limits_{j=1}^{J}(\frac{1}{\hat{L}_j}\sum\limits_{l=1}^{L}(\frac{1}{n_{TP}^{(i,j)}}\sum\limits_{i=1}^{I}\frac{\min(r_P^{(j,l)},r_T^{(i,l)})}{r_P^{(j,l)}}\times m_{TP}^{(i,j)}))\\ t_{\text{Start}}^{(i_2)}\geqslant t_{\text{Start}}^{(i_1)}+t_{\text{Process}}^{(i_1)},o_{i_1i_2}=1\\ t_{\text{Start}}^{(i)}\geqslant\max\limits_{j\in\{k|m_{TP}^{(i,k)}=1,l(k)\neq 0,k=1,2,\cdots,J\}}(t_{\text{Start}}^{(l(j))}+t_{\text{Process}}^{(l(j))}+d_{i,l(j)}^{(j)}\times\frac{\sqrt{(x_T^{(l(j))}-x_T^{(i)})^2+(y_T^{(l(j))}-y_T^{(i)})^2}}{v_j})\\ \sum\limits_{j=1}^{J}r_P^{(j,l)}m_{TP}^{(i,j)}\geqslant r_T^{(i,l)}\\ \sum\limits_{i_2=0}^{I}d_{i_1i_2}^{(j)}\leqslant 1\\ \sum\limits_{i_1=1}^{I}d_{i_1i_2}^{(j)}=m_{TP}^{(i_2,j)}\\ m_{TP}^{(i,j)},d_{i_1i_2}^{(j)}\in\{0,1\}\\ i_1,i_2,i=1,2,\cdots,I;j=1,2,\cdots,J;l=1,2,\cdots,L\end{cases} \tag{9}$$

3 平台资源规划问题的求解

3.1 算法思想及流程

平台资源规划求解有两种方案：一是串行规划生成方案(Serial Schedule Generation Scheme，SSGS)，二是并行规划生成方案(Parallel Schedule Generation Scheme，PSGS)。使用SSGS可以获得平台资源规划最优解，而使用PSGS通常可以产生平台空闲时间较少的结果，但不一定包含最优解[7]。

DLS是典型的SSGS算法。本文基于DLS的思想提出了基于DLS和ACO的平台资源规划方法，其基本流程如图1所示。

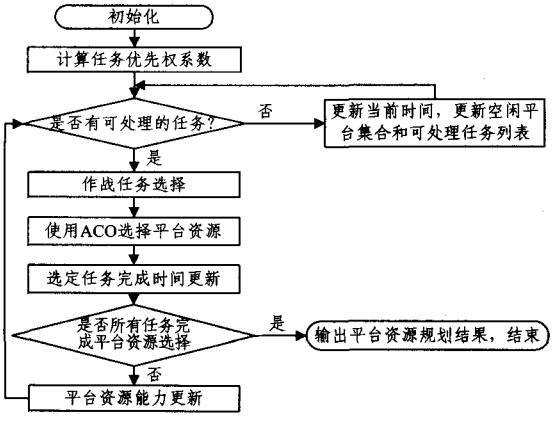

图1 平台资源规划方法的流程

本文提出的基于DLS和ACO的平台资源规划方法有两个重要环节：一是在可选任务集合中选择一个任务；二是使用ACO为该任务选择平台资源。

3.2 计算任务优先权系数及作战任务选择

当一个任务 T_i 的所有前导任务都处理完成时，该任务便进入可以处理的任务集合 T_{ready} 中按序进行平台资源规划。从 T_{ready} 中选择任务的依据是任务优先权系数，任务的优先权系数越大，表示该作战任务的优先级越高，其就会被优先选择进行平台规划。

计算任务优先权系数的依据是任务信息以及它们之间的序列关系。任务优先权系数计算算法主要包括[3]层次分配(Level Assignment，LA)算法、关键路径(Critical Path，CP)算法、加权长度(Weighted Length，WL)算法和加权关键路径(Weighted Critical Path，WCP)算法。CP算法、WL算法和WCP算法是同一类方法，其中CP算法是早期经典的任务优先权系数计算方法，其缺陷是没有考虑任务的后续任务结构；WL算法和WCP算法都考虑到了当前任务执行复杂度、后续任务执行复杂度和后续任务数量这3个因素，它们在确定任务优先权系数上效果较好，两个算法的效果基本相当，但WL算法具有较少的计算复杂度。

因此，本文采用WL算法计算任务的优先权系数。在WL算法中，任务的优先权系数取决于任务的处理时间、任务的直接后续任务数量以及它们的优先权系数。用WL算法计算各个任务优先权系数的公式为[3]

$$p_T(i)=t_{\text{Process}}^{(i)}+\max_{j\in\text{OUT}(i)}p_T(j)+\frac{\sum\limits_{i\in\text{OUT}(i)}p_T(j)}{\max\limits_{j\in\text{OUT}(i)}p_T(j)} \tag{10}$$

式中，$p_T(i)$ 为用WL算法求得的任务 T_i 的优先权系数；$\text{OUT}(i)$ 为任务序列图中任务 T_i 的直接后续任务集；$t_{\text{Process}}^{(i)}$ 为任务处理时间。

3.3 任务的平台选择

在可分配任务集合中，选择任务优先权系数高的一个任务。本小节将研究如何在空闲平台集合 P_{free} 中为该任务选择最佳的平台资源。任务的平台选择是平台资源规划问题的关键环节，单个任务的平台选择方案的好坏对整个任务的平台规划结果影响较大。单个任务的平台选择问题(Platform Selection Problem for Single Task，PSPFST)是多维0-1背包规划问题[8]的变体。

蚁群算法(Ant Colony Optimization Algorithm，ACO)通过对自然界中真实蚁群觅食机制的模拟，将规划问题中的启发信息表达到蚁群的信息素释放/挥发机制中；通过在规划过程中引入正反馈机制，大大提高了算法的收敛速度。对于平台资源规划这类问题，还需结合具体应用，在算法实现过程中做出相应的修改。

3.3.1 编码方案及候选解构造策略

对问题求解空间的编码是利用蚁群算法求解平台资源规划问题的基础。本文使用二值编码方法对每一个PSPFST的规划结果进行编码。定义二值变量 $x_{ij}\in\{0,1\}$，$x_{ij}=1$ 表示平台资源分配给当前处理的任务 T_i，$x_{ij}=0$ 表示平台资源未分配给当前处理的任务 T_i，则对于 T_i，其二值编码的平台分配方案表示为 $X_i=(x_{i1},x_{i2},\cdots,x_{iJ})$。

对于任务 T_i 而言，每个平台作为待选项都有两种情况：被选择或不被选择。不同于遗传算法在构造候选解时随机产生初始候选解以及使用交叉、变异操作产生新的候选解，ACO构造候选解的依据是信息素。定义任务 T_i 的平台选择问题的信息素模型为 $\{\tau_{jr}\,|\,r\in\{0,1\},j=1,2,\cdots,J\}$，$\tau_{j0}$ 对应 $x_{ij}=0$，τ_{j1}

对应 $x_{ij}=1$。本文将 PSPFST 描述成图 2 所示的蚂蚁行走路线图[9]，即每只蚂蚁按照平台序号从 1 走到 $J+1$（$J+1$ 表示结束节点，不代表实际平台）的顺序，依次在两个节点之间选择一条路径，完成一个候选解的构造。

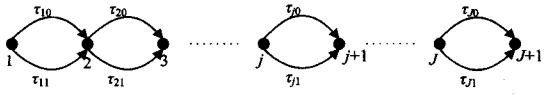

图 2　任务的平台选择中的蚂蚁行走路线图

对于任务 T_i，在设计选择平台资源的蚁群算法时，首先可以筛选候选平台，得到进一步的可选平台集。为了避免蚂蚁选择无效路径，本文设置了一个禁忌表 $tabu$，对于 $\forall j\in tabu$，蚂蚁在行走过程中选择路径 τ_{j0}。该禁忌表中包含两类平台：

(1) $j\in\{h\mid h\notin P_{\text{free}}\}$，即不在空闲平台集合中的平台；

(2) $j\in\{h\mid \exists s; r_P^{(h,s)}=0, s\in\{l\mid r_T^{(i,l)}\neq 0\}, h\in P_{\text{free}}\}$，即在任意一个功能类型上都不能为任务 T_i 提供功能能力的平台。

在对信息素进行初始设置时，$\forall j\in tabu$，设置 τ_{j0} 为一个较大的常数，$\tau_{j1}=0$；$\forall j\notin tabu$，设置 $\tau_{j0}=\tau_{j1}=0.5$。

ACO 中蚂蚁按照信息素分布进行路径选择。在求解平台资源规划问题的 ACO 中，路径选择是任务 T_i 对当前平台做出是否选择的判断。

在迭代步 t，对于在禁忌表中的平台，蚂蚁 k 设置从节点 j 到 $j+1$ 的行走路径 τ_{j1} 的概率 $p_{j1}^{(k)}(t)=0$；对于不在禁忌表中的平台，蚂蚁 k 按式(11)计算从节点 j 到 $j+1$ 的行走路径 τ_{j1} 的概率，即计算 $x_{ij}=1$ 的概率。

$$p_{j1}^{(k)}(t)=\frac{\tau_{j1}(t)}{\tau_{j0}(t)+\tau_{j1}(t)} \tag{11}$$

由于蚂蚁 k 从节点 j 到 $j+1$ 的行走路径仅有两条，因此 $x_{ij}=0$ 的概率为 $p_{j0}^{(k)}(t)=1-p_{j1}^{(k)}(t)$。

3.3.2　不可行候选解的修正策略

输入一个不可行解 $\mathbf{X}_{\text{old}}^{(i)}=(x_{i1},x_{i2},\cdots,x_{iJ})$

1　$\mathbf{X}^{(i)}\leftarrow\mathbf{X}_{\text{old}}^{(i)}$; feasible=false; 令 s=0;
2　按式（11）计算不在禁忌表中的平台分配给任务的优先权，并按递减顺序排序，这些平台的序号构成 $\mathbf{Q}_i$（$n_i=|\mathbf{Q}_i|=J-|tabu|$）;
3　计算 $R_l=\sum_{j=1}^{J}r_{\text{P}}^{(j,l)}x_{ij}, \forall l\in\{1,2,\cdots,L\}$
4　while feasible=false do /*增加平台*/
5　　s=s+1;
6　　j=$\mathbf{Q}_i(s)$;
7　　if x_{ij}=0 then
8　　x_{ij}=1;
9　　$R_l=R_l+r_{\text{P}}^{(j,l)}, \forall l\in\{1,2,\cdots,L\}$;
10　　if $R_l\geq r_{\text{T}}^{(i,l)}, \forall l\in\{1,2,\cdots,L\}$ then
11　　feasible=true; /*满足任务资源需求，跳出 for 循环*/
12　　endif
13　endif
14 endwhile
15 $\mathbf{X}_{\text{new}}^{(i)}\leftarrow\mathbf{X}^{(i)}$
输出可行解 $\mathbf{X}_{\text{new}}^{(i)}$

图 3　不可行候选解修正策略的伪代码

由于算法搜索过程的随机性，按照表 1 所列编码方式构造的候选解可能是不可行解。不可行解指的是至少有一个约束条件没有得到满足。如果生成的不可行解过多，将会导致算法的求解效率下降。在所有蚂蚁完成了一次解的构造过程后，ACO 检查在一次迭代过程中产生解的可行性，并利用修正策略将不可行解转化为可行解。

本文设计了修正算子(repair operator)，并利用式(11)计算得到的值按递减顺序检查所有的平台到任务的分配变量。如果 $x_{ij}=0$，则将平台 P_j 增加到分配给任务 T_i 的平台集合中；若新增平台使得任务功能能力需求得到了满足，则停止检查。

图 3 描述了不可行候选解修正策略（将一个不可行解 $X_{\text{old}}^{(i)}$ 转化为一个可行解 $X_{\text{new}}^{(i)}$）的伪代码。

3.3.3　信息素更新过程

在每一迭代步 t，所有 n 只蚂蚁在构造了候选解，并对其进行了修正处理和局部优化之后，可以得到 n 个可行解。如果本次迭代最优解 S_{tb} 优于全局历史最优解 S_{gb}，则令 $S_{gb}=S_{tb}$。

当蚂蚁完成一次解构造和处理之后，需要对路径上的信息素浓度进行更新处理，模仿人类记忆的特点，消弱旧信息；根据本次迭代的路径选择情况，加强相关路径上的信息素浓度。因此，ACO 中信息素更新过程包括两个部分：

(1)对所有路径上的信息素都进行挥发处理，即

$$\tau_{jr}(t+1)=(1-\rho)\tau_{jr}(t), r\in\{0,1\}, j\notin tabu \tag{12}$$

式中，$\rho(\rho\in(0,1))$是信息素的挥发系数。

(2)在挥发处理基础上，对本次迭代最优解 S_{tb} 对应选择的路径进行加强处理，即

$$\begin{cases}\tau_{j0}(t+1)=\tau_{j0}(t+1)+Q, j\notin tabu, x_{ij}=0\\ \tau_{j1}(t+1)=\tau_{j1}(t+1)+Q, j\notin tabu, x_{ij}=1\end{cases} \tag{13}$$

式中，Q 是一个常量，是本次迭代中最佳路径上信息素量的增量。

3.3.4　适应度函数构造

每个蚂蚁构造的候选解 $X_i=(x_{i1},x_{i2},\cdots,x_{iJ})$ 代表任务 T_i 的平台选择结果，候选解的适应值是根据选择平台情况计算出的平台组合性能。适应度函数应当体现平台资源规划的目标，即全部任务完成的截止时间最小化和平台资源的利用率最大化。因此，适应度函数构造应当考虑任务选择平台的时间优先权和平台功能能力优先权。另外，作为单个任务的平台选择，还要考虑当前任务的平台资源选择对后续任务的影响，以避免当前任务选择的平台极大地影响后续任务的执行。

基于上述分析，本文构造的适应度函数应当包含以下 3 个因素：任务选择平台的时间优先系数、平台功能能力优先系数和后续任务对平台需求程度。

(1)时间优先系数

在平台资源规划过程中，存在两种平台：一种是已经分配了任务的平台，该类平台需要在上一个任务完成后进行任务位置转换；另一种是还未分配任务的平台，该类平台不需要从一个任务位置转换到当前任务，即可以设定该类平台在当前任务发生时已经达到任务位置。

任务 T_i 的开始时间 $t_{\text{Start}}^{(i)}$ 是指选择的平台中最晚达到任务 T_i 发生位置的时间，且不能早于当前时刻。即

$$t_{\text{Start}}^{(i)}=\max\Big(t_{\text{current}}, \max_{\substack{j\in\{k\mid k\notin tabu, l(k)\neq 0,\}\\ k=1,2,\cdots,J}}\Big\{t_{\text{Start}}^{(l(j))}+t_{\text{Process}}^{(l(j))}+\frac{\sqrt{(x_T^{(i)}-x_T^{(l(j))})^2+(y_T^{(i)}-y_T^{(l(j))})^2}}{v_j}\Big\}\Big) \tag{14}$$

式中，t_{current} 为当前时刻；l_j 表示平台 P_i 之前处理任务中的最

后一个任务，若 T_i 为平台 P_i 处理的第一个任务，则 $l_j=0$。

$t_{\text{Start}}^{(i)}$ 的值越小，越有利于完全任务尽早完成。为了与其它指标聚合，对一个候选解下的任务开始时间进行去量纲处理。由式(14)可知，任务 T_i 可能的最早开始时间 $t_{\text{earliest}}^{(i)}$ 为 t_{current}，可能的最晚开始时间 $t_{\text{latest}}^{(i)}$ 为空闲平台中最晚达到任务 T_i 位置的平台的达到时间和 t_{current} 的较大者，即

$$t_{\text{earliest}}^{(i)}=t_{\text{current}};\ t_{\text{latest}}^{(i)}=\max\Big(t_{\text{current}},\ \max_{\substack{j\in\{k\mid k\notin tabu,\,l(k)\neq 0,\\ k=1,2,\cdots,J\}}}\Big\{t_{Start}^{(l(j))}+t_{\text{Process}}^{(l(j))}+\frac{\sqrt{(x_T^{(i)}-x_T^{(l(j))})^2+(y_T^{(i)}-y_T^{(l(j))})^2}}{v_j}\Big\}\Big) \tag{15}$$

由式(14)和式(15)，构造时间优先系数为

$$C_T=\frac{t_{\text{Start}}^{(i)}-t_{\text{earliest}}^{(i)}}{t_{\text{latest}}^{(i)}-t_{\text{earliest}}^{(i)}} \tag{16}$$

(2)平台功能能力优先系数

平台功能能力优先系数是考察平台对任务满足程度的指标。候选解 X_i 中为任务 T_i 选择的平台应当具有较小的功能能力冗余 R_{FC}，其计算公式为

$$R_{FC}=\sqrt{\sum_{l=1}^{L}\Big(\sum_{j\in\{k\mid x_{ik}=1,k=1,2,\cdots,J\}}r_P^{(j,l)}-r_T^{(i,l)}\Big)^2} \tag{17}$$

当所有不属于禁忌表中的平台都被选择时，功能能力冗余最大，为

$$R_{FC\_All}=\sqrt{\sum_{l=1}^{L}\Big(\sum_{j=1,2,\cdots,J,j\notin tabu}r_P^{(j,l)}-r_T^{(i,l)}\Big)^2} \tag{18}$$

根据式(14)和式(15)，对候选解 X_i 中选择平台的功能能力冗余进行去量纲处理，构造平台功能能力优先系数为

$$C_{PF}=\frac{R_{FC}}{R_{FC\_All}} \tag{19}$$

(3)后续任务对平台需求程度

为了避免单个任务对平台的选择造成后续任务由于缺少必要资源而等待较长时间，应当在当前任务选择平台时，考虑后续任务对平台的需求。后续任务是指当前可处理任务集合 T_{ready} 除去任务 T_i，再加上任务 T_i 完成后新增的可处理任务，即后续任务集合为 T'_{ready}。

后续任务 T'_{ready} 对候选解 X_i 中为任务 T_i 选择的平台的平均功能能力需求为

$$N_{FC}=\frac{1}{n_{CP}}\times\sum_{j\in\{k\mid x_{ik}=1,k=1,2,\cdots,J\}}\ \sum_{i\in T'_{\text{ready}}}\ \sum_{l=1}^{L}\min(r_P^{(j,l)},r_T^{(i,l)}) \tag{20}$$

式中，n_{CP} 为 X_i 中元素等于 1 的个数，即为任务 T_i 选择的平台的数目。

当所有不属于禁忌表中的平台都被选择时，后续任务对平台的功能能力需求最大，为

$$N_{FC\_All}=\frac{1}{n_P}\times\sum_{j=1,2,\cdots,J,j\notin tabu}\ \sum_{i\in T'_{\text{ready}}}\ \sum_{l=1}^{L}\min(r_P^{(j,l)},r_T^{(i,l)}\} \tag{21}$$

式中，n_P 为不属于禁忌表中的所有平台的数目。

由式(20)和式(21)，本文将后续任务 T'_{ready} 对候选解 X_i 中为任务 T_i 选择的平台的平均需求程度进行去量纲处理，构造后续任务对平台需求程度为

$$D_{PN}=\frac{N_{FC}}{N_{FC\_All}} \tag{22}$$

综合式(16)、式(19)和式(22)，本文构造的适应度函数为

$$F=\omega_1\times C_T+\omega_2\times C_{PF}+\omega_3\times D_{PN} \tag{23}$$

式中，$\omega_1,\omega_2,\omega_3\in(0,1)$，这 3 个参数表示规划者在 3 个因素上的主观倾向。适应度函数值越小，为任务 T_i 选择的平台组合越适合于任务 T_i。

3.3.5 PSPFST 求解流程描述

本文采用 ACO 算法求解 PSPFST，其实现步骤如下：

初始化

(1)根据任务信息和平台信息设置禁忌表 $tabu$；

(2)设置信息素 $\{\tau_{jr}\mid r\in\{0,1\},j=1,2,\cdots,J\}$、信息素的挥发系数 ρ、信息素量的增量 Q、蚂蚁数目 n 和迭代次数预定值 $N_{\max}$；

(3)令当前迭代步 t、本次迭代最优解 S_{tb}、全局历史最优解 S_{gb}、$x_{ij}^{(k)}$ 均为 0。

步骤 1　生成 n 只蚂蚁，并将所有蚂蚁置于节点 1。

步骤 2　每只蚂蚁进行候选解构造。令 $t=t+1$。$\forall j\in tabu$，$p_{j1}^{(k)}(t)=0$；$\forall j\notin tabu$，按式(11)计算路径选择概率 $p_{j1}^{(k)}(t)$，并根据赌轮盘规则选择从节点 j 到节点 $j+1$ 的路径，即确定 $x_{ij}^{(k)}=0$ 或 $x_{ij}^{(k)}=1$。每只蚂蚁在候选解构造过程中，任务的功能能力需求得到满足，则停止蚂蚁行走。

步骤 3　判断每只蚂蚁构造的待选解是否满足平台功能能力约束，若是，则转步骤 5；否则，进入步骤 4。

步骤 4　利用不可行候选解修正策略，将一个不可行解转化为可行解。

步骤 5　引入逆转变异算子对待选解进行局部优化，改进待选解的质量。

相关研究表明，引入局部优化策略能够较大程度地提升蚁群算法搜索效率，并被证明它是非常有效的改进解的质量的方法[7]。本文引入了逆转变异算子(2-opt 局部优化)[11]对经修正处理后的待选解进行局部优化。如果所得的新待选解满足约束条件，且优于对应的原有平台资源规划结果，则以局部优化后的新待选解替换原有计划。

步骤 6　按照式(23)计算各蚂蚁构造的待选解的适应度函数值，从本次迭代各蚂蚁构造的待选解中选出本次迭代的最优解 S_{tb}。如果 S_{tb} 优于全局历史最优解 S_{gb}，则令 $S_{gb}=S_{tb}$。

步骤 7　按照式(12)和式(13)进行信息素更新。

步骤 8　判断 t 是否等于 $N_{\max}$。若是，则终止算法执行，并输出此时的全局历史最优解；否则，进入步骤 9。

步骤 9　若 ACO 搜索已收敛，则除了保持 t 值不变外，ACO 算法重启；否则，转步骤 1。

4　算例分析

以一次登岛战役作战为例，其作战使命任务为登陆抢占敌方的机场和港口，为后续部队向纵深推进扫清障碍。根据这个使命任务和作战环境可以将作战任务分解为 18 个子任务。作战可以使用的平台资源包括[11]：驱逐舰、护卫舰、巡洋舰、两栖工兵分队、步兵分队、扫雷舰、侦察卫星等 15 类、20 个作战平台。功能设定为 8 种，分别为防空作战功能、反舰作战功能、反潜作战功能、地面攻击功能、火力支援功能、重装攻击功能、攻势布扫雷作战功能、探测识别功能，即任务数目 $I=18$，平台数目 $J=20$，功能数目 $L=8$。

作为平台资源规划的输入信息，任务属性数据、平台属性数据及任务间的逻辑关系见文献[11]。

假设虚拟的终止任务的任务优先权系数为 0，使用式(10)计算得到任务优先权系数，如表 1 所列。

表 1　任务的优先级系数

任务	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9
优先权系数	78.00	78.00	58.00	58.00	81.80	81.80	70.80	58.55	47.00
任务	T_{10}	T_{11}	T_{12}	T_{13}	T_{14}	T_{15}	T_{16}	T_{17}	T_{18}
优先权系数	47.00	26.00	26.00	36.00	36.00	15.00	15.00	31.00	20.00

本文将 DLS&ACO 算法与 MDLS、MPLDS 进行了比较。这些算法分别来自于文献[3,4]。

分别使用 MDLS、MPLDS 以及本文算法对算例进行平台资源规划，规划结果如表 2 所列。其中，本文算法所设置的 ACO 参数为 $n=20$，$N_{max}=100$，$\rho=0.3$，$Q=0.01$，$\tau_{jr}(0)=0.5$，$\tau_{j1}(0)=0.5$(设置禁忌表中 $\tau_{j1}(0)=0$，$\tau_{j0}(0)=1$)；$\omega_1=0.4$，$\omega_2=0.3$，$\omega_3=0.3$；MPLDS 的参数为 $\lambda=1$。表 2 中的每个项为$(i,t_{Start}^{(i)})$，其分别表示任务序号和开始时间。

表 2　算例的平台资源规划结果对比

平台	不同算法下的平台资源规划结果		
	MDLS	MPLDS	DLS & ACO
P_1	(3,0),(11,30.08),(10,90.15)	(17,0),(11,37.12)	(3,0),(9,30),(16,120.15)
P_2	(1,0),(2,60.15)	(1,0),(2,60.15)	(1,0),(2,60.15)
P_3	(1,0),(12,106.35)	(2,60.15)	(5,0),(8,27.29),(10,90.15)
P_4	(18,10),(12,106.35)	(8,32.30),(12,48.78),(15,116.39)	(18,10),(12,46.98)
P_5	(6,0),(7,20.08),(9,30.08),(12,106.35),(15,156.24)	(6,0),(7,20.08),(9,47.90),(10,90.15),(16,137.93)	(6,0),(8,27.29),(16,120.15)
P_6	(2,60.15)	(1,0),(9,47.90)	(2,60.15)
P_7	(4,0),(8,43.01),(15,156.24)	(1,0),(18,66.39),(14,102.61)	(4,0),(8,27.29),(14,100.15)
P_8	(6,0),(2,60.15)	(6,0),(7,20.08),(11,37.12),(13,96.39)	(6,0),(7,13.40),(2,60.15)
P_9	(17,0),(11,30.08),(14,100.15)	(3,0),(8,32.30),(12,48.78),(15,116.39)	(17,0),(11,24.08),(13,40.82)
P_{10}	(7,20.08),(13,40.82),(16,120.15)	(4,0),(2,60.15),(16,0)	(7,13.40),(12,46.98)
P_{11}	(5,0),(7,20.08),(9,30.08),(12,106.35)	(5,0),(7,20.08),(8,32.30),(12,48.78),(13,96.39)	(1,0),(10,90.15)
P_{12}	(5,0),(7,20.08),(9,30.08),(12,106.35)	(5,0),(7,20.08),(8,32.30),(12,48.78),(10,90.15)	(1,0),(10,90.15)
P_{13}	(5,0),(9,30.08)	(5,0),(7,20.08),(8,32.30),(12,48.78),(10,90.15)	(1,0),(10,90.15)
P_{14}	(5,0),(11,30.08)	(5,0),(11,37.12),(18,66.39)	(5,0),(11,24.08)
P_{15}	(1,0),(13,40.82)	(1,0),(2,60.15),(14,102.61)	(1,0),(13,40.82),(14,100.15)
P_{16}	(17,0),(2,60.15)	(17,0),(2,60.15),(14,102.61)	(17,0),(2,60.15)
P_{17}	(18,10),(8,43.01),(14,100.15)	(7,20.08),(18,66.39),(13,96.39),(16,137.93)	(18,10),(15,70.06)
P_{18}	(6,0),(8,10),(15,156.24)	(6,0),(8,32.30),(18,66.39),(15,116.39)	(6,0),(11,24.08),(15,70.06)
P_{19}	(1,0),(12,106.35),(16,120.15)	(17,0),(9,47.90),(10,90.15),(16,137.93)	(1,0),(15,70.06)
P_{20}	(1,0),(12,106.35),(16,120.15)	(17,0),(9,47.90),(15,116.39)	(7,13.40),(12,46.98),(16,120.15)
算法性能	任务完成截止时间：171.24h 平台利用率：0.53	任务完成截止时间：152.93h 平台利用率：0.52	任务完成截止时间：135.15h 平台利用率：0.58

由表 2 给出的 3 种算法的平台规划结果比较可知，本文提出的 DLS & ACO 算法相比于 MDLS 和 MPLDS，其任务执行过程更为紧凑，具有最短的全部任务完成截止时间，同时具有较高的平台资源利用率。

结束语　本文结合 DLS 和 ACO 提出了一种面向作战任务的平台资源规划算法。DLS&ACO 算法使用 DLS 作为问题求解的整体框架，分别采用 WL 算法和 ACO 算法以实现任务选择和选定任务的平台资源选择。使用 ACO 为选定任务选择平台资源是平台资源规划问题的关键环节，本文设计了 ACO 的二进制编码方案及候选解构造策略、不可行候选解的修正策略、信息素更新方法，构造了适应度函数，并对算法的复杂度进行了分析。仿真实验结果表明，基于 DLS 和 ACO 的平台资源规划算法具有较高的性能。

然而，ACO 算法作为一种新的智能进化算法，还没有形成系统分析的方法和坚实的数学基础，各种实验参数的设置还缺乏理论上的指导，这有待进一步研究。

参 考 文 献

[1] 刘跃峰，张安. 有人机/无人机编队协同任务分配方法[J]. 系统工程与电子技术，2010，32(3)：584-588

[2] Huy B, Han Xu, Mandal S, et al. Optimization-based decision support algorithms for a team-in-the-loop planning experiment [A]// Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics[C]. San Antonic, TX, USA, 2009：4684-4689

[3] Levchuk G M, Levchuk Y N, Luo Jie, et al. Normative design of organizations-partI: mission planning [J]. IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans, 2002, 32(3)：346-359

[4] 阳东升，张维明，刘忠，等. 战役任务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

[5] 陈洪辉，赵亮，芮红，等. 作战任务和资源间的匹配模型及求解算法研究[J]. 系统工程与电子技术，2008，30(9)：1712-1716

[6] 刘宏芳，阳东升，刘忠，等. 全局资源视图生成战术资源视图的方法研究[J]. 计算机科学，2006，33(6)：119-123

[7] Merkle D, Middendorf M, Schmeck H. Ant colony optimization for resource-constrained project scheduling[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(4)：333-346

[8] Fréville A. The multidimensional 0-1 knapsack problem: An overview[J]. European Journal of Operational Research, 2004 (155)：1-21

[9] Kong Min, Tian Peng, Kao Yu-cheng. A new ant colony optimization algorithm for the multidimensional knapsack problem[J]. Computers & Operations Research, 2008, 35：2672-2683

[10] 苏菲，陈岩，沈林成. 基于蚁群算法的无人机协同多任务分配[J]. 航空学报，2008，29(增刊)：S184-S191

[11] Yu Fei-li, Tu Fang, Pattipati K R. A novel congruent organizational design methodology using group technology and a nested genetic algorithm[J]. IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans, 2006, 36(1)：5-18