

资源不确定军事任务计划预测调度模型与算法*

张迎新, 陈超, 刘忠, 石建迈

(国防科技大学 信息系统工程重点实验室, 湖南 长沙 410073)

摘要:针对军事任务计划执行环境中普遍存在的资源不确定性,提出了基于资源缓冲区的军事任务计划预测调度算法。算法首先基于平台有效资源功能向量进行任务—平台分配,分配过程中通过调节平台有效资源功能向量,获得不同的平台资源缓冲区分配方案;然后基于 NSGA-II 算法框架对军事任务计划进行多目标求解,进而获得问题的 Pareto 最优解集。文章通过仿真算例对算法的可行性和有效性进行验证,实验结果表明,该算法能够有效求解资源不确定军事任务计划问题。

关键词:军事任务计划;作战行动序列;预测调度;资源缓冲区

中图分类号:C394 **文献标志码:**A **文章编号:**1001-2486(2013)03-0030-06

Method for modeling and solving military mission planning with uncertain resource availability

ZHANG Yingxin, CHEN Chao, LIU Zhong, SHI Jianmai

(Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha 410073, China)

Abstract: According to the ubiquitous uncertainty of resource availability, a multi-objective military mission planning model with uncertain resource availability is proposed based on reasonable hypotheses. A proactive schedule method based on resource buffer was designed according to the special features and particularity of the model. In the method, the resource buffer was added into the project during the task-platform allocation in order to minimize the disadvantage of resource uncertainty; the multi-objective evolutionary algorithm NSGA-II was adopted to calculate the Pareto Front of the problem. Finally, an example was illustrated to show the feasibility and practicability of the proposed model and method.

Key words: military mission planning; course of action; proactive schedule; resource buffer

军事任务计划问题的实质是对平台资源进行优化调度,确定每个任务的开始时间和执行资源^[1]。近年来国内外关于军事任务计划问题的研究,主要集中在静态假设条件下任务—平台资源的优化调度^[2-3]。然而,在静态假设条件下获得的“最优”军事任务计划方案在实际执行过程中,往往很难取得预想的结果,甚至变得不再可行。分析其主要原因在于,军事任务计划方案的执行环境中往往存在着很多动态不确定因素,如任务执行时间变动等,如何在军事任务计划中考虑这些不确定因素,已成为解决实际问题的关键。本文将影响军事任务计划方案执行的不确定因素归结为平台资源的不确定性。在平台资源不确定条件下,如何通过预测调度生成一种“受到保护”的军事任务计划方案是本文的研究目的。

预测调度是一种通过考虑未来不确定因素并预先防范来增强方案承受不确定干扰能力的调度方法。预测调度方法主要采用基于冗余的技术,

即在方案中通过增加时间或资源缓冲区来防止不确定干扰的过度蔓延。现有的预测调度方法主要采用基于时间缓冲区的冗余技术,资源不确定往往被归结为任务工期的不确定性进行处理^[4]。文献[5-6]对项目调度领域有关增加时间缓冲区和资源缓冲区的启发式算法进行了很好的综述。

平台资源不确定性是现实军事任务计划中普遍存在且必须考虑的一类典型不确定因素。在平台资源不确定条件下,增加资源缓冲区或时间缓冲区是增强军事任务计划方案承受不确定干扰能力的两种可选途径。结合军事任务计划问题的特点,本文提出了基于资源缓冲区的军事任务计划预测调度算法。

1 资源不确定军事任务计划问题描述与建模

* 收稿日期:2012-12-11

基金项目:国家自然科学基金资助项目(71201169, 71101149, 71001105, 71031007);湖南省自然科学基金资助项目(11JJ4065)

作者简介:张迎新(1983—),男,山东齐河人,博士生, E-mail: zhyingxin@163.com;

陈超(通信作者),男,讲师,博士, E-mail: chenc1977@gmail.com

1.1 问题参数描述

典型的军事计划问题可以这样描述:给定一个任务图 G 和可用平台集合 P , 任务图 G 确定了需要处理的所有任务 $T = (t_1, t_2, \dots, t_J)$ (J 表示任务数量) 以及任务之间的逻辑关系, 图中结点表示任务, 结点之间的有向弧表示任务之间的逻辑关系。同时, 每个任务 t_i 具有确定的任务属性, 包括任务执行时间需求 dur_i 、任务位置坐标 (x_i, y_i) , 以及任务执行资源需求向量 $Trr_i = \{trr_{i1}, trr_{i2}, \dots, trr_{iN}\}$ (N 表示资源种类数量), 其中 trr_{in} ($n \in N$) 表示任务 t_i 需要资源类型 r_n ($n \in N$) 的数量。可用平台集合 $P = \{p_1, p_2, \dots, p_K\}$ (K 表示平台数量) 确定执行任务可获取的所有平台的集合。同时, 每个平台 p_k ($k \in K$) 具有给定的平台属性, 包括最大运动速度 v_k 、平台资源功能向量 $prc_k = \{prc_{k1}, prc_{k2}, \dots, prc_{kN}\}$ 等, 其中 prc_{kn} ($n \in N$) 表示平台 p_k 载有资源类型 r_n ($n \in N$) 的数量。需要注意的是, t_1 和 t_J 都是虚任务, 分别表示整个计划的开始和结束, 其任务执行时间需求和执行资源需求均为零。

任务执行必须满足如下约束: 资源约束、时序约束和完成期限约束。资源约束要求分配执行某个任务所有平台的平台资源功能向量之和不小于该任务的执行资源需求向量; 时序约束要求任务的执行必须满足任务图中给定的逻辑关系; 完成期限约束要求任务 t_j 的完成时间不大于给定的任务完成期限 δ_j 。

1.2 基本假设与符号

(1) 基本假设

资源不确定军事任务计划问题存在的难点较多, 为简化和说明问题, 作如下合理假设:

- 1) 每个任务只能选择一种执行模式, 任务一旦开始, 其执行模式不能改变;
- 2) 一个平台每次只能处理一个任务;
- 3) 同类平台资源的故障时间和修复时间服从相同并且独立的指数分布。

(2) 模型参数与变量

模型中的其余相关参数定义如下:

st_i : 任务 t_i 的计划开始时间;

ST_i : 任务 t_i 的实际开始时间, 考虑到平台资源失效引起的任务完成时间的不确定性, ST_i 是随机变量;

ω_i : 任务 t_i 的不稳定权重, 表示 ST_i 偏离 st_i 引起的单位费用成本;

$Succ_i$: 任务 t_i 的 (直接和间接) 后续任务

集合;

MTF_{kn} : 平台 p_k 中资源 r_n 的单位平均故障时间;

MTR_{kn} : 平台 p_k 中资源 r_n 的单位平均修复时间;

模型中的决策变量 α_{ki} 、 β_{ij} 、 φ_{ijk} 定义如下:

如果平台 p_k 被分配执行任务 t_i , 则 $\alpha_{ki} = 1$, 否则 $\alpha_{ki} = 0$;

如果任务 t_i 必须在任务 t_j 完成之后才能开始, 则 $\beta_{ij} = 1$, 否则 $\beta_{ij} = 0$;

如果平台 p_k 在完成的任务 t_i 之后还需要执行任务 t_j , 则 $\varphi_{ijk} = 1$, 否则 $\varphi_{ijk} = 0$ 。

1.3 模型描述

综上所述, 资源不确定军事任务计划问题的数学描述如下^[2]:

$$f_1 = \max P(ST_j < \delta_j) \quad (1)$$

$$f_2 = \min \sum_{i=2}^J \omega_i \cdot E | ST_i - st_i | \quad (2)$$

$$\sum_{k=1}^K \alpha_{ki} \cdot Prc_k \geq Trr_i, i \in J \quad (3)$$

$$st_i + dur_i + \varphi_{ijk} \cdot (L_{ij}/v_k + \beta_{ij} \cdot Y) \leq \beta_{ij} \cdot Y + st_j \quad (4)$$

$$L_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (5)$$

$$st_j \leq \delta_j \quad (6)$$

$$i, j \in J; k \in K \quad (7)$$

式(1)和式(2)分别定义了两种常用的方案鲁棒性度量, 式(1)要求任务计划在期限内完成的概率最大; 式(2)要求任务实际开始时间偏离计划开始时间的加权和最小, 式中 $E|\cdot|$ 表示期望值; 式(3)表示任务的资源约束, 它要求分配执行每个任务的平台组合的资源功能向量和不小于该任务的执行资源需求向量; 式(4)表示任务的时序约束, 它要求每个任务的执行必须满足如下两个条件: 该任务之前的所有任务都已完成; 被分配执行该任务的所有平台都已到达指定位置, 式中 Y 为无穷大辅助变量; 式(5)定义了两个非虚任务之间的距离; 式(6)表示任务计划的完成期限约束, 它要求所有任务的结束时间不大于给定的任务完成期限 δ_j 。

1.4 替代目标函数

在资源不确定条件下, 式(1)和式(2)所示目标值需要通过对任务实际工期的先验知识并由仿真获得。它依赖于对有关干扰的仿真数据和统计知识, 但这种统计知识很难得到, 通常采用蒙特卡罗仿真或近似算法来近似, 这一过程非常复杂

和耗时。为避免在求解和优化过程中对中间目标进行仿真,本文设计两种不需要太大计算代价的函数 f'_1 和 f'_2 来替代问题模型中的目标函数 f_1 和 f_2 。

替代目标函数 f'_1 和 f'_2 定义如下

$$f'_1 = \min st_J \quad (8)$$

$$f'_2 = \max \left(\sum_{i=2}^{J-1} NS_i \cdot \sum_{n=1}^N \lambda_{in} \sum_{x=1}^{SR_{in}} e^{-x} \right) \quad (9)$$

式(8)表示最小化任务完成时间,式中 st_J 表示完成虚任务的开始时间,即所有任务的完成时间。式(9)表示最大化任务资源缓冲区的加权和,式中 NS_i 表示任务 t_i 及其所有后续任务的不稳定权重之和,即 $NS_i = \omega_i + \sum_{t_j \in Succ_i} \omega_j$; ρ_{in} 为辅助变量,当 t_i 对资源类型 r_n ($n \in N$)的需求 $trr_{in} > 0$ 时, $\lambda_{in} = 1$,否则 $\lambda_{in} = 0$; SR_{in} 表示 t_i 分配得到的资源类型 r_n 的冗余量,即 $SR_{in} = \sum_{k=1}^K \alpha_{ki} \cdot prc_{kn} - trr_{in}$ 。式(9)隐含着如果资源缓冲区中已经包含了某类资源,则继续增加该类资源的必要性降低。

2 基于资源缓冲区的军事任务计划预测调度算法

该算法基于平台有效资源功能向量进行任务—平台分配,平台有效资源功能向量是一种取值介于实际功能向量与期望功能向量之间,大小可以调节的资源功能向量。该算法的基本思想是:首先计算平台有效资源功能向量,确定资源冗余分配方案;然后进行任务—平台分配,计算任务执行模式;最后,利用多目标进化算法对问题进行多次迭代求解,生成问题的 Pareto 最优解集。

算法具体步骤描述如下:

步骤 1:初始化调节参数,计算平台有效资源功能向量;

步骤 2:根据平台有效资源功能向量,进行任务—平台分配,计算每个任务可用执行模式集;

步骤 3:对军事任务计划方案进行多目标求解,计算给定平台有效资源功能向量对应的 Pareto 解集;

步骤 4:对多次迭代获得所有 Pareto 解集进行非支配排序,获得 Pareto 最优解集;

步骤 5:输出问题解。

2.1 计算平台有效功能向量

平台有效功能向量是指资源不确定条件下,平台在任意时刻的实际有效数量。假设资源的故障时间和修复时间服从指数分布,则资源在任意时刻的有效数量将服从符合历史经验的泊松分

布。利用数学知识,根据平台资源的平均故障时间和平均修复时间,能够容易计算得到平台资源的有效数量。

令 X_{kn} 和 Y_{kn} 分别为平台 p_k 中资源 r_n 的故障时间和修复时间, MTF_{kn} 、 MTR_{kn} 分别为 X_{kn} 和 Y_{kn} 的期望平均值,即 $E(X_{kn}) = MTF_{kn}$, $E(Y_{kn}) = MTR_{kn}$ 。则平台 p_k 中资源的 r_n 单位有效概率为

$$A_{kn} = \frac{E(X_{kn})}{E(X_{kn}) + E(Y_{kn})} = \frac{MTF_{kn}}{MTF_{kn} + MTR_{kn}} \quad (10)$$

令 $\rho_{kn} = \rho \cdot \frac{MTR_{kn}}{MTF_{kn}}$,其中 ρ ($0 \leq \rho \leq 1$)为资源冗余调节参数,

$$E_{kn} = \frac{1}{1 + \rho_{kn}} \quad (11)$$

利用式(11),平台 p_k 载有的 prc_{kn} 个资源 r_n 中,有效数量为 j 的概率为

$$\begin{aligned} Pr(rc_{kn} = j) &= \binom{prc_{kn}}{j} \cdot (A_{kn})^j \cdot (1 - A_{kn})^{(prc_{kn}-j)} \\ &= \binom{prc_{kn}}{j} \frac{\varphi_{kn}^{prc_{kn}-j}}{(1 + \varphi_{kn})^{prc_{kn}}} \end{aligned} \quad (12)$$

利用式(12),可计算平台 p_k 中资源类型 r_n 的实际有效数量

$$E(prc_{kn}) = \sum_{i=0}^{prc_{kn}} i \cdot Pr(rc_{kn} = j) \quad (13)$$

计算平台 p_k 中每类资源的有效数量,即可得到 p_k 的有效功能向量

$$E(Prc_k) = \{E(prc_{k1}), E(prc_{k2}), \dots, E(prc_{kN})\} \quad (14)$$

根据定义可知,资源冗余调节参数 $\rho = 1$ 时, $E_{kn} = A_{kn}$,平台有效资源功能向量等于平台期望资源功能向量;当 $\rho = 0$ 时, $E_{kn} = 1$,平台有效资源功能向量等于平台资源功能向量;可见,平台有效功能向量是平台载有效资源数量的一般表达,平台资源功能向量和平台期望资源功能向量是它的两种特殊形式。

2.2 任务—平台分配算法

任务与平台通过任务执行资源需求向量和平台资源功能向量相关联,进行任务—平台分配,从而得到满足任务执行资源约束的平台组合。满足任务执行资源约束的一种平台组合称为该任务的一种执行模式。由于平台功能的多样性,一种任务可能有多种执行模式,一个平台可能包含在多个任务执行模式中。考虑到平台数量的有限性以及平台之间协同的复杂性,人们总是希望用尽量少的平台去执行尽量多的任务^[7]。因此,任务执行模式必须满足如下两个条件:1) 任务执行模式

包含的平台集合必须满足任务的执行资源需求;
2) 任务执行模式中减少任一个平台后,该模式将不再满足任务的执行资源需求。

给定平台有效功能向量,任务—平台分配的具体步骤如下:

步骤1:令 $j = 2$;

步骤2:令 $l = 1, Mode_j = \emptyset$;

步骤3:计算由 l 个平台组成的任务 t_j 的执行模式集合 FM_l ;

步骤4: $Mode_j = Mode_j \cup FM_l$, 计算任务 t_j 可用执行模式数量 $|Mode_j|$, 如果 $|Mode_j| \geq N_{mode}$, 则转步骤6, 否则转步骤5, 其中, N_{mode} 是预定义常数, 用来定义每个任务的最少执行模式数量;

步骤5: $l = l + 1$, 如果 $l \leq k$ (k 为平台总数), 转步骤3, 否则转步骤6;

步骤6: $j = j + 1$, 如果 $j \leq J - 1$, 转步骤2, 否则, 转步骤7;

步骤7:输出任务执行模式集。

基于平台有效功能向量进行任务—平台分配的过程中,如果资源冗余调节参数 ρ 很大,有可能出现这样一种情况:所有平台的有效资源功能向量之和无法满足某个任务的执行资源需求,有

$$\sum_{k=1}^K E(prc_{kn^*}) < trr_{i^*n^*} \quad (15)$$

此时,需要对部分平台的 $E(prc_{kn^*})$ 进行调整,通过增加资源类型 r_{n^*} 的有效数量来满足任务 t_{i^*} 的执行资源需求。具体调整过程为:首先找出对于资源 r_{n^*} 平台有效数量与实际数量差值最大的平台 p_{k^*} ;然后将该平台中资源 r_{n^*} 的有效数量增加一个单元,即 $E(prc_{k^*n^*}) = E(prc_{k^*n^*}) + 1$;重复上述两个步骤,直到所有平台的有效功能向量之和满足任务 t_{i^*} 的执行资源需求,有

$$\sum_{k=1}^K E(prc_{kn^*}) = trr_{i^*n^*} \quad (16)$$

2.3 军事任务计划多目标求解算法

根据前文问题模型描述可知,军事任务计划是一类典型的多目标优化问题,因此本文选择多目标优化领域最为优秀的 NSGA II^[8] 进化算法对问题进行求解。结合军事任务计划问题特点,本文设计了新的问题编码方式和交叉、变异算子。

(1) 问题编码

针对本文问题特点,采用了任务列表编码方式对问题进行编码,每个染色体由任务 ID 序列号及其执行模式组成。一个完整任务计划方案染色体编码可描述为

$$X = \{(t_1, mode_1), (t_2, mode_2), \dots, (t_j, mode_j)\}$$

式中 t_i 表示任务 ID, $mode_i$ 表示 t_i 的执行平台组合。

(2) 初始解集

本文采用串行进度生成机制、随机任务优先规则、随机模式优先规则生成 P 个初始可行解。迭代过程为:从任务 t_1 开始,每步迭代,在满足执行条件的候选任务集中随机选择一个任务,并为该任务随机选择一种执行模式,使其最早开始,直到最后一个任务完成,生成一个初始可行解。

(3) 交叉算子

根据问题特点,本文选择优先基因交叉算子 (Precedence Set Crossover, PSC)^[9] 对选中的父代染色体进行交叉操作。交叉操作只改变编码中任务执行顺序,不改变任务执行模式,其主要过程为:1) 随机选择具有时序约束的多个优先基因;2) 确定优先基因在两个父代染色体(父亲和母亲)中出现的先后次序;3) 在保持优先基因在父亲(或母亲)染色体中先后顺序的前提下,按照母亲(或父亲)染色体中的基因序列生成儿子(或女儿)染色体。

(4) 变异算子。

本文选择单点变异算子对染色体进行变异操作,变异操作只改变任务的执行模式,不改变任务执行顺序。变异过程首先随机选择一个变异基因,然后改变该基因位置任务的执行模式。

3 仿真算例

3.1 实验想定

为验证本文方法可行性和有效性,设计如下案例进行仿真实验(案例数据来源于 A2C2 实验七^[10] 的公开数据)。已知某联合登陆战役给定任务图如图 1 所示,其中,任务 t_1 和 t_{20} 都为虚任务,分别表示任务计划的开始和结束。所有任务的属性描述如表 1 所示。给定任务计划的可用平台集合及其属性描述如表 2 所示。算例在 Intel Core2 Duo CPU 2.33 GHz + 1GB 内存的个人计算机上以 Matlab (R2009a) 语言编程实现。

3.2 实验参数设计

为简化计算,案例假设所有平台中同类资源的故障时间和修复时间服从独立并且相同的指数分布。资源的平均故障时间服从 $0.5T_{\min}$ 至 $1.5T_{\min}$ 之间的均匀分布, T_{\min} 为不考虑资源随机失效,即资源冗余调节参数 $\rho = 0$ 时求得的任务计划最短完成时间;资源平均修复时间服从 1 至 5 之间的均匀分布。非虚任务的不稳定权重 ω_i 服从 1 到 10 之

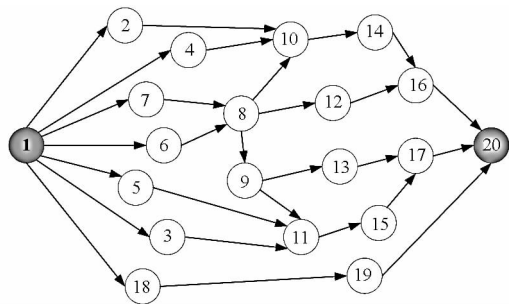


图 1 任务图

Fig. 1 Task graph of the test scenario

间的三角形分布,可知 ω_i 在 1 到 10 之间取值的概率为 $P(\omega_i = x) = 0.21 - 0.02 \cdot x$,任务 t_j 的不稳定权重通常设为非虚任务均值的 β 倍,即 $\omega_j = \beta \cdot \bar{\omega}$,令 $\beta = 10$,已知 $\bar{\omega} = 3.85$,因此 $\omega_j = 38.5$ 。任务完成期限 δ_j 与 T_{\min} 有关,人们通常不希望 δ_j 偏离 T_{\min} 太大,因此定义 $\delta_j = (1 + \lambda) \cdot T_{\min}$, λ 取值由决策者根据任务完成时间和任务延迟成本的折衷权衡而定。

算法中涉及的各项参数取值如下:种群规模 $P = 100$,最大进化代数 $G = 2000$,交叉概率取值为 1,变异概率取值 $0.2 \sim 0.3$,当种群连续 50 代没有更新时,算法终止。

3.3 实验结果分析

根据前文 2.1 节所述,资源冗余调节参数 ρ 用于调节方案中的平台资源冗余量, $0 \leq \rho \leq 1$, ρ 取值越大,表示方案中的资源缓冲区越大。令 ρ 取值分别为 $\{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$ 进行实验,统计结果如图 2 所示。为方便比较,图中仅描述了 $\rho = 0$ 、 $\rho = 1$ 时得到的 Pareto 最优前沿 $PF_{\rho=0}$ 、 $PF_{\rho=1}$,以及 ρ 取所有值 $(0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1)$ 时获得 Pareto 最优前沿 PF_{get} 。

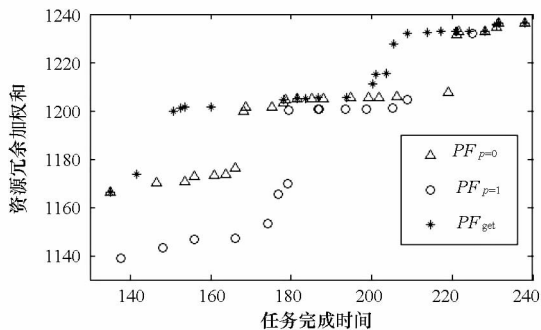


图 2 Pareto 最优解集

Fig. 2 Pareto fronts for the different ρ values

表 1 任务属性

Tab. 1 Task attributes description

任务 ID	时间需求	资源需求								位置坐标
		r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	
1	0	0	0	0	0	0	0	0	0	-
2	30	5	3	10	0	0	8	0	6	70,15
3	30	5	3	10	0	0	8	0	6	64,75
4	10	0	3	0	0	0	0	0	0	15,40
5	10	0	3	0	0	0	0	0	0	30,95
6	10	0	3	0	0	0	0	10	0	28,73
7	10	0	0	0	10	14	12	0	0	24,60
8	10	0	0	0	10	14	12	0	0	28,73
9	10	0	0	0	10	14	12	0	0	28,83
10	10	5	0	0	0	0	5	0	0	28,73
11	10	5	0	0	0	0	5	0	0	28,83
12	10	0	0	0	0	0	10	5	0	25,45
13	10	0	0	0	0	0	10	5	0	5,95
14	20	0	0	0	0	0	8	0	6	25,45
15	20	0	0	0	0	0	8	0	6	5,95
16	15	0	0	0	20	10	4	0	0	25,45
17	15	0	0	0	20	10	4	0	0	5,95
18	10	0	0	0	0	0	8	0	4	5,60
19	20	0	0	0	8	6	0	4	10	5,60
20	0	0	0	0	0	0	0	0	0	-

表 2 平台属性

Tab. 2 Platform attributes description

平台 ID	速度	平台资源							
		r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8
1	2	10	10	1	0	9	5	0	0
2	2	1	4	10	0	4	3	0	0
3	2	10	10	1	0	9	2	0	0
4	4	0	0	0	2	0	0	5	0
5	1.35	1	0	0	10	2	2	1	0
6	4	5	0	0	0	0	0	0	0
7	4	3	4	0	0	6	10	1	0
8	4	1	3	0	0	10	8	1	0
9	4	1	3	0	0	10	8	1	0
10	4	1	3	0	0	10	8	1	0
11	4.5	6	1	0	0	1	1	0	0
12	4.5	6	1	0	0	1	1	0	0
13	4.5	6	1	0	0	1	1	0	0
14	2	0	0	0	0	0	0	10	0
15	5	0	0	0	0	0	0	0	6
16	7	0	0	0	0	0	0	0	6
17	2.5	0	0	0	6	6	0	1	10
18	1.35	1	0	0	10	2	2	1	0
19	1.35	1	0	0	10	2	2	1	0
20	1.35	1	0	0	10	2	2	1	0

在图 2 中,对 $PF_{\rho=0}$ 、 $PF_{\rho=1}$ 和 PF_{get} 的加权资源冗余数量和任务完成时间目标值进行比较,很显然 PF_{get} 能够获得更好的 Pareto 最优前沿,证明了本文方法的优越性。

下面分析 ρ 取值对实验结果——Pareto 最优解集的影响。为比较 ρ 不同取值对应 Pareto 解集的优劣, 首先引入世代距离 (Generational Distance, GD)^[11] 的概念。GD 是用来表示两个 Pareto 前沿之间接近程度的概念。假设已知某一 ρ 值对应的最优解集为 PF_ρ , 多次实验 (ρ 取值不同) 获得的最优解集为 PF_{get} , 则 PF_ρ 与 PF_{get} 之间的 GD 能够在某种程度上度量 PF_ρ 的优劣, 进而可用来评价 ρ 值对 Pareto 解集的影响。

GD 定义如式(17)所示。

$$GD = \frac{1}{n} \left(\sum_{i=1}^n dist_i^2 \right)^{1/2} \quad (17)$$

式中 $dist_i$ 表示 PF_ρ 中第 i 个解与 PF_{get} 中最近解之

间的欧氏距离, n 为 PF_ρ 中解的个数。

根据上节所述, $\delta_j = (1 + \lambda) \cdot T_{min}$, 因此, 令 λ 取不同值, 可计算不同完成期限约束下, ρ 不同取值对应 Pareto 解与多次实验获得的最优解之间的 GD, 计算结果如表 3 所示。

对表 3 中实验数据进行分析, 表中每行数据表示 λ 值给定 (即给定 δ_j), 不同 ρ 值对应的 Pareto 解集与所得最优解 PF_{get} 之间的 GD。比较表中各行数据可知, 随着 λ 取值增大, GD 变小, 这说明在 Pareto 解集 PF_ρ 中, 完成时间越长的解越接近最优解。图 2 中数据验证了这一结论: 图中 $PF_{\rho=0}$ 和 $PF_{\rho=1}$ 的解 f_1 取值越大越接近 PF_{get} 。

表 3 不同 ρ 对应的 Pareto 解与最优解之间的 GD

Tab. 3 GD between the Pareto fronts

λ/φ	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	平均值
0.2	5.55	8.10	3.77	15.52	1.86	9.07	3.20	14.97	8.37	11.92	15.94	8.93
0.4	3.00	6.27	3.51	3.69	1.39	6.49	2.60	8.52	4.21	6.51	8.84	5.46
0.6	2.44	3.56	2.84	7.73	1.15	4.66	1.66	5.75	3.65	5.09	6.59	4.10
0.8	1.90	2.82	2.35	6.62	0.92	4.25	1.49	5.37	3.19	4.35	6.18	3.59
平均值	3.22	5.19	3.12	9.64	1.33	6.12	2.24	8.65	4.86	6.97	9.39	

在表 3 中, 每列数据表示 ρ 取值给定, 不同 λ 值对应的 Pareto 解集与所得最优解 PF_{get} 之间的 GD。分析表 3 中各列数据可以发现: (1) GD 与 ρ 值之间并不存在单调的关联关系, 因此“资源冗余越多, 方案越鲁棒”的直观判断并不正确, 图 2 中数据验证了这一判断, 如 $PF_{\rho=0}$ 与 $PF_{\rho=1}$ 相比, 更加接近 PF_{get} ; (2) 给定实验想定和实验参数, 存在一个令 GD 最小, 即取值最接近最优解 PF_{get} 的 ρ 值, 如表 3 中 ρ 最优取值为 0.4。

4 结 论

平台资源不确定性是军事任务计划中普遍存在的一类典型不确定因素。本文基于资源缓冲区的冗余技术提出了资源不确定军事任务计划预测调度算法, 该算法以方案稳定性为优化目标, 基于 NSGA-II 算法框架对问题进行多目标求解。最后通过仿真算例对本文算法进行验证, 实验结果证明, 本文算法能够对资源不确定军事任务计划进行有效求解。

参考文献 (References)

[1] 刘忠, 张维明, 阳东升, 等. 作战计划系统技术[M]. 北京: 国防工业出版社, 2007.
LIU Zhong, ZHANG Weiming, YANG Dongsheng, et al. Military mission planning system technology [M]. Beijing: National Defense Industry Press, 2007. (in Chinese)

[2] Levchuk G M, Levchuk Y N, Luo J, et al. Normative design

of organizations. I. Mission planning[J]. IEEE Transactions on Systems, Man and Cybernetics Part A: Systems and Humans, 2002, 32(3): 346-359.

[3] Belfares L, Klihib W, Lo N, et al. Multi-objectives tabu search based algorithm for progressive resource allocation [J]. European Journal of Operational Research, 2007, 177(3): 1779-1799.

[4] Lambrechts O, Demeulemeester E, Herroelen W. Time slack-based techniques for robust project scheduling subject to resource uncertainty [J]. Annals of Operations Research, 2011, 186(1): 443-464.

[5] Lambrechts O, Demeulemeester E, Herroelen W. Proactive and reactive strategies for resource-constrained project scheduling with uncertain resource availabilities[J]. Journal of Scheduling, 2008, 11(2): 121-136.

[6] Van de Vonder S, Demeulemeester E, Herroelen W. A classification of predictive-reactive project scheduling procedures [J]. Journal of Scheduling, 2007, 10(3): 195-207.

[7] 张维明, 刘忠, 阳东升, 等. 体系工程理论与方法[M]. 北京: 科学出版社, 2010.
ZHANG Weiming, LIU Zhong, YANG Dongsheng, et al. System of system engineering theory and methods [M]. Beijing: Science Press, 2010. (in Chinese)

[8] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182-197.

[9] Alcaraz J, Maroto C. A robust genetic algorithm for resource allocation in project scheduling [J]. Annals of Operations Research, 2001, 102(1): 83-109.

[10] Kleinman D, Young P, Higgins G. The DDD-III: A tool for empirical research in adaptive organizations [C]// Proceedings on Command and Control Research and Technology Symposium, 1996: 827-836.

[11] Veldhuizen V. On measuring multiobjective evolutionary algorithm performance [C]// Congress on evolutionary computation, 2000: 204-211.