

装备精确保障任务规划建模与混沌蝙蝠算法求解

王坚浩^{1†}, 张亮¹, 史超¹, 车飞¹, 武杰², 李超¹

(1. 空军工程大学 装备管理与无人机工程学院, 西安 710051; 2. 中国人民解放军 94402 部队, 济南 250002)

摘要: 针对装备精确保障任务规划中任务时序逻辑约束和资源占用冲突等问题, 建立以时效优先为目标的数学模型, 提出基于多维动态列表规划和混沌蝙蝠算法的混合任务规划方法. 通过多维动态列表规划选择处理的任务, 设计具有自适应搜索策略和变异操作的离散混沌蝙蝠算法, 为选定的任务分配资源. 全局搜索中自适应调整惯性权重和学习因子以达到探索与开发能力的最佳平衡, 局部搜索中采用混沌变异操作以协助种群跳出局部最优. 仿真算例表明, 所提出算法具有较快的收敛速度和较高的求解精度.

关键词: 装备精确保障; 任务规划; 多维动态列表规划; 自适应搜索; 变异; 离散混沌蝙蝠算法

中图分类号: TP18; TP301

文献标志码: A

Task scheduling modeling and chaotic bat algorithm solving method of equipment efficient support

WANG Jian-hao^{1†}, ZHANG Liang¹, SHI Chao¹, CHE Fei¹, WU Jie², LI Chao¹

(1. Equipment Management and Unmanned Aerial Vehicles Engineering College, Air Force Engineering University, Xi'an 710051, China; 2. PLA 94402 Troop, Ji'nan 250002, China)

Abstract: For the problems of task sequential logic constraints and resource occupancy conflicts among equipment efficient support task scheduling, a mathematical model in pursuit of the priority of task implementation time is established, a hybrid task scheduling method based on multi-dimensional dynamic list scheduling(MDLS) and the chaotic bat algorithm is proposed. In the proposed method, the task to be disposed is selected by MDLS, then the discrete chaotic bat algorithm(DCBA) with the adaptive searching strategy and mutation operator is designed to allocate the resource to the selected task. Inertia weight and acceleration coefficients are adjusted adaptively in global search to coordinate the exploration and exploitation ability, and the chaotic mutation operator is adopted in local search to help the swarm jump out from local optimum. The simulation example illustrates that the proposed method has better performance in convergence speed and solving precision.

Keywords: equipment efficient support; task scheduling; multi-dimensional dynamic list scheduling; adaptive searching; mutation; discrete chaotic bat algorithm

0 引言

针对装备精确保障^[1]任务规划中保障任务不确定性、保障任务时序逻辑关系约束和保障资源占用冲突等实际问题, 如何合理地处理保障任务与保障资源之间的分配问题是装备精确保障任务规划需要重点考虑的问题. 装备精确保障任务规划需要将保障资源合理地分配到存在时序逻辑关系约束的任务中, 这属于时间扩展型任务分配, 是 NP-Hard 问题. 对于此类问题, 通常采用致力于寻找最优解的整数规划、分支界定法和枚举法等精确求解方法, 但在面向规模

较大、约束复杂和实时性较强的任务规划问题时, 无法在有效时间内实现优化求解. 为此, 人们提出了基于动态列表规划(DLS)的任务规划方法^[2-3]. DLS方法具有求解效率高、可获得最优解或满意解的优点, 但由于采用基于贪心策略的局部搜索策略, 无法保证以稳定的高概率寻找最优任务分配方案.

近年来, 具有生物进化思想和动物行为群体的群体智能优化算法引起了众多学者的关注, 如较为典型的遗传算法(GA)和粒子群优化算法(PSO). 在此基础上, 一些学者将DLS与GA、PSO等群体智能优化

收稿日期: 2017-05-04; 修回日期: 2017-06-26.

基金项目: 国家自然科学基金项目(71601183).

责任编委: 夏元清.

作者简介: 王坚浩(1982-), 男, 讲师, 博士, 从事装备保障任务规划、资源调度的研究; 张亮(1982-), 男, 副教授, 博士, 从事装备综合保障工程学研究.

†通讯作者. E-mail: hamilton-wang@sina.com

算法相结合,发表了一系列研究成果.例如:文献[4]针对任务-资源分配模型,提出了基于DLS和GA的任务分配策略,使用全局搜索GA为选定的任务分配最佳的平台资源,提高了任务分配方案趋近于最优的概率,但传统GA寻优时收敛速度慢,任务分配耗时较长;文献[5]利用量子遗传算法(QGA)在寻优速度和全局探索能力上的优势,提出了基于DLS和QGA的任务-资源分配方法;文献[6]在任务完成时间限制下建立任务-资源分配模型,提出了基于优先排序和离散粒子群优化(DPSO)的装备保障任务规划方法,但传统PSO算法存在早熟收敛问题,导致算法全局探索能力较弱,且提出的DPSO仅是对传统PSO算法进行了简单的四舍五入离散化处理.

蝙蝠算法(BA)^[7]是继PSO和GA优化算法后提出的又一种群体智能优化算法.已有研究表明^[8-9],BA优化算法将GA、PSO及和声搜索(HS)^[10]算法进行了有机结合,并且PSO和HS优化算法可以认为是BA优化算法在经过适当简化后的一种特殊情况,相比于GA和PSO,BA优化算法具有更快的收敛速度和更好的全局寻优能力,因此,BA优化算法一经提出便受到众多学者的关注和研究.但是,标准BA优化算法同样存在易于陷入局部最优、算法后期收敛速度较慢、收敛精度不高等问题.

本文围绕装备精确保障任务-资源分配问题,首先建立以时效优先为目标的任务规划数学模型,设计以多维动态列表规划(MDLS)算法为主的求解框架,嵌套具有自适应搜索策略和变异操作的离散混沌蝙蝠算法(DCBA)为选定任务分配资源的任务分配策略,通过自适应调整惯性权重和学习因子以达到探索与开发能力的最佳平衡,并采用混沌变异操作协助种群跳出局部最优.案例仿真表明,该算法可以实现装备精确保障任务规划和资源调度的科学性和有效性.

1 装备精确保障任务规划问题描述

1.1 基本概念

保障任务(Task, T):根据保障任务的性质、保障装备的类型以及保障地点,对保障任务进行分解,得到所需要执行的任务序列.对保障任务分解的目的是保证分解后的任务能够由某项资源独立完成,不需要其他资源参与.记包含 N 项任务的保障任务集为 $T = \{T_1, T_2, \dots, T_N\}$,集合 T 中各任务之间具有顺序依赖和执行约束关系.对于 $\forall T_i \in T$,其属性包括:任务开始时间 S_{T_i} ;任务持续时间 C_{T_i} ;任务优先级 P_{T_i} ;保障任务集 T 中任务间的时序约束关系 $C^{T-T} = (c_{ij}^{T-T})_{N \times N}$,若任务 T_i 是任务 T_j 的前导任

务,则 $c_{ij}^{T-T} = 1$,否则 $c_{ij}^{T-T} = 0$.

保障资源(Resource, R):将保障资源按照保障装备类型、保障单位及保障性质进行分解,得到可匹配任务的资源.分解后的资源均具有单独承担某项或某几项任务的能力,且均能同时被占用.记包含 M 项资源的保障资源集为 $R = \{R_1, R_2, \dots, R_M\}$.任务占用不同资源,其执行时间是不同的.记任务与资源匹配的相应执行时间矩阵为 $RT = (rt_{mi})_{M \times N}$, rt_{mi} 表示资源 R_m 匹配任务 T_i 的执行时间,当 $rt_{mi} = 0$ 时表示对应的资源与任务不可匹配.

保障任务规划,即任务-资源分配方案,可由矩阵 $D^{R-T} = (d_{mi}^{R-T})_{M \times N}$ 表示,其中若资源 R_m 分配给任务 T_i 使用,则 $d_{mi}^{R-T} = 1$,否则 $d_{mi}^{R-T} = 0$.当一个类型的资源同时分配给多项任务使用时,该资源将按照任务的优先级大小和任务执行的时序逻辑顺序依次分配给各项任务.保障任务集 T 中任务优先级可采用关键路径法、层次分配法、加权长度法等确定^[2].

1.2 任务规划目标函数

对于 $\forall T_i \in T$,其结束时间记为 $E_{T_i} = B_{T_i} + C_{T_i}$,其前导任务构成的集合记为 $S_{T_i}^T = \{T_k | c_{ki}^{T-T} = 1, 1 \leq k \leq N\}$,处理任务 T_i 的资源构成的集合记为 $S_{T_i}^R = \{R_m | d_{mi}^{R-T} = 1, 1 \leq m \leq M\}$.对于 $\forall R_m \in R$,资源发生占用冲突的任务集合,即占用资源 R_m 的任务构成的集合记为 $S_{R_m}^T = \{T_i | d_{mi}^{R-T} = 1, 1 \leq i \leq N\}$.

当资源按照任务-资源分配方案分配给各项任务时,其任务执行时间(TIT)即为所有任务的最终结束时间,有

$$\text{TIT} = \max_{T_i \in T} (E_{T_i}). \quad (1)$$

装备精确保障任务规划考虑以最短保障时间为目标,因此任务规划模型的目标函数为

$$\min \text{TIT} = \min (\max_{T_i \in T} (E_{T_i})). \quad (2)$$

1.3 任务规划约束分析

装备精确保障任务规划问题的约束条件包括以下3类:1)任务-资源分配约束;2)任务执行时序逻辑约束;3)任务执行时间约束.

1) 任务-资源分配约束.

通过对任务分解保证任务可由某项资源独立完成,即

$$\sum_{m=1}^M d_{mi}^{R-T} = 1, d_{mi}^{R-T} = \{0, 1\}. \quad (3)$$

2) 任务执行时序逻辑约束.

当任务 T_i 和 T_j 同时占用资源 R_m 时,任务 T_i 和 T_j 执行时序满足

$$[B_{T_i}, E_{T_i}] \cap [B_{T_j}, E_{T_j}] = \emptyset, \quad \forall T_i, T_j \in S_{R_m}^T, i \neq j. \quad (4)$$

为了减少资源处于空闲状态的时间,对于资源占用冲突的任务 T_i 和 T_j ,假设任务 T_i 的优先级 P_{T_i} 高于任务 T_j 的优先级 P_{T_j} ,则满足

$$E_{T_i} = B_{T_j}, \forall T_i, T_j \in S_{R_m}^T, i \neq j, P_{T_i} \geq P_{T_j}. \quad (5)$$

对于带有顺序关系的任务 T_i 和 T_j ,假设任务 T_i 为 T_j 的前导任务,则任务 T_i 开始时间满足

$$E_{T_i} \geq B_{T_j}, \forall c_{ij} = 1, i \neq j. \quad (6)$$

3) 任务执行时间约束.

保障任务执行时间 TIT 不能大于保障任务完成时限 (TCL),且任务 T_i 执行时间不能大于任务 T_i 完成时限,即

$$TIT = \max_{T_i \in T} (E_{T_i}) \leq TCL_{tot}, \quad (7)$$

$$C_{T_i} = E_{T_i} - B_{T_i} \leq TCL_{T_i}. \quad (8)$$

1.4 任务规划数学模型

以保障任务执行时间 TIT 值最小化为目标,综合考虑任务-资源分配过程的约束条件,装备精确保障任务规划问题的数学模型描述如下:

$$\begin{aligned} \min TIT &= \min(\max_{T_i \in T} (E_{T_i})). \\ \text{s.t.} \quad &\sum_{m=1}^M d_{mi}^{R-T} = 1, d_{mi}^{R-T} = \{0, 1\}; \\ &[B_{T_i}, E_{T_i}] \cap [B_{T_j}, E_{T_j}] = \emptyset, \\ &\forall T_i, T_j \in S_{R_m}^T, i \neq j; \\ &E_{T_i} = B_{T_j}, \forall T_i, T_j \in S_{R_m}^T, i \neq j, P_{T_i} \geq P_{T_j}; \\ &E_{T_i} \geq B_{T_j}, \forall c_{ij} = 1, i \neq j; \\ &TIT = \max_{T_i \in T} (E_{T_i}) \leq TCL_{tot}; \\ &C_{T_i} = E_{T_i} - B_{T_i} \leq TCL_{T_i}. \end{aligned} \quad (9)$$

2 基于MDLS-DCBA的任务规划算法求解

2.1 算法流程

基于 MDLS-DCBA 的混合任务分配算法流程如图 1 所示.该算法主要包含两个关键环节:1)使用 MDLS 在待选保障任务集中选择一项当前需要处理的任任务;2)设计具有自适应搜索策略和变异操作的 DCBA 为选定的任务分配相应资源.

2.2 基于MDLS的任务选择

采用 MDLS 选择任务的步骤如下.

Step 1: 定义过程变量:可处理任务集 T_{ready} 、已处理任务集 $T_{complete}$ 和空闲资源集 P_{free} ;

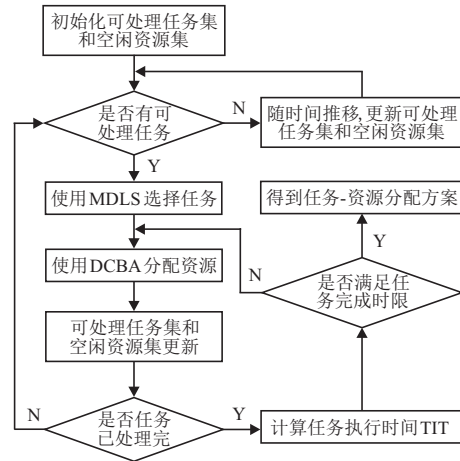


图 1 基于MDLS-DCBA的任务分配算法流程

Step 2: 初始化所有任务的优先级 P_{T_i} ;

Step 3: 更新 $T_{complete}$ 任务执行时间,若 $T_{complete} = \emptyset$,则跳过;

Step 4: 在前导任务已完成的任务中,选择优先级高的一项任务;

Step 5: 为该任务在空闲资源集 P_{free} 中选择一项资源以满足任务完成时限;

Step 6: 判断是否所有任务已完成分配,如果是则终止算法,否则转入 Step 2.

其中当任务 T_i 的所有直接前导任务都已处理完成时,该任务 T_i 便进入可处理任务集 T_{ready} ,在 T_{ready} 中根据任务优先级 P_{T_i} 依次处理任务,即任务优先级高的任务优先占用资源.

2.3 基于DCBA的资源分配

2.3.1 标准BA

BA 优化算法模拟蝙蝠使用回声定位进行捕食的行为:当蝙蝠寻找猎物时,发出的超声波脉冲响度为最大,利于超声波远距离传播;当发现猎物后,脉冲响度逐渐减弱而脉冲频率逐渐增大,利于精确定位猎物.在标准 BA 优化算法中,对于种群大小为 n 的蝙蝠群体,蝙蝠个体速度和位置更新如下:

$$f_i = f_{min} + (f_{max} - f_{min})\beta, \quad (10)$$

$$v_i^{t+1} = v_i^t + (x_i^t - x_{gbest})f_i, \quad (11)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1}. \quad (12)$$

其中: x_i^t 、 v_i^t ($i = 1, 2, \dots, n$) 分别为第 i 只蝙蝠在第 t 次进化时的位置和速度, f_i 、 f_{min} 和 f_{max} 分别为第 i 只蝙蝠当前时刻的脉冲频率、脉冲频率的最小值和最大值, $\beta \in [0, 1]$ 为一个服从均匀分布的随机数, x_{gbest} 为当前全局最优解.

从中随机选择一只蝙蝠,并更新该蝙蝠相应的位置,即在被选择解的附近产生一个新解,该过程可被

理解为局部搜索过程,有

$$\mathbf{x}_{\text{new}}^t = \mathbf{x}_{\text{old}}^t + \varepsilon A_{\text{ave}}^t. \quad (13)$$

其中: $\mathbf{x}_{\text{old}}^t$ 为当前全局最优解集中随机选择的一个解, $\varepsilon \in [-1, 1]$ 为一个服从均匀分布的随机数, A_{ave}^t 为当前代前*i*只蝙蝠的平均响度.

蝙蝠在实际捕猎过程中,其脉冲响度*A*随着与猎物距离的减小而逐渐减弱,脉冲发生率*r*随着与猎物距离的减小而逐渐增大.因此,在每次进化过程中,脉冲响度 A_i^t 和脉冲发生率 r_i^t 更新如下:

$$A_i^{t+1} = \alpha A_i^t, \quad (14)$$

$$r_i^{t+1} = r_i^0 [1 - \exp(-\gamma t)]. \quad (15)$$

其中: $0 < \alpha < 1$ 和 $\gamma > 0$ 为常量,类似模拟退火算法中冷却过程的冷却因子,一般情况下,取 $\alpha = \gamma$.不难发现,当 $t \rightarrow \infty$ 时, $A_i^t \rightarrow 0$, $r_i^t \rightarrow r_i^0$.

2.3.2 具有自适应搜索策略和变异操作的DCBA

从当前空闲资源集 P_{free} 中为选定的任务分配资源,其实质是多维0-1背包问题.标准BA优化算法主要用于求解连续空间优化问题,对于此类离散型空间优化问题的求解优势并不明显.受二进制粒子群算法(BPSO)^[11-12]等离散群体智能优化算法在0-1背包问题解决上具有显著优势的启发,Mirjalili等^[13]提出了二进制蝙蝠算法(BBA).标准BBA优化算法为实现BA优化算法向离散域的拓展,引入如下连续域向二进制离散域的映射公式:

$$V(v_{id}^t) = \left\lfloor \frac{2}{\pi} \arctan \left(\frac{\pi}{2} v_{id}^t \right) \right\rfloor. \quad (16)$$

$$x_{id}^{t+1} = \begin{cases} (x_{id}^t)^{-1}, & \text{rand} < V(v_{id}^{t+1}); \\ x_{id}^t, & \text{rand} \geq V(v_{id}^{t+1}). \end{cases} \quad (17)$$

其中: $i = 1, 2, \dots, n$; $d \in D$.

标准BBA优化算法虽然实现了BA优化算法由连续域向离散域的拓展,但仍然存在易陷入局部最优、算法后期收敛速度较慢、收敛精度不高等问题.对此,本文提出一种具有自适应搜索策略和变异操作的DCBA优化算法,算法流程如图2所示.

与标准BBA相比,本文对图2中的阴影部分进行改进,包括以下3点:1)解的编码和约束处理;2)结合混沌搜索和自适应搜索策略更新脉冲频率和蝙蝠个体速度;3)根据脉冲发生率和脉冲响度选择局部最优解.

1)解的编码和约束处理.

任务-资源分配矩阵 $D^{R-T} = (d_{mi}^{R-T})_{M \times N}$ 即为求解对象,直接对矩阵 D^{R-T} 进行编码,则解的维数 $D = M \times N$,因此蝙蝠个体位置 $\mathbf{x}_i = [x_{i1}, \dots, x_{id},$

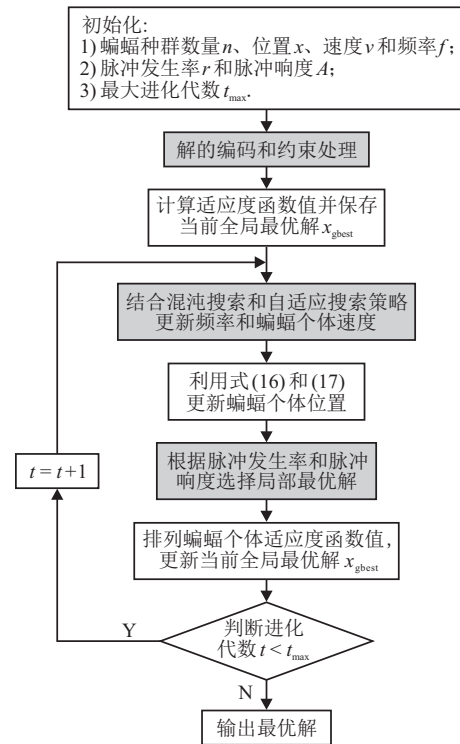


图2 DCBA优化算法流程

$\dots, x_{iD}]$.此时,解编码的冗余度较高,最优解空间较大.针对这种情况,采用强制修正蝙蝠个体位置的方法,即若蝙蝠个体位置不满足任务-资源分配约束,即 $\sum_{m=1}^M d_{mi}^{R-T} \neq 1$,则给该任务随机分配某项资源,因此,最优解的搜索范围将大大减少;

2)结合混沌搜索和自适应搜索策略更新脉冲频率和蝙蝠个体速度.

蝙蝠个体利用回声定位能力测算与当前最优解的距离,并根据脉冲频率调节速度,标准BBA优化算法中脉冲频率往往在频率区间 $[f_{\text{min}}, f_{\text{max}}]$ 随机取值,可能导致速度的变化陷入局部区间,特别是当蝙蝠搜索到当前最优解时,速度会被迫“停滞”^[14],影响搜索效率.此外,标准BBA优化算法并未考虑在进化初期需要保持较强的探索能力,而在进化后期需要保持较高的开发能力这一问题,导致算法存在开发与探索能力不平衡的缺点.为此,结合混沌搜索和自适应搜索策略更新脉冲频率和蝙蝠个体速度,提高算法搜索效率以及平衡开发和探索能力.

脉冲频率采用自适应折叠混沌搜索,即

$$f_i = f_{\text{min}} + (f_{\text{max}} - f_{\text{min}}) \cdot |y_i^k|. \quad (18)$$

其中: y_i^k 为自适应折叠混沌^[15],其混沌映射函数为

$$y_i^k = \cos[1/(y_i^{k-1})^2], \quad (19)$$

$y_i^0 \neq 0, k \in Z^+$.

蝙蝠个体速度的更新采用自适应搜索策略,即

$$v_{id}^{t+1} = \omega_{id}^t v_{id}^t + (x_{id}^t - x_{gbest_d}) f_i \eta_1 + (x_{id}^t - x_{jd}^t) f_i \eta_2. \quad (20)$$

其中: ω_{id}^t 为自适应惯性权重; $\eta_1 + \eta_2 = 1$ 为自适应学习因子; $j = 1, 2, \dots, n$ 且 $j \neq i$ 为随机选择的下标.

自适应惯性权重选取非线性递减惯性权重控制方案

$$\omega_{id}^t = \omega_{\min} + \frac{(t_{\max} - t)^{k_1}}{(t_{\max})^{k_1}} (\omega_{\max} - \omega_{\min}). \quad (21)$$

其中: $k_1 \in Z^+$, ω_{\min} 和 ω_{\max} 为惯性权重最小值和最大值. 自适应学习因子应保证算法在进化初期能够保持较强的探索能力, 提高全局搜索能力, 避免陷入局部最优; 随着进化代数增加, 逐渐由全局搜索转向对特定区域的精细化搜索. 因此, 自适应学习因子 η_1 和 η_2 随进化代数的更新为

$$\eta_1 = 1 - \text{rand} \frac{(t_{\max} - t)^{k_2}}{(t_{\max})^{k_2}},$$

$$\eta_2 = \text{rand} \frac{(t_{\max} - t)^{k_2}}{(t_{\max})^{k_2}}. \quad (22)$$

其中: $k_2 \in Z^+$, rand 为 $[0, 1]$ 范围内均匀分布产生的随机数.

3) 根据脉冲发生率和脉冲响度选择局部最优解.

标准 BBA 优化算法直接将当前全局最优解集中随机选择的解 \mathbf{x}_{old}^t 作为新解, 即

$$\mathbf{x}_{new}^t = \mathbf{x}_{old}^t. \quad (23)$$

通过借鉴标准 BA 优化算法的局部搜索过程, 并采用变异操作避免陷入局部最优, 即对当前全局最优解集中随机选择的解 \mathbf{x}_{old}^t 进行如下变异操作:

$$x_{newj}^t = \begin{cases} x_{oldj}^t, & j = 1, \dots, d_1 - 1; \\ 1, & \text{rand} < V(\varepsilon A_{ave}^t), & j = d_1, \dots, d_2; \\ 0, & \text{rand} \geq V(\varepsilon A_{ave}^t), & j = d_1, \dots, d_2; \\ x_{oldj}^t, & j = d_2 + 1, \dots, D. \end{cases} \quad (24)$$

其中 d_1 和 d_2 为随机位置.

在每次进化过程中, 采用自适应折叠混沌搜索更新脉冲发生率, 即

$$r_i^{t+1} = \left[r_{\min} + (r_{\max} - r_{\min}) \frac{t}{t_{\max}} \right] \cdot |y_i^t|, \quad (25)$$

其中 r_{\min} 和 r_{\max} 为脉冲发生率的最小值和最大值.

根据脉冲发生率和脉冲响度选择局部最优解的伪代码如下:

- 1) if $\text{rand} > r_i^t$;
- 2) 从最优解集中选择一个解 \mathbf{x}_{old}^t 作为最优解;
- 3) 利用式 (24) 在选择的局部最优解 \mathbf{x}_{old}^t 附近产生一

个新解 \mathbf{x}_{new}^t ;

4) end if;

5) 计算新解的适应度函数值 $f(\mathbf{x}_{new}^t)$;

6) if ($\text{rand} < A_i^t$ & $f(\mathbf{x}_{new}^t) \leq f(\mathbf{x}_{gbest})$);

7) 接受新解, 并按式 (14) 减小脉冲响度, 按式 (25) 增大脉冲发生率;

8) end if.

3 仿真算例

以联合作战装备精确保障任务^[6]为例, 保障任务集为 $T = \{T_1, T_2, \dots, T_{10}\}$, 任务优先级为 $P_T = \{P_{T_1} > P_{T_3} > P_{T_7} > P_{T_4} > P_{T_9} > P_{T_2} > P_{T_5} > P_{T_8} > P_{T_6} > P_{T_{10}}\}$, 各任务执行时序逻辑约束为 $T_3 \rightarrow T_4 \rightarrow T_5$, 保障任务完成时限为 $\text{TCL}_{\text{tot}} = 18$, 任务完成时限约束为 $\text{TCL}_{T_8} = 5$, 保障任务集 T 中任务间的时序约束关系 C^{T-T} 为

$$C^{T-T} = \begin{cases} 1, & c_{34}^{T-T}, c_{45}^{T-T}; \\ 0, & c_{ij}^{T-T}, i \neq 3, 4, j \neq 4, 5. \end{cases}$$

保障资源集为 $R = \{R_1, R_2, \dots, R_{12}\}$, 任务与资源匹配的相应执行时间矩阵 RT 为

$$\text{RT} = \begin{bmatrix} 3.5 & 2.1 & 5.0 & 0 & 2.5 & 3.3 & 0 & 2.5 & 5.1 & 0 \\ 3.8 & 1.8 & 4.8 & 8.3 & 2.0 & 0 & 4.8 & 3.0 & 0 & 3.4 \\ 0 & 3.1 & 5.5 & 8.2 & 2.4 & 3.1 & 4.0 & 2.0 & 5.5 & 4.4 \\ 3.5 & 3.2 & 5.1 & 9.0 & 3.0 & 3.2 & 5.0 & 2.3 & 0 & 3.8 \\ 4.5 & 3.0 & 4.5 & 7.9 & 3.1 & 3.7 & 5.5 & 0 & 5.6 & 3.3 \\ 4.5 & 2.0 & 4.7 & 7.5 & 2.5 & 3.0 & 4.5 & 0 & 5.8 & 3.5 \\ 0 & 2.5 & 0 & 8.0 & 2.7 & 3.9 & 4.1 & 2.0 & 5.1 & 4.0 \\ 4.0 & 1.9 & 0 & 0 & 0 & 4.0 & 4.3 & 3.6 & 6.0 & 4.8 \\ 5.0 & 3.8 & 5.4 & 0 & 0 & 0 & 0 & 2.3 & 5.5 & 3.3 \\ 3.7 & 0 & 5.5 & 0 & 2.8 & 0 & 0 & 2.0 & 5.3 & 4.1 \\ 3.4 & 0 & 6.0 & 7.9 & 3.0 & 4.1 & 4.7 & 2.5 & 0 & 3.4 \\ 3.0 & 0 & 5.2 & 7.6 & 3.5 & 3.6 & 4.8 & 0 & 5.3 & 4.0 \end{bmatrix}.$$

经过反复多次仿真实验, 确定本文所提出算法的最优参数设置如下: 种群数量 $n = 30$, 最大进化代数 $t_{\max} = 200$, 脉冲频率的最小值 $f_{\min} = 0$ 和最大值 $f_{\max} = 2$, 惯性权重最小值 $\omega_{\min} = 0.2$ 和最大值 $\omega_{\max} = 0.9$, $k_1 = k_2 = 2$, 脉冲发生率的最小值 $r_{\min} = 0.1$ 和最大值 $r_{\max} = 0.8$, 冷却因子 $\alpha = 0.9$. 采用本文算法得到的装备精确保障任务规划甘特图如图3所示.

由图3可知: 任务执行时序满足 $T_3 \rightarrow T_4 \rightarrow T_5$ 的逻辑顺序, 解决了任务执行时序逻辑约束; 任务 T_1, T_2, T_8 和 T_9 均占用资源 R_1 , 任务 T_5 和 T_7 均占用资源 R_2 , T_6 和 T_{10} 均占用资源 R_3 . 根据任务优先级对各任务进行协同规划, 可解决任务对资源占用冲突问题. 图3中最优解对应的任务-资源分配方案见表1.

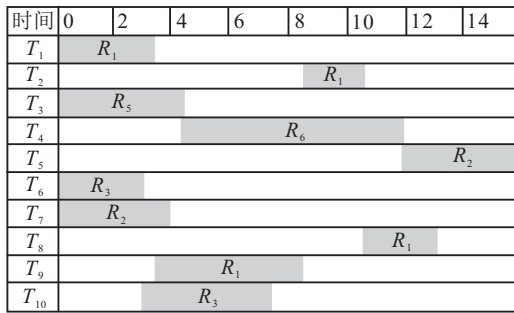


图3 装备精确保障任务规划甘特图

表1 任务-资源分配方案

| 任务 | 资源 | 任务 | 资源 |
|-------|-------|----------|-------|
| T_1 | R_1 | T_6 | R_3 |
| T_2 | R_1 | T_7 | R_2 |
| T_3 | R_5 | T_8 | R_1 |
| T_4 | R_6 | T_9 | R_1 |
| T_5 | R_2 | T_{10} | R_3 |

将本文算法与文献[6]中基于DPSO和GA的算法进行比较。DPSO和GA算法参数设置为: 惯性权重 $\omega = 0.5$, 加速因子 $c_1 = c_2 = 1.49$, 交叉概率 $p_c = 0.7$, 变异概率 $p_m = 0.01$ 。为了便于区分, 3种算法分别命名为MDLS-DCBA, MDLS-DPSO和MDLS-GA。3种算法的进化收敛曲线对比如图4所示。

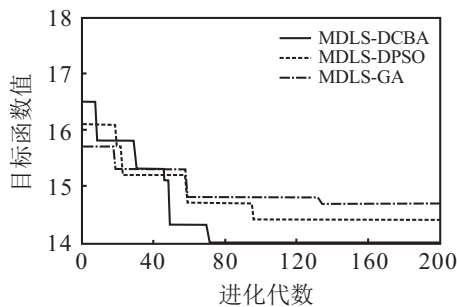


图4 3种算法进化收敛曲线

由图4可知: 本文算法得到的目标函数值即最优任务执行时间 $TIT = 14h$, 收敛代数 $t = 71$, 相比于另外两种方法具有更快的速度和更高的求解精度。

4 结论

本文针对装备精确保障任务-资源分配问题, 建立了以时效优先为目标的任务规划数学模型, 设计了以MDLS算法为主的求解框架, 嵌套具有自适应搜索策略和变异操作的DCBA优化算法为选定任务分配资源的任务分配方法。该算法利用MDLS选择处理的资源, 通过自适应调整惯性权重和学习因子实现优化算法的探索与开发能力的最佳平衡, 并采用混沌变异操作协助种群跳出局部最优。装备精确保障任务规划案例仿真结果表明, 所提出算法可以有效解决任务时序逻辑约束和资源占用冲突问题, 实现了装备精确保障任务规划和资源调度的科学性和有效性。

参考文献(References)

- [1] Xu H, Chen C L. Equipment efficient support generality[M]. Beijing: National Defense Industry Press, 2012: 1-8.
- [2] Levchuk G M, Levchuk Y N, Luo J, et al. Normative design of organization: Mission planning[J]. IEEE Trans on Systems, Man, and Cybernetics, Part A: Systems and Humans, 2002, 32(3): 346-359.
- [3] Yang D S, Zhang W M, Liu Z, et al. Research on mathematical description and solving algorithm of tasks scheduling for campaign[J]. Systems Engineering—Theory & Practice, 2006, 26(1): 26-34.
- [4] 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.
- [5] Wan 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.
- [6] Peng P F, Yu Q, Li Q Y. A planning method of equipment support task based on priority ordering and particle swarm optimization algorithm[J]. Acta Armamentarii, 2016, 37(6): 1082-1088.
- [7] Yang X S. A new metaheuristic bat-inspired algorithm[C]. Nature Inspired Cooperative Strategies for Optimization. Springer, 2010, 284: 65-74.
- [8] Yang X S, He S X. Bat algorithm literature review and applications[J]. Int J of Bio-Inspired Computation, 2013, 5(3): 141-149.
- [9] Pravesjit S. A hybrid bat algorithm with natural-inspired algorithm for continuous optimization problem[J]. Artificial Life and Robotics, 2016, 21(1): 112-119.
- [10] Geem Z W, Kim J H, Loganathan G V. A new heuristic optimization algorithm: Harmony search[J]. Simulation, 2001, 76(2): 60-68.
- [11] Kennedy K, Eberhart R C. A discrete binary version of the particle swarm optimization[C]. IEEE Int Conf on Computational Cybernetics and Simulation. Orlando, 1997: 4104-4108.
- [12] Bansal J C, Deep K. A modified binary particle swarm optimization for knapsack problems[J]. Applied Mathematics and Computation, 2012, 218(22): 11042-11061.
- [13] Mirjalili S, Mirjalili S M, Yang X S. Binary bat algorithm[J]. Neural Computing & Applications, 2014, 25(3/4): 663-681.
- [14] Chen J F, Wu T J. The property of search strategies of swarm intelligent algorithms and their influences on stagnation[J]. Systems Engineering—Theory & Practice, 2013, 33(6): 1587-1595.
- [15] Fu W Y, Ling Z D. An adaptive iterative chaos optimization method[J]. J of Xi'an Jiaotong University, 2013, 47(2): 33-38.

(责任编辑: 李君玲)