

doi: 10.7690/bgzd.2023.07.013

## 基于遗传算法的作战任务分配和资源调度问题研究

刘 建, 陈桂明, 李新宇

(火箭军工程大学保障学院, 西安 710025)

**摘要:** 在作战行动中出现多任务与多种保障资源合理调度困难的问题十分常见。笔者以任务完成的总时间最短为目标函数构建数学模型, 使用遗传算法进行迭代优化得到保障资源调度的全局最优解。对遗传算法进行自适应改进、移民交叉算子操作和迭代条件优化, 解决了过早收敛无法求出全局最优解的问题, 并使算法运行效率提高了76.4%。仿真实验结果表明: 该方法切实可行, 可以快速准确地完成多种保障资源调度并形成任务分配方案, 满足现阶段作战部队资源保障的现实要求。研究成果在高效完成保障资源调度的同时不产生冗余负担, 具有较好的应用价值和发展前景。

**关键词:** 任务分配; 资源调度; 装备保障; 遗传算法

**中图分类号:** TJ06 **文献标志码:** A

## Research on Combat Task Assignment and Resource Scheduling Based on Genetic Algorithm

Liu Jian, Chen Guiming, Li Xinyu

(College of Operational Support, Rocket Force University of Engineering, Xi'an 710025, China)

**Abstract:** In combat operations, it is very common to have difficulties in the rational scheduling of multi-task and multi-support resources. In this paper, a mathematical model is constructed with the objective function of the shortest total time of task completion, and the global optimal solution of support resource scheduling is obtained by using genetic algorithm for iterative optimization. The genetic algorithm is improved by self-adaptation, the operation of emigration and crossover, and the optimization of iterative conditions, which solves the problem that the global optimal solution can not be obtained by premature convergence, and improves the efficiency of the algorithm by 76.4%. The simulation results show that the method is feasible, and it can quickly and accurately complete a variety of support resource scheduling and form a task allocation scheme to meet the realistic requirements of combat force resource support at this stage. The research results have good application value and development prospects, which can efficiently complete the support resource scheduling without generating redundant burden.

**Keywords:** task allocation; resource scheduling; equipment support; genetic algorithm

### 0 引言

对多种保障要素进行统一调度, 可以有效解决长期以来多种保障要素独立行动、不协调、不同步、保障能力不均衡的问题<sup>[1]</sup>。现代作战任务往往是多任务交错并存且需要多种保障要素, 而保障资源在一定时间内的供给量及保障能力是一个有限的随机变量<sup>[2]</sup>。科学合理制定任务分配计划及资源调度方案, 对充分发挥保障效能, 高效精确地完成作战任务至关重要, 从而实现“任务—资源—时间”三者之间的最佳匹配<sup>[3]</sup>。资源调度问题的研究通过协同任务规划得到了初步解决, 但仍存在着诸多问题。张家良等<sup>[4]</sup>只考虑了任务时序逻辑关系约束和任务完成效能约束, 任务需求和实际作战能力仍存在不匹配的问题。王艳鹏等<sup>[5-7]</sup>基于动态列表规

划(dynamic list scheduling, DLS)方法, 结合遗传算法(genetic algorithm, GA)和量子遗传算法(quantum genetic algorithm, QGA)提出的混合任务分配方法以及针对装备保障任务规划问题, 提出基于优先排序和粒子群优化(particle swarm optimization, PSO)的混合任务规划方法, 与其他启发式智能优化算法一样, 普遍存在着全局搜索能力与局部搜索能力难以协调等问题。姚敏<sup>[8]</sup>使用的多种群遗传算法通过研究多目标并行调度问题一定程度上克服了GA的弊端。

笔者针对部队作战与装备运用的实际特点, 确定任务分配和资源调度的约束条件, 构建保障资源调度模型, 设计一种全新的资源分配算法, 并通过遗传算法进行优化迭代。通过一个仿真举例完成对

收稿日期: 2023-03-27; 修回日期: 2023-04-20

作者简介: 刘 建(1987—), 男, 山东人, 从事装备保障与资源优化研究。E-mail: 1025997454@qq.com。

算法的检验，得到任务序列甘特图和资源调度分配方案，证实了该算法的可行性和实用性。对传统的遗传算法进行了改进，算法性能进一步提升。

### 1 问题描述与模型构建

在多种要素保障多任务实施的问题背景下，分析约束条件，建立模型假设，通过对资源调度优化的指标设计确定优化指标函数，构建优化数学模型。

#### 1.1 问题描述

实施作战行动，在保障资源有限的前提下，既要满足任务的实施要求，又要尽可能使任务完成时间最短。假设保障单元需要保障  $n$  个作战任务。每项任务需要的保障要素数量各不相同。 $n$  个任务之间有多种联系，可以是串行的、并行的，也可以是串并共行的，如图 1 所示。

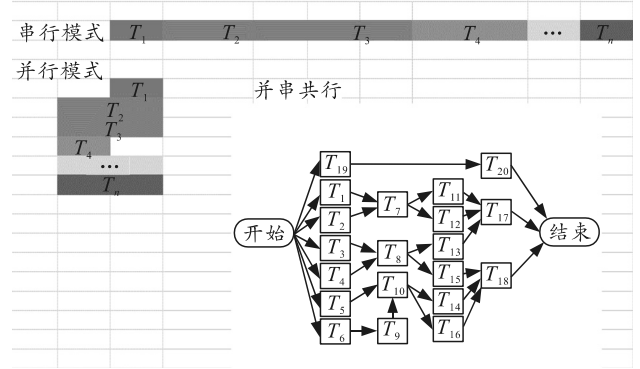


图 1 作战任务序列

$n$  个任务可构建出一个任务集  $T=\{T_i\}$ ，其包含的属性有任务完成时间，以及任务保障要素需求向量，构建一个任务需求要素表直观地展现出来，如表 1 所示。作战单位拥有的保障资源数量是有限的，可用编组保障要素属性如表 2 所示。

表 1 作战任务要素需求

作战任务	要素需求向量				$R$	$R_n$	开始时间	任务完成时间	结束时间
	油料车	通信车	诸元车	.....					
$T_1$	$x_1$	$y_1$	$z_1$				$Ts_1$	$t_1$	$Tf_1$
$T_2$	$x_2$	$y_2$	$z_2$		...		$Ts_2$	$t_2$	$Tf_2$
...				...			$Ts_3$	...	$Tf_3$
$T_n$	$x_n$	$y_n$	$z_n$		...		$Ts_4$	$t_n$	$Tf_4$

表 2 编组要素属性

属性	编组拥有的要素资源向量					
	油料车	通信车	诸元车	.....	$R$	$R_n$
拥有数量	$x$	$y$	$z$	...		

在作战任务的牵引下，将保障要素按照其需求进行分配。剩余的保障要素数量可以用于其他任务的保障，以同时进行的所有任务定义为一个梯队，梯队之间为串行序列。根据此办法依次求得满足约束条件下任务完成的序列模式和资源调度方案。

#### 1.2 问题约束与假设

假设问题资源更新时不考虑损耗情况，一个梯队打击完成后，所有保障要素就地进行原料补给与休整，并可以恢复到最初状态，等待下一梯队的保障行动，忽略了天气、人员士气、单位训练水平、装备质量等无关变量的影响，不再考虑其对作战保障效能发挥的影响。

##### 1.2.1 任务需求约束

参与第  $u$  项任务的第  $j$  种保障要素用  $X_{ju}$  描述，该量必须要大于等于完成该任务的需求量即  $Y_{ju}$ 。

$$X_{ju} \geq Y_{ju} \tag{1}$$

##### 1.2.2 资源限定约束

同一梯队打击中，要素消耗后不能补充，所以

执行所有任务所需要的某种保障要素不能超过该保障要素的总数。

$$\sum_{u=1}^U X_{ju} \leq Q_j \tag{2}$$

##### 1.2.3 现实条件约束

根据题设要求并结合实际可知，每个保障要素只能编入一个保障编组内或者不参与编组。

$$R_i(u)=1 \tag{3}$$

上式表示第  $i$  个保障要素保障第  $u$  个保障任务，否则为 0。

### 1.3 优化指标的设计

在保障单元协同作用和保障效能上精确匹配下，整个作战任务高度衔接，密切配合，使得任务完成的时间最短。由于梯队内任务为并行序列，在并行任务中，任务是同时进行的，各个任务之间互不干扰，各自为战。各个任务完成时间的最大值即为该梯队完成的时间，即为需要优化的指标。可以将其表示为：

$$\text{mint}_{bc}[j] \tag{4}$$

式中  $t_{bc}[j]$  表示梯队内任务为并行序列完成时间的最大值，即优化的原理为最小最大原则。

梯队之间为串行序列，在串行序列中，一个梯

队完成以后再执行下一个梯队，各个梯队之间紧密衔接，梯队之间存在原料补给的时间间隔  $t_{bj}$ 。各个梯队的任务完成时间之和就是需要优化的指标：

$$\min Tz = \sum_{j=1}^m t_{bc}[j] + (m-1) \times t_{bj}; \quad (5)$$

$$t_{bc}[j] = \max \{t[u]\}。 \quad (6)$$

式中： $t[u]$ 为同一梯队内  $u$  个任务完成时间； $j$  为某梯队； $m$  为共有梯队数量。

### 1.4 构建数学模型

由上述分析可知，保障单元编组模型的构建是一个资源调度和动态任务规划问题。在满足约束条件的前提下，优化编组和任务序列，使得优化的指标最小，即  $T$  最小，数学表达式为：

$$\min Tz = \sum_{j=1}^m \max \{t[u]\} + (m-1) \times t_{bj}。 \quad (7)$$

式中各符号含义与上文相同。

约束条件为：

$$\left. \begin{aligned} X_{ju} &\geq Y_{ju} \\ \sum_{u=1}^U X_{ju} &\leq Q_j \end{aligned} \right\}。 \quad (8)$$

## 2 优化算法的设计

由于保障要素数量受限，保障任务较多，所以将任务确定为串并共行的打击方案，并根据要素需求划分梯队打击。笔者的算法设计主要思路为随机产生一个任务序列，按照任务序列的顺序从前至后求在同一梯队的任务，直至最后一个任务分配结束，并按照这个序列进行遗传算法的实数编码，而后优化求解。

### 2.1 算法步骤

步骤 1：确定梯队内首先执行的任务。在编码给定顺序的保障任务集中选定编号第一个为首先发射的任务  $T[i]$ ，并按需求要素表分配保障要素数量。成功分配后更新现有的保障要素剩余总数。

步骤 2：确定同梯队最多并行任务。每确定一个任务，现有的保障要素剩余总数都会得到更新，将更新后的保障要素数量表和任务要素需求表进行对比，按照编码确定的任务顺序寻找出可以并行的最多任务编号，进而计算该梯队的完成时间  $t_{bc}[j]$ 。梯队完成后，将所有的保障要素恢复初始值，表示原料补给完成。

步骤 3：确定发射梯队。将已完成分配的任务

从未分配保障任务集中剔除，并重复以上步骤 1 和 2，进行发射梯队的确定，直到未分配保障任务集为空集。

步骤 4：计算任务完成总时间。根据步骤 2 的方法计算各个梯队完成的时间，在梯队确定后，加上梯队之间原料补给的时间，即为任务完成的总时间  $Tz$ 。

### 2.2 遗传算法寻优解算

在算法使用时需预先设定其遗传代数  $gen$ ，交叉概率  $pc$ ，变异概率  $pm$ ，种群规模  $n$ ，并将终止条件设定为达到遗传代数。

笔者采取实数编码的方式对任务序列进行编码，编码的数字代表任务编号，编码的位置代表任务进行的顺序<sup>[9]</sup>。以 8 个任务为例，编码示意图如图 2 所示。

3	1	7	2	5	8	6	4
---	---	---	---	---	---	---	---

图 2 编码

由 2.1 节的步骤 4 可计算得到总时间，即优化目标是要求总时间最短；因此，可将适应度函数值设定为  $BZfitnessVal=1/Tz$ ，当总时间最短时，适应度取得最大值。选择操作通过锦标赛法进行实现，即随机选择种群中 4 个个体，比较 4 个个体并选择其中适应度值最大的作为父代，可以将优良基因遗传下去<sup>[10]</sup>。

基于本问题的实际背景，变异操作使其发生在同一个体内，即个体自身变异更新个体<sup>[11]</sup>。当满足变异操作要求时，使交叉得到的优良个体进行变异，随机选择 2 个位置的编号，使其交换位置后得到新个体。变异示意如图 3 所示。交叉操作采用最常用的单点交叉，交叉只能发生在父代和母代相同的交叉区域。单点交叉即母代交叉点后段的链接到父代交叉点的前段，父代交叉点后段的链接到母代交叉点的前段，如图 4 所示。

个体1:	6	8	2	5	4	1	7	3
新个体:	6	8	7	5	4	1	2	3

图 3 变异

父代:	8	6	2	5	4	3	1	7
母代:	6	8	4	2	5	1	7	3
子代1:	8	6	2	5	4	1	7	3
子代2:	6	8	4	2	5	3	1	7

图 4 交叉

遗传算法进行优化的流程如图 5 所示。

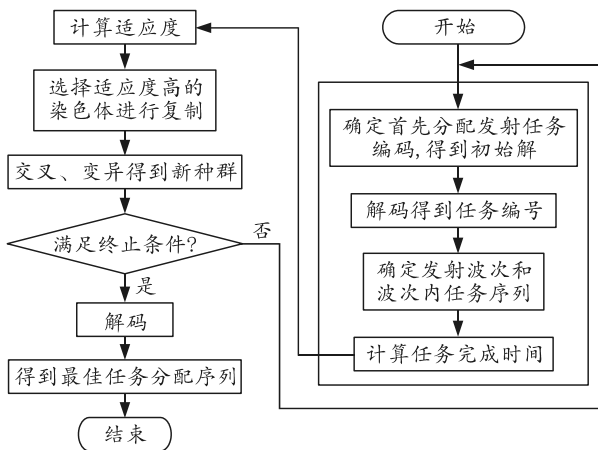


图 5 遗传算法优化的流程

### 3 实验仿真与结果分析

假设某次作战任务发射单元的编组共出动了 15 个发射架；因此，明确共有 15 个保障任务，进行补给的时间均为  $t_{bj}=5 \text{ min}$ 。简化问题便于实践，笔者只考虑油料补给车，车辆维修车和通信保障车 3 种保障要素，并给定 15 个任务的要素需求量和资源总数如表 3、4 所示。

表 3 拥有保障要素数量

油料补给车	车辆维修车	通信保障车
10	15	12

表 4 任务保障要素需求量

作战任务	要素需求向量			任务完成时间/min
	油料	维修	通信	
$T_1$	3	1	4	9
$T_2$	4	5	5	12
$T_3$	2	2	3	15
$T_4$	1	3	6	13
$T_5$	2	3	2	20
$T_6$	5	4	5	14
$T_7$	1	2	1	11
$T_8$	4	2	3	18
$T_9$	1	3	3	10
$T_{10}$	5	5	5	12
$T_{11}$	2	2	2	19
$T_{12}$	3	1	1	21
$T_{13}$	6	4	4	23
$T_{14}$	2	6	2	15
$T_{15}$	5	3	5	24

#### 3.1 遗传算法优化求解

遗传算法的相关参数设定为遗传代数  $gen=500$ ，种群规模  $n=200$ ，交叉概率  $pc=0.9$ ，变异概率  $pm=0.05$ ，按照图 5 的流程进行优化。算法实现的任务分配结果如图 6、7 所示。

由 7 图可知，任务完成的总时间最短为 112 min，但任务分配的方式并不唯一。为更直观地显示任务的梯队划分和保障资源的编组情况，分配结果的甘特图和保障编组情况如图 8、表 5 所示。

```

命令行窗口
9 1 2 13 3 5 8 4 7 11 10 6 14 12 15
第1梯队打击的任务是：9 1 2
第1梯队完成任务的时间是：12 min
第2梯队打击的任务是：13 3 5
第2梯队完成任务的时间是：23 min
第3梯队打击的任务是：8 4 7 11
第3梯队完成任务的时间是：19 min
第4梯队打击的任务是：10 6
第4梯队完成任务的时间是：14 min
第5梯队打击的任务是：14 12 15
第5梯队完成任务的时间是：24 min
最短任务完成时间为：112 min
    
```

图 6 任务分配结果

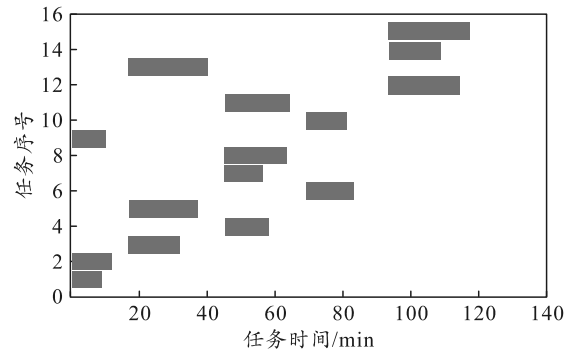


图 7 运行结果甘特图

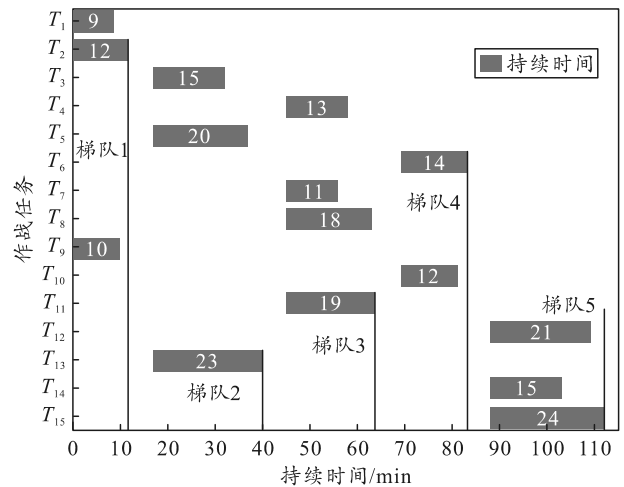


图 8 任务分配结果甘特图

表 5 保障要素编组结果

波次	发射任务	编组	油料补给车	车辆维修车	通信保障车
梯队 1	$T_9$	AG <sub>1</sub>	1	3	3
	$T_1$	AG <sub>2</sub>	3	1	4
	$T_2$	AG <sub>3</sub>	4	5	5
梯队 2	$T_{13}$	AG <sub>4</sub>	6	4	4
	$T_3$	AG <sub>5</sub>	2	2	3
	$T_5$	AG <sub>6</sub>	2	3	2
梯队 3	$T_8$	AG <sub>7</sub>	4	2	3
	$T_4$	AG <sub>8</sub>	1	3	6
	$T_7$	AG <sub>9</sub>	1	2	1
梯队 4	$T_{11}$	AG <sub>10</sub>	2	2	2
	$T_{10}$	AG <sub>11</sub>	5	5	5
	$T_6$	AG <sub>12</sub>	5	4	5
梯队 5	$T_{14}$	AG <sub>13</sub>	2	6	2
	$T_{12}$	AG <sub>14</sub>	3	1	1
	$T_{15}$	AG <sub>15</sub>	5	3	5

迭代进化曲线如图 9 所示，可知适应度函数值呈上升趋势，并在第 10 代左右已与最优适应度值趋平后在最优值附近波动并逐步收敛。该程序的运行时间约 52 s，运行速度较慢；因此，利用该遗传算法进行优化存在的问题，具有较大的改进空间。

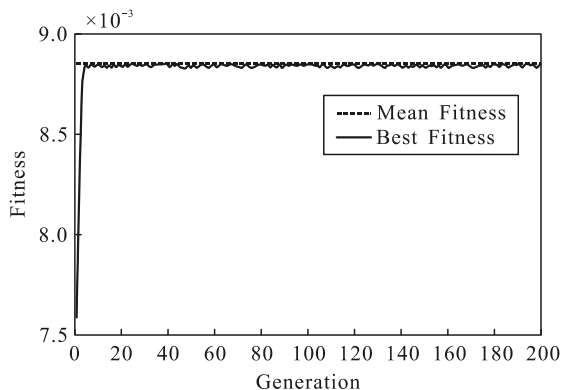


图 9 适应度值迭代进化曲线

### 3.2 遗传算法的改进

传统的遗传算法存在一定的局限性，笔者提出了自适应遗传算法和交叉移民算子 2 种改进方法，用于改善传统遗传算法过早收敛无法求出全局最优解的问题。对于计算时间较长的弊端，笔者通过对遗传算法的终止条件进行改善的方法加快了收敛速度，计算效率得到较好程度的改善。

#### 3.2.1 自适应遗传算法

自适应遗传算法通过对遗传参数的调整大大提高了遗传算法的收敛精度，加快了收敛速度<sup>[12]</sup>。所谓自适应是指交叉概率与种群的适应度值直接相关，适应度值越高，交叉概率越低，即可减少对优良种群的破坏，可以更好地将优良基因遗传下去。将实例 *pc* 值改为  $0.006/\text{fitnessVlaue}$  再进行实验，使得每个种群中的每个个体的 *pc* 概率值都根据个体的特点而有所不同，具备足够的独特性和适应性，每一次进化后交叉概率得以更新，并且使下次进化更有针对性，遗传算法的实践结果更加准确。

#### 3.2.2 移民交叉算子操作

姚敏<sup>[8]</sup>采用移民操作和交叉操作混合使用的方法，每次进化都使得目标种群既避免外来优秀个体迅速操作“种群入侵”，又可以优化进化效果。具体实现的方法是每次进化的变异操作过后选出个体中适应度值最大的个体和最小的个体作为算子操作对象。定义 *Ipc* 为移民交叉算子操作的触发条件，若满足移民交叉发生条件，则通过产生一个交叉点，使得最优个体交叉点位置之后的染色体片段移接到

最劣个体交叉点位置之后，同理将最劣个体交叉点位置之后的染色体片段移接到最优个体交叉点位置之后，便实现了同种群个体间的协同进化。

通过上述 2 种方法的改进，实验结果的收敛性得到较大程度的改善，改进后的遗传算法适应度值迭代进化曲线如图 10 所示。

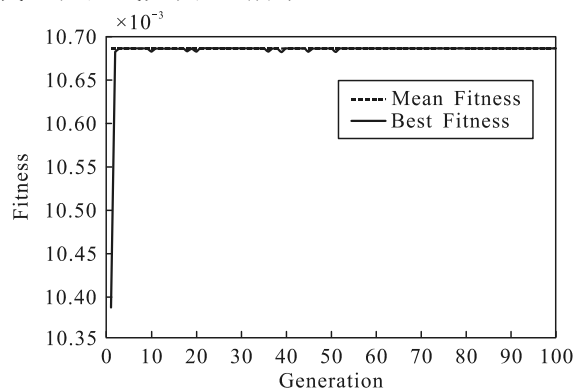


图 10 改进后的适应度值迭代进化曲线

#### 3.2.3 计算终止条件的改善

若固定进化代数作为遗传算法的终止条件，在进化到一定代数之后，适应度值在最优值附近摆动，对于优化的意义并不大，而且时间代价比较高，导致算法的效率极低。由之前方法可以获得最优任务完成时间在 115 s 左右的先验知识，若将适应度值引入到迭代终止条件中去，便可以提高算法的运行效率，缩短任务分配和保障要素编组时间。

若整个进化过程中均不满足循环终止条件，则当满足最大进化代数后算法终止。实验结果表明，在 118 代时已满足收敛条件。尽管算法处理时间具有一定的随机性，相比于传统算法 500 代时得到收敛最优解，该方法在 118 代即可完成，运行效率仍提高了 76.4%。结果运行示意如图 11 所示。

```

命令行窗口
进化至 第118代 中的 第1个体 为最优解:
最短时间任务执行方案为:
ans=
    2 1 9 14 7 11 8 6 10 15 4 12 3 5 13
第1梯队打击的任务是: 9 1 2
第1梯队完成任务的时间是: 12 min
第2梯队打击的任务是: 13 3 5
第2梯队完成任务的时间是: 23 min
第3梯队打击的任务是: 8 4 7 11
第3梯队完成任务的时间是: 19 min
第4梯队打击的任务是: 10 6
第4梯队完成任务的时间是: 14 min
第5梯队打击的任务是: 14 12 15
第5梯队完成任务的时间是: 24 min
最短任务完成时间为: 112 min
    
```

图 11 终止条件改善后运行结果