

基于高斯扰动粒子群算法的导弹并行测试策略优化方法

何子路, 吴亚星, 葛善乘, 丁伟, 崔北鹏, 关凯元, 唐健

(上海机电工程研究所, 上海 201109)

摘要: 在批产阶段, 需要在有限的时间内对多个导弹进行多项测试, 通过并行测试方式对测试项目进行重新调度排布可以压缩测试时间并提高测试资源利用率; 现有的并行测试方法大多预设了某一测试资源在某一时刻仅供一个测试项目使用, 并预先规定了线程数量, 未考虑测试资源实际上可支撑多个项目同时进行; 为了优化导弹测试流程, 提出了一种通过高斯扰动粒子群算法获取导弹并行测试最佳策略的方法, 将调度问题转化为优化问题, 并在有限的测试资源的约束下对某导弹测试项目进行了重新排布; 经过试验验证获取了合理的测试项目调度策略, 有效降低了测试时间, 大大提高了批产过程的测试效率。

关键词: 改进粒子群算法; 高斯扰动项; 并行测试; 导弹测试项目; 项目调度

Optimization Method for Parallel Testing Strategy of Missiles Based on Particle Swarm Algorithm with Gaussian Perturbation

HE Zilu, WU Yaxing, GE Shancheng, DING Wei, CUI Beipeng, GUAN Kaiyuan, TANG Jian

(Shanghai Electro-Mechanical Engineering Institute, Shanghai 201109, China)

Abstract: During the batch production stage, it is necessary to conduct multiple tests on multiple missiles within a limited time frame. Testing items are rescheduled and rearranged through parallel testing, which can reduce testing time and improve the utilization of testing resources. However, existing parallel testing methods often preset a certain testing resource to be used only for one testing project at a certain time, and specify the number of threads in advance, without considering that the testing resources can actually support multiple projects simultaneously. To optimize the missile testing process and obtain the optimal scheduling strategy for parallel testing of missiles, a particle swarm optimization algorithm with Gaussian perturbation is proposed. The testing items for a certain missile are rearranged under the constraint of limited testing resources, so the scheduling problem can be transformed into an optimization problem. After experimental verification, a reasonable testing project scheduling strategy is obtained, thus effectively reducing the testing time and significantly improving the testing efficiency during the batch production process.

Keywords: improved particle swarm algorithm; Gaussian perturbation term; parallel testing; missile testing project; project scheduling

0 引言

现阶段由于国际形势严峻, 我国需要研制与批产大量导弹以满足战略储备。而导弹是由多个单机设备组成的复杂系统, 需要对多个待验证的功能或参数设计相应的测试项目。常规的串行测试模式在导弹测试时只能在某一时间段内执行单一的测试项目, 测试设备中有相当

一部分资源在某项测试任务中处于闲置状态, 造成了测试设备的资源浪费, 延长了测试时间, 严重降低了工作效率。另外, 测试设备为导弹通电时间过长会导致温度迅速积累, 而达到一定的温度时需要降温以防止导弹内部构件烧坏, 因此进一步延长了实际的测试时间。在时间紧张的竞标、批产等阶段, 传统的串行测试方法会对测试人员产生很大压力^[1], 甚至造成因误操作而导致的

收稿日期:2025-03-26; 修回日期:2025-05-06。

作者简介:何子路(1991-),男,博士研究生,工程师。

引用格式:何子路,吴亚星,葛善乘,等.基于高斯扰动粒子群算法的导弹并行测试策略优化方法[J].计算机测量与控制,2025,33(9):182-190.

导弹损坏。在此背景下, 并行测试技术应运而生, 解决了传统串行测试效率低、资源利用率低等问题^[2]。并行测试最明显的特点是在某一时间段内能够同时执行一个或多个测试任务, 整体提高测试工作的效率, 显著缩短测试时间; 根据各个测试资源的实际情况减少资源空置时间, 提高测试系统整体的工作利用率。因此, 对支持多任务并行执行的调度策略优化方法进行研究具有重要意义^[3-9]。

目前, 并行测试任务排布的求解方法主要有两类^[10]。一种是采用 petri 网通过图形化建模工具获取并行测试流程图。海军航空工程学院的管晗等人通过建立广义 petri 网将激发的变迁序列集作为并行测试的任务调度路径, 并在此基础上对最优路径进行搜索^[11]; 重庆大学的刘正雷通过任务相关图转化算法, 建立了并行测试的 petri 网模型^[12]; 军械工程学院的卓佳靖等人通过 petri 网挖掘了各任务之间的并行性, 为测试任务调度提供了依据^[13]。然而, petri 网在任务数量增加的情况下容易造成模型状态空间信息爆炸, 因此通常只能用于小模型或与其他方法联合求解。

另一种常用的方法是通过各种智能优化算法对并行测试调度方法进行求解。北京交通大学的毛志宾等人采用混合人工蜂群算法对模型进行了动态规划与求解^[14]; 北京宇航系统工程研究所的和枫等人设计了有效的人工蜂群启发算法, 获得了测试项目的先后安排^[15]; 中国运载火箭技术研究院的李冬予等人对比了蚁群算法和自适应混沌免疫算法, 通过验证得出了蚁群算法效果更好的结果^[16]。另外, 遗传算法与粒子群算法等也能够用于并行测试的任务调度研究^[17-20]。

然而, 现有的并行测试任务调度研究工作大多预先限制了测试线程的数量, 即同时进行的测试项目的数量。而且通常把测试资源在某一时刻只分配给一个测试项目, 未考虑某些测试资源能够同时提供给若干个测试项目。因此, 需要针对导弹测试项目的实际情况对调度方法进行进一步优化。本文在分析导弹测试项目的特点下, 对实际情况进行了针对性建模, 提出了一种基于高斯扰动粒子群算法对导弹测试项目进行并行化改进排布的方法, 实现了对常规串行测试策略的优化。其中, 通过动态自适应调节惯性参数, 加快算法收敛速度并兼顾算法的全局与局部搜索能力; 另外添加了高斯扰动项, 进一步避免了粒子陷入局部最优陷阱的可能性, 维持了粒子在寻优过程中的活力。通过应用于某两个型号导弹在不同测试状态下的测试任务, 得到了各自的并行测试排布方案, 验证了该优化方法的有效性和正确性。

1 基于改进粒子群算法的优化方法

1.1 常规优化算法简介与对比

优化算法是一类以概率统计作为底层逻辑, 按照一

定步骤搜寻目标函数解空间中最大值或最小值的方法。常见的优化算法包括禁忌算法、遗传算法、粒子群算法、免疫算法等, 每种方法根据原理的不同而具有不同的特点和适用范围, 如表 1 所示。

表 1 常见优化算法特点对比

算法名称	原理特点	优点	缺点
禁忌搜索算法	每轮迭代中建立一个移动集逐步探索当前解周边区域,同时维护一个禁忌列表以降低局部最优陷阱的影响	求解过程前期寻优质量很高,具有较好的局部搜索能力	求解过程后期寻优质量下降,在搜索空间大的情况下可能无法找到全局最优解
遗传算法	通过选择、交叉、变异等方式模拟自然选择和遗传传播,搜寻最适应环境(函数)的基因(解)	具有较强的全局搜索能力	效率低,容易过早收敛,求解时间长
免疫算法	克隆原有的抗原(当前解),经过突变、接种等操作获取新抗原,直到达到免疫效果(收敛于最优解附近)	全局搜索能力较强,对复杂问题具有较好的适应性	需要跟多的计算资源和时间才能达到理想效果,对参数设置的要求较高
粒子群算法	创造多个粒子对整个解空间进行随机探索,记录当前与全局最优位置	步骤简单,参变量少,平均收敛速度快;求解过程中获取全局最优解的概率更大,所有粒子均可记忆优解信息,适用范围更大	数学理论基础不够牢固,需要根据实际情况进行适应性调整

由于粒子群算法步骤简单,调参容易,收敛速度快,本文通过基于权值自适应调节的改进粒子群算法将导弹测试项目的调度问题转化为求解并行测试时间最小值的优化问题,从而间接得到能够花费最短时间完成测试任务的并行测试项目排布方案。

1.2 粒子群算法

在粒子群算法中,首先需要确定粒子数量与结束条件。在一定数量范围内,粒子越多,覆盖性越大,粒子的搜寻范围越广泛,从而获得全局最优解的概率也会提升。但是过多的粒子数量会占据大量的计算资源,导致求解时间延长,而且粒子本身也具有相当程度的寻优能力,因此粒子的数量只需维持在一个合理范围之内即可。结束条件包括迭代次数和停止阈值,当算法循环求解一定次数或每次求解获得的结果维持在某一固定值附近的一段区域内,即可停止计算。

每一个粒子在求解过程中仅具备速度和位置两个属

性。粒子本身可视为一个能够描述原问题自变量的 D 维向量。在迭代过程中，每一个粒子的速度与位置均会进行更新，并且将位置作为候选解。每轮各个粒子的速度与位置进行比较后得到当前最优解。经过若干轮迭代之后，求解过程完成收敛或达成终止条件，从而得到全局最优解^[21-29]。粒子群算法中各个粒子的属性更新按照式 (1) 和式 (2) 进行：

$$\begin{aligned} V_{id}(k+1) &= \omega V_{id}(k) + \\ &C_1 r_1 [P_{id}(k) - x_{id}(k)] + \\ &C_2 r_2 [G_{id}(k) - x_{id}(k)] \quad (1) \\ x_{id}(k+1) &= x_{id}(k) + V_{id}(k+1) \quad (2) \end{aligned}$$

式中， $i=1, 2, \cdots, N, d=1, 2, \cdots, D, \mathbf{x}_i = [x_{i1}, x_{i2}, \cdots, x_{id}]$ 和 $\mathbf{V}_i = [V_{i1}, V_{i2}, \cdots, V_{id}]$ 为第 i 个粒子的位置和速度。 ω 为惯性权重因子； k 为当前迭代次数； C_1 和 C_2 为学习因子； r_1 和 r_2 为 0 到 1 之间均匀分布的随机数； $P_{id}(k)$ 和 $G_{id}(k)$ 分别为第 i 个粒子的适应度和全局适应度，即待求解问题的最大值或最小值。

1.3 粒子群算法的改进措施

收敛速度和全局搜索能力是优化算法最重要的两个性能。收敛速度越快，算法的求解速度越快，处理复杂问题的效率越高；全局搜索能力代表算法的准确性，目标函数往往都是具有多个极值的多维函数，全局搜索能力越高，则算法求解时跳出极值的概率越大，越容易收敛于函数的最值处。粒子群算法由于原理简单，步骤较少，平均的收敛速度较其他算法更快，但是仍然具备一定的优化空间。另外，尽管粒子群算法中的粒子具有记忆性，但全局搜索能力相比于其他算法优势仍不算突出，需要针对算法特性对全局搜索能力进行进一步的提升。

惯性参数 ω 是粒子群算法里的关键参数，其大小能够影响粒子的探索能力。 ω 的取值通常在 0 到 1 之间，其值由大到小则搜索能力可以从全局转移到局部。常规的粒子群算法中， ω 为接近 1 的常量，在求解时会快速收敛，但是容易陷入局部最优解之中无法跳出，难以获取整个域内的最优值。为了保障算法在整个求解过程中兼顾效率与准确性，需要对粒子群算法的 ω 的取值方式进行改进。根据算法的求解特点，应使 ω 取值随着每一轮的迭代逐步递减，从前期的探索每个粒子周围的局部最优解转移到后期的快速收敛于精英粒子所在的全局最优解附近。

根据这一策略，采用惯性参数自适应动态调节方法，可以保障粒子在前期具有较大的活跃度，而在后期能够使粒子汇聚于全局最优解附近。通常可以使用以下几种比较易于实现的方式调整权值：

1) 线性降低；

- 2) 凹二次函数型降低；
- 3) 凸二次函数型降低；
- 4) 正切函数型降低；
- 5) 反正切函数型降低。

以上几种调整权值的方式各有特点，在使用中应根据实际情况与要求进行选择与参数设计。

表 2 不同权值调整策略的优点

调整策略	特点
线性降低	计算简单,参数设置容易
凹二次函数型降低	计算相对简单,参数设置相对容易,前期惯性参数降低速度快,收敛速度快
凸二次函数型降低	计算相对简单,参数设置相对容易,前期惯性参数降低速度慢,收敛速度慢
正切函数型降低	计算较为复杂,参数设置相对复杂,全局收敛速度快
反正切函数型降低	计算较为复杂,参数设置相对复杂,计算稳定性强

惯性参数自适应动态调节策略能够有效加速粒子群算法的收敛，但是对全局搜索能力的提升存在一定限制。各个粒子仍然有一定概率逐渐收敛于局部最优解，粒子随着迭代次数的增加被束缚于这个解周围无法跳离，从而无法获得全局最优解，导致并行测试调度策略效果大打折扣。在迭代过程中为粒子群加入高斯扰动可以对粒子群搜索速度进行干扰，保持差异性粒子数量，大大提高粒子跳出局部最优的概率。添加高斯局部扰动项后的粒子速度更新如式 (3)：

$$\begin{aligned} V_{id}(t+1) &= \omega V_{id}(t) + \\ &C_1 r_1 [P_{id}(t) + r_{g1} H_{id}(t) - x_{id}(t)] + \\ &C_2 r_2 [G_{id}(t) - x_{id}(t)] \quad (3) \end{aligned}$$

式中， r_{g1} 是 (0, 1) 之间的随机数； $H_{id}(t)$ 为粒子 i 在第 t 次迭代中的高斯扰动项。

$$H_{id}(t) = r_{g2} \text{Gaussian}(\mu, \sigma^2) \quad (4)$$

式中， r_{g2} 是 (0, 1) 之间任一随机数； μ 是高斯分布的均值； σ^2 是方差。

整个算法的计算流程如图 1 所示。

为方便理解高斯扰动对寻优机制的影响，以图 2 的二元函数为例进行示意。在添加高斯扰动项之前，粒子一旦陷入局部最优陷阱便只能收敛于局部最优解。

添加高斯扰动项之后，大量粒子中总会有部分粒子跳出局部最优陷阱，进而搜寻到全局最优解，如图 3 所示。

2 导弹测试项目特点与模型建立

2.1 导弹测试项目特点

在现代战争中，复杂的战场环境对导弹性能与功能

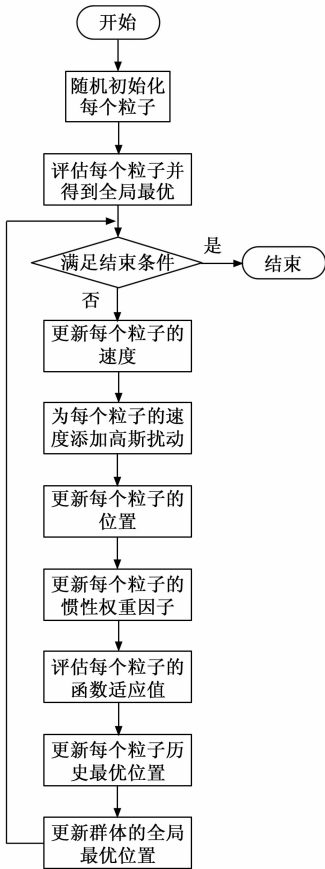


图 1 高斯扰动粒子群算法流程图

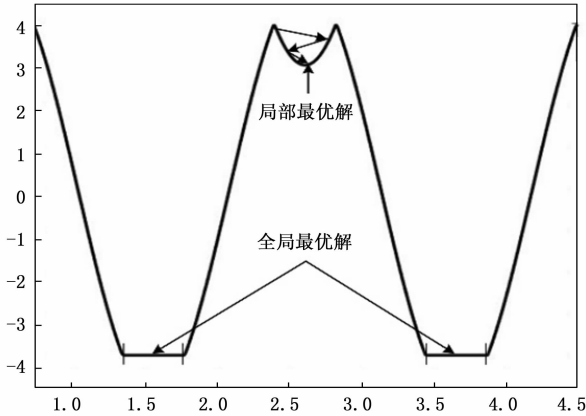


图 2 粒子陷入局部最优

提出了越来越高的要求，因此导弹的系统结构也越发复杂。导弹内部的各个单机系统及其模块构成众多且精确度高，内部关联性强，在总装过程中的操作不当或某一模块的设计不合理等问题均有可能造成连锁反应导致整个导弹发生故障。导弹系统成本高，无法完成作战任务的导弹会对研制资金造成浪费。另外，发生故障的导弹具有相当的危险性，对人的生命财产与运载装备造成严重威胁。因此在导弹的全寿命周期内的每一个阶段均需

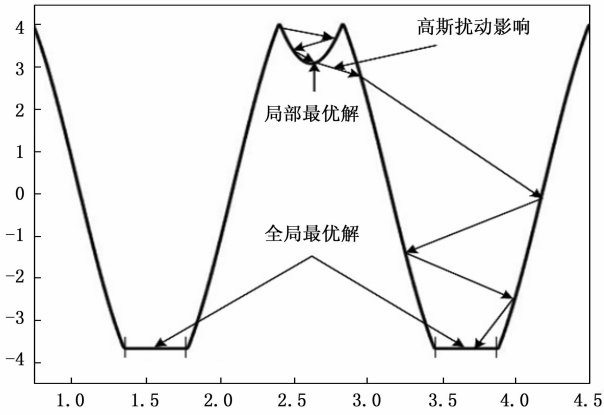


图 3 粒子受高斯扰动跳出局部最优

要众多测试项目以监测导弹的健康状态，保障导弹能够在发射前保持正常，如有问题能够及时发现并处理。因此，导弹测试项目具有如下特点：

- 1) 数量较多。测试项目需要按照测试要求覆盖各个测试状态下关键参数与功能的测试；
- 2) 时间参差不齐。有些测试项目只需保证指令回告正常即可，有些测试项目需要持续一段时间以获取足够的数据，因此各个测试项目的持续时间差异较大；
- 3) 存在部分时序要求。部分测试项目需要按照一定顺序进行，并且少数项目的执行时间必须大于等于一个固定的时间，否则可能获取错误结果甚至损坏测试设备；
- 4) 测试资源复用率高。每个测试资源都需要分配给多个测试项目使用。有些测试资源支持多个项目同时进行，有些不支持。另外，各种测试资源能够同时支持的项目数量也各不相同，因此测试资源为并行测试提供的约束存在较大的复杂性。

2.2 并行测试数学模型建立

针对上述导弹测试项目特点，可以将数学模型构建如下：

测试任务集，即 k 个测试任务组成的集合：

$$M = [m_1 \quad m_2 \quad m_3 \quad \cdots \quad m_k] \tag{5}$$

测试资源集，即所有测试任务需要用到的 n 个资源：

$$R = [r_1 \quad r_2 \quad r_3 \quad \cdots \quad r_n] \tag{6}$$

测试时间集，即单独运行 k 个测试任务的所需的时间：

$$T = [t_1 \quad t_2 \quad t_3 \quad \cdots \quad t_k] \tag{7}$$

资源容量集，即各个资源能够同时容纳的测试任务的数量：

$$C = [c_1 \quad c_2 \quad c_3 \quad \cdots \quad c_n] \tag{8}$$

任务资源占用矩阵，代表第 i 个测试任务与第 j 个

资源之间的占用关系：

$$\mathbf{MR}^{k \times n} = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{k1} & p_{k2} & \cdots & p_{kn} \end{bmatrix} \quad (9)$$

其中：若第 i 个任务占用了第 j 个资源，则 $p_{ij}=1$ ，否则 $p_{ij}=0$ 。特殊地，由于测试线缆承受电流的能力有限，可以把测试任务的电流看作一种可量化的特殊的资源。此时 p_{ij} 可以根据实际情况取大于等于 0 的值。

时序关系矩阵，代表各个测试任务之间的时序约束：

$$\mathbf{TC}^{k \times k} = \begin{bmatrix} q_{11} & q_{12} & \cdots & q_{1k} \\ q_{21} & q_{22} & \cdots & q_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ q_{k1} & q_{k2} & \cdots & q_{kk} \end{bmatrix} \quad (10)$$

其中：若第 j 个任务需要在第 i 个任务结束后 τ s 后才能开始进行测试，则 $q_{ij}=\tau$ ，否则 $q_{ij}=0$ 。

起始时间矩阵，代表各个测试任务的开始时间：

$$\mathbf{TS} = [t_{s1} \quad t_{s2} \quad t_{s3} \quad \cdots \quad t_{sk}] \quad (11)$$

并行测试的数学模型求解目标为：获取并行测试任务调度的最优序列，即求解得到合适的 $\mathbf{TS}$ 向量，使得测试结束时间最短。粒子群算法的适应度函数即目标函数可以表示为式 (12)：

$$F = \min(\mathbf{TS} + \mathbf{T}) \quad (12)$$

3 试验验证

以某两个型号的测试项目为例进行并行测试任务调度优化。

型号一在某测试状态下共有 12 个测试项目，整套测试设备中共有 8 个测试资源可供使用，其中资源 3 为电流资源。型号一具体的参数设置如表 3、表 4 所示。

表 3 型号一测试任务情况

测试任务	测试时间/s	测试电流/A
m_1	25	3
m_2	15	2
m_3	95	3
m_4	45	2
m_5	98	3
m_6	42	2
m_7	12	3
m_8	240	2
m_9	30	3
m_{10}	60	2
m_{11}	90	3
m_{12}	10	3

表 4 型号一测试资源情况

测试资源	资源容量
r_1	5
r_2	3
r_3	20
r_4	5
r_5	3
r_6	10
r_7	10
r_8	10

在型号一中，任务 m_1 占用的资源为 r_1 、 r_2 、 r_3 、 r_4 、 r_6 、 r_7 、 r_8 ，其中 r_3 为电流资源，因此取值为 3。以此类推，任务资源占用矩阵为

$$\mathbf{MR}_1 = \begin{bmatrix} 1 & 1 & 3 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 2 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 3 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 2 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 3 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 2 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 3 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 2 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 3 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 2 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 3 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 3 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (11)$$

在型号一中，测试项目二必须在测试项目一结束至少 30 s 之后才能开始；测试项目四在测试项目三结束 20 s 之后才能开始；测试项目五在测试项目九结束 5 s 之后才能开始；测试项目六在测试项目五结束 240 s 之后才能开始；测试项目七在测试项目五结束 5 s 之后才能开始；测试项目八在测试项目五结束 5 s 之后才能开始；测试项目八在测试项目七结束 5 s 之后才能开始；测试项目六在测试项目八结束 5 s 之后才能开始。由于时序关系矩阵大小为 12×12 ，仅有少数项目之间存在顺序约束，因此该矩阵为稀疏矩阵，在式 (12) 中仅对存在约束的元素进行赋值定义如下：

$$\left\{ \begin{array}{l} \mathbf{TC}_1(1,2) = 30 \\ \mathbf{TC}_1(3,4) = 20 \\ \mathbf{TC}_1(9,5) = 5 \\ \mathbf{TC}_1(5,6) = 240 \\ \mathbf{TC}_1(5,7) = 5 \\ \mathbf{TC}_1(5,8) = 5 \\ \mathbf{TC}_1(7,8) = 5 \\ \mathbf{TC}_1(8,6) = 5 \end{array} \right. \quad (12)$$

通过基于高斯扰动的粒子群算法对上述型号一的模型进行并行测试策略求解。其中, C_1 和 C_2 取值为 2, 粒子数为 1 000 个, 迭代次数为 500 次, 惯性参数采用反正切函数型降低方式从 0.9 降至 0.1。获得的并行测试时序图如图 4 所示。

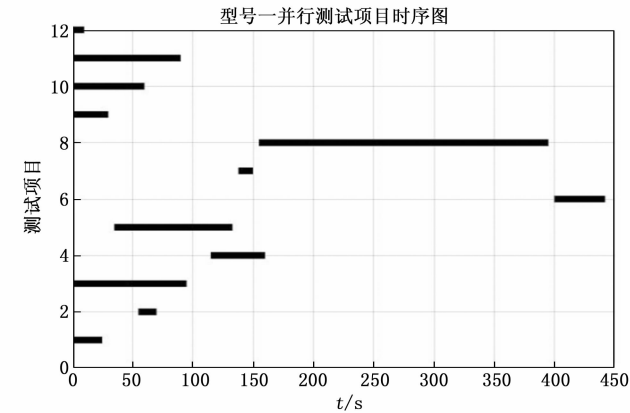


图 4 型号一并行测试时序图

由图 4 可以看出, 8 个测试资源的最大占用情况分别为 5、3、17、4、2、6、6、6, 符合测试资源的占用要求; 测试项目二在测试项目一结束之后 30 s 开始, 测试项目四在测试项目三结束之后 20 s 开始, 测试项目五在测试项目九结束之后 5 s 开始, 测试项目六在测试项目五结束之后 267 s 开始, 测试项目七在测试项目五结束之后 5 s 开始, 测试项目八在测试项目五结束之后 22 s 开始, 测试项目八在测试项目七结束之后 5 s 开始, 测试项目六在测试项目八结束之后 5 s 开始, 均满足测试项目的时序要求。各项测试任务的开始时间分别为 0、55、0、115、35、400、138、155、0、0、0、0 s, 总测试时间为 442 s, 串行测试总时长为 762 s, 优化得到的并行测试策略相比常规串行测试节省时间为 320 s。

为了验证算法的鲁棒性, 以型号一为例, 保持其他因素不变, 分别对测试数量、测试顺序和资源容量进行微调。首先调整测试项目的数量。剔除表 3 中的 m_{12} 后进行求解, 获得了图 5 所示的并行测试时序图。

由图 5 可以看出, 8 个测试资源的最大占用情况分别为 5、2、14、3、1、5、5、5, 符合测试资源的占用要求; 测试顺序约束与未调整之前相同。各项测试任务的开始时间分别为 0、55、0、115、35、400、138、155、0、0、0、0 s, 总测试时间为 442 s。

其次调整测试项目的顺序。将原来测试项目 2 在项目 1 结束至少 30 s 后开始调整为项目 1 在项目 2 结束至少 30 s 后开始, 获得了图 6 所示的并行测试时序图。

由图 6 可以看出, 8 个测试资源的最大占用情况分别为 5、2、16、4、2、6、6、6, 符合测试资源的占用

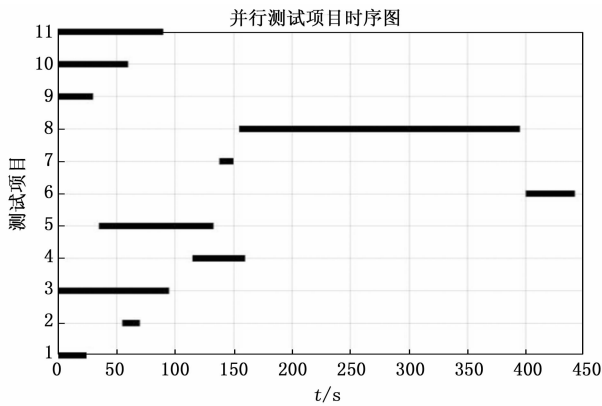


图 5 调整测试数量后的时序图

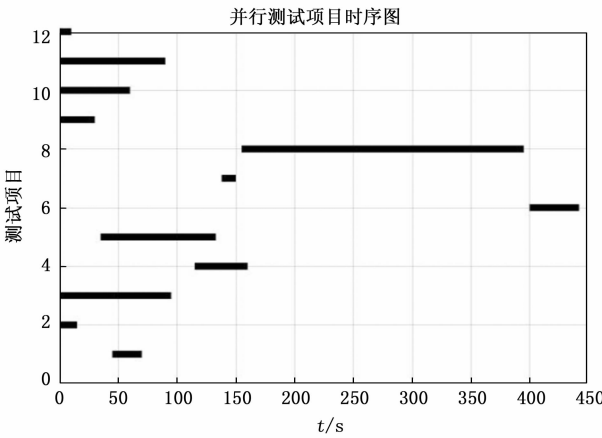


图 6 调整测试顺序后的时序图

要求; 测试项目一在测试项目二结束之后 30 s 开始, 其他顺序约束与未调整之前相同。各项测试任务的开始时间分别为 45、0、0、115、35、400、138、155、0、0、0、0 s, 总测试时间为 442 s。

最后调整测试资源的容量。将表 4 中的 r_8 的值调整为 5, 获得了图 7 所示的并行测试时序图。

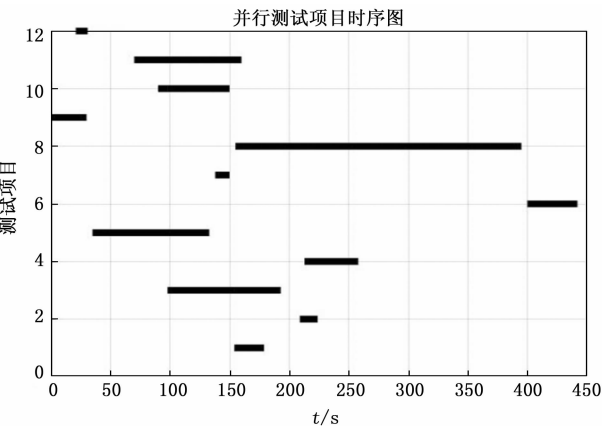


图 7 调整资源容量后的时序图

由图 7 可以看出, 8 个测试资源的最大占用情况分别为 4、2、11、3、1、4、4、4, 符合测试资源的占用要求; 测试项目二在测试项目一结束之后 30 s 开始, 测试项目四在测试项目三结束之后 20 s 开始, 测试项目五在测试项目九结束之后 5 s 开始, 测试项目六在测试项目五结束之后 267 s 开始, 测试项目七在测试项目五结束之后 5 s 开始, 测试项目八在测试项目五结束之后 22 s 开始, 测试项目八在测试项目七结束之后 5 s 开始, 测试项目六在测试项目八结束之后 5 s 开始, 均满足测试项目的时序要求。各项测试任务的开始时间分别为 154、209、98、213、35、400、138、155、0、90、70、21 s, 总测试时间为 442 s。

型号二在另一测试状态下共有 20 个测试项目, 整套测试设备中同样共有 8 个测试资源可供使用, 其中资源 3 为电流资源。型号二具体的参数设置如表 5~6 所示。

表 5 型号二测试任务情况

测试任务	测试时间/s	测试电流/A
m_1	10	2
m_2	30	1
m_3	120	1
m_4	50	2
m_5	50	2
m_6	50	2
m_7	80	1
m_8	30	4
m_9	180	4
m_{10}	40	1
m_{11}	40	5
m_{12}	60	3
m_{13}	30	3
m_{14}	30	3
m_{15}	50	3
m_{16}	30	2
m_{17}	50	4
m_{18}	10	3
m_{19}	30	1
m_{20}	100	2

表 6 型号二测试资源情况

测试资源	资源容量
r_1	7
r_2	5
r_3	20
r_4	10
r_5	3
r_6	10
r_7	10
r_8	10

$$MR_2 = \begin{bmatrix} 1 & 0 & 2 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 2 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 2 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 2 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 4 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 4 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 5 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 3 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 3 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 3 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 3 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 2 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 4 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 3 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 2 & 1 & 0 & 1 & 1 & 1 \end{bmatrix} \quad (13)$$

型号二的时序关系矩阵为:

$$\begin{cases} TC_2(1,2) = 10 \\ TC_2(6,8) = 100 \\ TC_2(10,11) = 20 \\ TC_2(13,14) = 10 \\ TC_2(15,16) = 10 \\ TC_2(11,20) = 90 \end{cases} \quad (14)$$

通过基于高斯扰动的粒子群算法对型号二模型进行求解。采用与型号一相同的参数设置, 获得的并行测试时序图如图 8 所示。

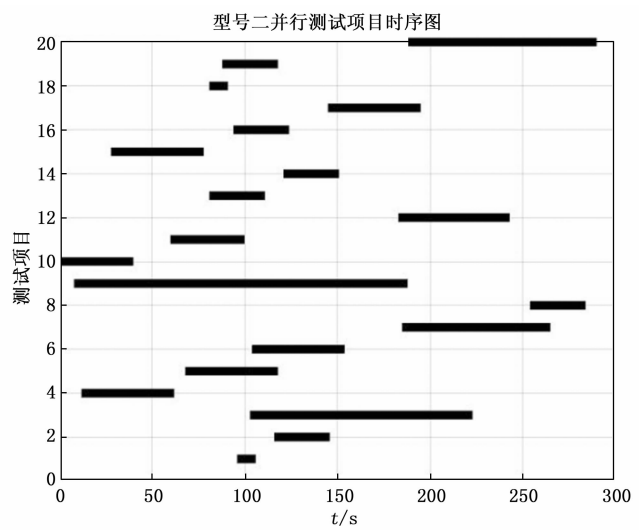


图 8 型号二并行测试时序图

型号二的任务资源占用矩阵为:

由图8可以看出,8个测试资源的最大占用情况分别为6、4、17、8、2、8、8、7,符合测试资源的占用要求;测试项目二在测试项目一结束之后10 s开始,测试项目八在测试项目六结束之后100 s开始,测试项目十一在测试项目十结束之后20 s开始,测试项目十四在测试项目十三结束之后10 s开始,测试项目十六在测试项目十五结束之后16 s开始,测试项目二十在测试项目十一结束之后90 s开始,均满足测试项目的时序要求。各项测试任务的开始时间分别为96、116、103、12、68、104、185、254、8、0、60、183、81、121、28、94、145、81、88、190 s,总测试时间为290 s,串行测试总时长为1 070 s,优化得到的并行测试策略相比常规串行测试节省时间为780 s。

通过对两个型号导弹测试项目的并行化改进,大大提高了测试效率,节约了时间资源与人力资源。实际上,对于某些约束较松的情况,不同的并行测试排布方案均可以取得最优解,某几个测试项目的时间可以提前或滞后一些,甚至可以采用其他不同的排布方案,其所需时间均为最短测试时间。此时实际取得的并行测试排布方案与粒子初始位置关系较大,如需获取特定排布方案,可以在模型当中添加约束。

4 结束语

针对以往解决并行测试问题时预设的某一测试资源在某一时刻只能供给一个测试项目这一简化模型,添加了动态约束条件,为同时使用某测试资源的测试项目数量增加了弹性,使整个求解过程更加符合客观实际。引入了动态自适应惯性因子和高斯局部扰动项对粒子群算法进行改进,提高了粒子的活性,避免了常规粒子群算法容易陷入局部最优的问题,提高了粒子群算法的收敛速度与求解精度,能够根据各种测试状态的特点更快地获得合理的并行测试调度策略。通过对两个不同型号导弹的测试任务进行并行化重新排布,分别将测试时间缩短至原来的58.0%和27.1%,验证了并行测试策略优化方法的效果。该方法可以在根据实际的测试情况在其他产品中具有更加广泛的应用。

参考文献:

- [1] 高括,刘磊,杨杰峰,等.适用于多星并行测试的通用测控地面测试平台设计与实现[J].计算机测量与控制,2019,27(1):1-4.
- [2] 卢慧卿,方丹,许爱华.通用ATS并行测试的任务调度模型研究[J].计算机测量与控制,2014,22(4):1033-1035.
- [3] 梁海波,姜苹,王尚,等.基于并行测试技术的地面测控系统自检方法[J].航天控制,2021,39(2):69-75.
- [4] 王淦,窦骄,纪文章,等.适用于多星并行测试的应答机设计方法与验证[J].遥测遥控,2020,41(3):42-47.
- [5] 王正元,刘卫东,景慧丽,等.一种并行测试任务调度优化方法[J].兵工学报,2018,39(2):399-404.
- [6] 宋扬,王清海.并行测试技术工程化应用研究[J].航空兵器,2017(2):65-70.
- [7] 夏锐,肖明清,朱小平,等.并行测试技术在自动测试系统中的应用[J].计算机测量与控制,2005,13(1):7-10.
- [8] 谢化勇,肖明清,陈伟明,等.导弹并行测试系统设计与任务调度仿真[J].西安交通大学学报,2009,43(10):22-26.
- [9] 卢茜.基于模拟退火禁忌遗传算法的并行测试任务调度[D].上海:上海交通大学,2015.
- [10] 陈利安,肖明清,高峰,等.人工蜂群算法在并行测试任务调度中的应用[J].计算机测量与控制,2012,20(6):1470-1472.
- [11] 管哈,李文海,王怡苹.基于GSPN和人工免疫算法的并行测试任务调度研究[J].测控技术,2017,36(12):67-70.
- [12] 刘正雷.基于petri网和蚁群算法的并行测试系统任务调度问题研究[D].重庆:重庆大学,2015.
- [13] 卓家靖,孟晨.并行测试系统的任务分解和任务过程模型[J].电子测量技术,2008,31(8):109-112.
- [14] 毛志宾,任慧敏,鲁承金,等.基于混合人工蜂群算法的并行测试任务优化研究[J].计算机测量与控制,2024,32(2):36-41.
- [15] 和枫,王晓明,叶志鹏,等.基于H-ABC算法的并行测试任务优化研究[J].导弹与航天运载技术(中英文),2023(5):146-152.
- [16] 李冬予,田小川,刘念,等.基于有限资源并行测试任务调度研究[J].兵器装备工程学报,2021,42(6):219-222.
- [17] 王娟,郑超,崔海青,等.基于改进粒子群的并行自动测试任务调度方法[J].电光与控制,2024,31(11):90-95.
- [18] 胡雷刚,付新华,肖明清,等.基于随机遗传算法的并行测试任务调度研究[J].电测与仪表,2008,45(10):41-45.
- [19] 夏锐,肖明清,程进军.基于混合遗传退火算法的并行测试任务调度优化[J].系统仿真学报,2007,19(15):3564-3567.
- [20] 戴少怀,王磊,李旻,等.基于遗传算法的SVM自适应干扰样式选择[J].空天防御,2020,3(2):59-64.
- [21] CRASSIDIS J L, CHENG Y. Three-axis magnetometer calibration using total least squares[J]. Journal of Guidance, Control, and Dynamics, 2021, 44(8): 1410

-1424.

- [22] WANG H, JIN Y, DOHERTY J. Committee-based active learning for surrogate-assisted particle swarm optimization of expensive problems [J]. IEEE Transactions on Cybernetics, 2017, 47 (9): 2664 - 2677.
- [23] BUI D T, BUI Q T, NGHYEN Q P, et al. A hybrid artificial intelligence approach using GIS-based neural-fuzzy inference system and particle swarm optimization for forest fire susceptibility modeling at a tropical area [J]. Agricultural and Forest Meteorology, 2017, 233: 32 - 44.
- [24] DING Y, CHENG L, PEDRYCZ W, et al. Global nonlinear kernel prediction for large data set with a particle swarm-optimized interval support vector regression [J]. IEEE Transactions on Neural Networks and Learning Systems, 2023, 34 (1): 175 - 186.
- (上接第 181 页)
- 参考文献:**
- [1] 姜 斌, 程月华, 孙 颢, 等. 一种利用 LSTM-FCN 的导弹回路故障诊断算法 [J]. 宇航学报, 2023, 44 (5): 687 - 698.
- [2] 王权达, 徐 博, 张 涛. AUV 导航系统故障诊断与容错技术研究综述 [J]. 战术导弹技术, 2023 (4): 26 - 36.
- [3] 丛林虎, 邓建球, 赵建忠, 等. 基于改进证据理论的导弹故障智能诊断推理方法 [J]. 战术导弹技术, 2018 (5): 31 - 37.
- [4] 刘兆政, 肖明清, 朱海振, 等. 基于置信规则库的机载导弹故障诊断 [J]. 空军工程大学学报: 自然科学版, 2020, 21 (3): 25 - 30.
- [5] 钦 劲, 董豆豆, 孙 权, 等. 基于故障树的导弹故障快速诊断决策支持系统 [J]. 装备指挥技术学院学报, 2006 (1): 111 - 114.
- [6] 朱向明, 周凤岐, 王继红. 防空导弹飞行试验集成故障诊断专家系统 [J]. 计算机应用研究, 2010, 27 (7): 2551 - 2555.
- [7] ZHU D, ZENG W, SU J. Construction of transformer substation fault knowledge graph based on a depth learning algorithm [J]. International Journal of Modeling, Simulation, and Scientific Computing, 2023, 14 (1): 493 - 504.
- [8] 李 津. 基于本体库的知识抽取及图谱构建技术 [J]. 科技创新与应用, 2024, 14 (11): 37 - 40.
- [9] 阮 彤, 孙程琳, 王昊奋, 等. 中医药知识图谱构建与应用 [J]. 医学信息学杂志, 2016, 37 (4): 8 - 13.
- [10] 李秀玲, 张树生, 黄 瑞, 等. 面向工艺重用的工艺知识图谱构建方法 [J]. 西北工业大学学报, 2019, 37 (6): 1174 - 1183.
- [11] 石文君, 刘万锁. 人工智能和知识图谱在无人机智能作战中的应用 [J]. 红外, 2020, 41 (8): 44 - 48.
- [12] WANG C, MA X, CHEN J. Information extraction and knowledge graph construction from geoscience literature [J]. Computers & geosciences, 2018 (112): 112 - 120.
- [13] LI S, WANG J, RONG J. Design-Oriented product fault knowledge graph with frequency weight based on maintenance text [J]. Advanced Engineering Informatics, 2023 (58): 22 - 29.
- [14] YAN H, YANG J, WAN J. KnowIME: A system to construct a knowledge graph for intelligent manufacturing equipment [J]. IEEE Access, 2020, 8: 41805 - 41813.
- [15] CORCOGLIONITI F, ROSPOCHER M, APROSIO P. A 2-phase frame-based knowledge extraction framework [C] //the 31st Annual ACM Symposium. ACM, 2016: 354 - 361.
- [16] 唐获音, 丁奕州, 王 轩, 等. 面向多源异构数据的航天器故障知识图谱构建方法 [J]. 空间控制技术与应用, 2023, 49 (4): 40 - 49.
- [17] 李金洪. 基于 BERT 模型的自然语言处理实战 [M]. 北京: 电子工业出版社, 2021.
- [18] 徐文博, 穆 兰, 吉丽君. 基于 ChatGPT 大语言模型的食品安全教育创新研究 [J]. 食品工业, 2024, 45 (5): 343 - 346.
- [19] 陈炫婷, 叶俊杰, 祖 璨, 等. GPT 系列大语言模型在自然语言处理任务中的鲁棒性 [J]. 计算机研究与发展, 2024, 61 (5): 1128 - 1142.
- [20] ZAYED A, DELVAUX N. Harnessing GPT-4 turbo for intelligent laboratory test recommendations: A step forward in next-generation clinical decision support [J]. Clinica Chimica Acta, 2024: 558 - 566.
- [21] YUSONG L. Research of approaches and development tools in constructing ontology [J]. Journal of Modern Information, 2009 (176): 3305 - 3308.
- [22] 潘谦红, 王 炬, 史忠植. 基于属性论的文本相似度计算 [J]. 计算机学报, 1999, 22 (6): 5 - 8.