

文章编号:1000-8055(2012)09-2147-08

# 基于自适应粒子群算法的航天测控系统 任务可靠性分配

张新贵, 武小悦

(国防科学技术大学 信息系统与管理学院, 长沙 410073)

**摘 要:** 采用测控时序图描述任务时序逻辑关系,建立了各阶段的任务可靠性模型,讨论了可靠性分配时的约束条件和分配目标,从而构建了航天测控系统任务可靠性分配模型.在此基础上,设计了自适应粒子群算法,算法在迭代过程中通过惯性权重系数  $\omega$  进行自适应调整,初始时能快速搜索全局解,后期能有效地搜索局部解.通过算例仿真,表明自适应粒子群算法在用于航天测控系统任务可靠性分配问题时具有分配结果优、收敛速度快等优点.

**关 键 词:** 可靠性分配; 航天测控系统; 任务可靠性; 自适应; 粒子群算法

**中图分类号:** V221; TP18

**文献标志码:** A

## Mission reliability allocation of spaceflight TT&C system based on adaptive particle swarm optimization

ZHANG Xin-gui, WU Xiao-yue

(College of Information Systems and Management,  
National University of Defense Technology, Changsha 410073, China)

**Abstract:** The mission sequential logic relations were described by sequential diagram, mission reliability model of each phase and the reliability allocation model were built, and the constraint conditions and the allocation objective were discussed. The adaptive particle swarm optimization (APSO) was designed, which adaptively adjusted the convergence speed by altering the value of weighted coefficient  $\omega$  during population iterating. With this method, the global results can be searched quickly in the initial stages, and the local results can be obtained effectively in the later stages. By simulation, it shows that the APSO can reach more optimal results and convergence in solving the mission reliability allocation of spaceflight TT&C (tracking telemetry and command) system.

**Key words:** reliability allocation; TT&C (tracking telemetry and command) system; mission reliability; adaptive; particle swarm optimization

航天测控 (tracking, telemetry and command, TT&C) 系统是指在航天器飞行的各个阶段对航天器及其有效载荷进行跟踪测量、监视与控制的技术系统<sup>[1]</sup>. 航天测控系统的任务可靠性

是指航天测控系统在航天器飞行的各个阶段完成规定的航天测控功能的能力. 航天测控系统的任务可靠性分配问题是基于航天测控任务把系统规定的任务可靠性指标按一定的分配方法分配给组

收稿日期:2011-09-15

网络出版地址: <http://www.cnki.net/kcms/detail/11.2297.V.20120921.1251.031.html>

基金项目:国家自然科学基金(71071159)

作者简介:张新贵(1982—),男,湖南永州人,博士生,主要从事航天装备可靠性技术研究.

成该系统的各地面站、测量船、中继卫星、控制中心、路由等设备的过程。

目前关于航天测控系统任务可靠性分配的研究比较少。文献[2]对靶场测控系统的可靠性分析框架进行了研究,在建立对应的可靠性原理框图的基础上,通过系统可靠性的 Bayes 统计推断计算得到整个测控系统可靠性水平。文献[3]把航天测控系统的一些性能参数(如多路损耗、传输噪声比等)作为指标,以航天测控系统的结构(主要是物理构成)最优化为目标,对可靠性进行了分析。但是,可靠性分配问题已经有许多成熟的算法,如遗传算法<sup>[4-5]</sup>、启发式算法<sup>[6-7]</sup>、动态规划法<sup>[8-9]</sup>等,通过研究发现遗传算法收敛速度较慢,且算法计算时间随问题规模成指数增长;启发式算法对问题依赖性较强,难以对所有问题得到较优的结果;动态规划法存在着维数“灾难”问题,并且对具有 3 个以上约束问题,求解相当困难。

航天测控系统中,任务类型多样,而且同一类型的任务可能包含多个时段。此外,各个时段涉及到的单元及可靠性结构也可能不同,同一设备也可能参与多个任务阶段。因此,航天测控系统设备间关系复杂,无显式关系式,同时,任务可靠性预计计算时间长,直接采用常规的求解算法将难以保证航天测控系统可靠性分析的计算精度和时间响应要求。为此,需要研究科学合理的任务可靠性分配方法。

粒子群算法 (particle swarm optimization, PSO)<sup>[10]</sup>是 Kennedy 和 Eberhart 在研究鸟群觅食行为的基础上于 1995 年提出的一种全局解空间智能寻优算法。该粒子群算法结构简洁,易于实现,在解决多维寻优问题中表现出搜索速度快、鲁棒性好等优点,且问题维度的增加对算法速度影响不是很大。

本文针对航天测控系统的特点,分析其任务可靠性分配的影响因素、约束条件和目标函数,采用可靠性框图建模方法建立了航天测控系统任务可靠性模型;设计了一种自适应调整种群的粒子群算法,通过算例应用表明粒子群算法对解决航天测控系统任务可靠性分配问题具有很好的效果。

## 1 航天测控系统任务可靠性分配模型

### 1.1 问题描述

为了进行可靠性分配,必须针对具体的任务建立相应的任务可靠性模型,航天测控系统的任

务可靠性模型是用来估计系统在执行任务过程中完成其规定功能的能力。航天测控系统执行一个任务可划分为多个时段,只有当所有时段任务都成功时,测控业务才算成功。可靠性框图建模方法从可靠性与任务完成的角度出发,可以反映在完成业务的条件下,与任务有关的所有单元间的相互逻辑依赖关系。

例如,设某航天器某飞行任务时间区间为  $[t_0, t_3]$ ,需要地面站设备  $S_1, S_2, S_3, S_4$  提供业务支持,其测控时序图如图 1 所示。在时间区间  $[t_0, t_1]$ ,需要  $S_3$  正常工作,且  $S_1, S_2$  至少一个正常工作;在时间区间  $[t_1, t_2]$ ,需要  $S_1$  设备正常工作,且  $S_2, S_3, S_4$  至少有一个正常工作;在时间区间  $[t_2, t_3]$ , $S_2, S_4$  至少有一个正常工作。据此可分析得到各阶段的任务可靠性逻辑框图结构模型,如图 2 所示。

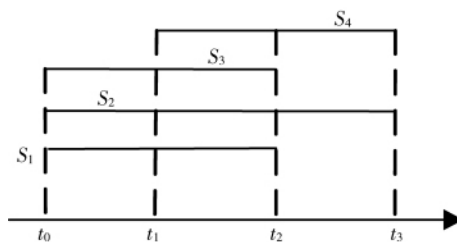


图 1 某飞行阶段测控站点跟踪时序图

Fig. 1 Sequential diagram of a TT&C mission

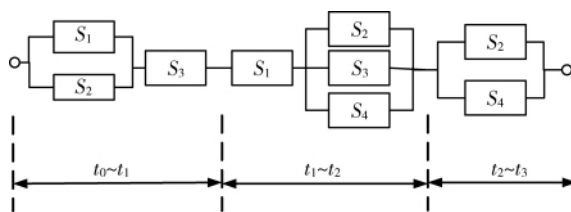


图 2 各任务阶段的任务可靠性框图模型

Fig. 2 Reliability block diagram of a TT&C system

根据可靠性框图模型,在确定各设备可靠性值的基础上,采用相应的方法可以计算出系统的可靠性指标,从而通过可靠性分配实时调整设备的可靠度值,使系统的可靠度指标满足要求。如前所述,一个具体的阶段包含的设备非常多。对于不同设备,首先具体的建设、使用和维护经费各不相同,其次建设周期也有差异,再加上技术水平和环境因素的制约,导致各具体设备可靠性水平的改变难易程度不一。设备的可靠度指标的取值只能在相应的范围内变化;分配方案应该使预计的可靠度值大于设定的目标值。

为此,航天测控系统任务可靠性分配需满足以下约束条件:

1) 航天测控系统资源受到系统复杂度、技术成熟度、任务时间、环境条件、重要度等因素的影响,导致其可靠性指标取值只能在某一个范围内变化。

2) 分配方案预计得到的系统可靠度值必须大于设定的可靠度指标。

## 1.2 分配模型

航天测控系统任务可靠性分配要保证分配完成后系统的可靠度须大于设定的可靠度指标,本文以系统可靠度的预计值作为分配结果的衡量指标。设给定的系统任务可靠性指标为  $R^*$ ,分配完成后系统的可靠度预计值为  $R$ ;第  $i$  个设备的可靠度分配值为  $R_i$ ;第  $i$  个设备工程实际中可靠度的上限指标为  $U_i$ ;第  $i$  个设备工程实际中可靠度的下限指标为  $L_i$ ;设备的个数为  $N$ 。则航天测控系统任务可靠性分配模型为

$$R = f(R_1, R_2, \dots, R_N) \quad (1)$$

$$\text{s. t. } R > R^* \quad (2)$$

$$U_i > R_i > L_i, \quad i = 1, 2, \dots, N \quad (3)$$

式(1)为目标函数:所有设备可靠度的预计值;式(2)为目标函数约束:系统的预计值必须大于设定的可靠性指标;式(3)为设备约束:每个设备的取值只能在一定的取值范围内。

对于式(1),由于航天测控系统的元素之间存在着复杂时空关联、可视与信息关联等动态约束关系,造成可靠性分配输入(即可靠性分配方案  $(R_1, R_2, \dots, R_N)$ )与输出(航天测控系统任务可靠性指标  $R$ )之间存在复杂的非线性关系,难以采用常规的数学方法进行预计求解,只能对每个可行的分配方案进行运算,计算其相应的可靠性指标,这使得航天测控系统任务可靠性分配过程十分复杂、耗时较长,严重影响了问题解决的效率,为此本文拟采用自适应粒子群算法。

## 1.3 粒子群优化算法

PSO 算法的数学描述如下:

首先,给定的解空间中随机初始化粒子群,待优化问题的变量数决定了解空间的维数。假设解空间为  $M$  维,粒子的数目为  $N$ ,第  $i$  个粒子在第  $k$  步迭代中的位置表示为  $X_i^k = (x_{i1}^k, x_{i2}^k, \dots, x_{iM}^k)$ ,位置变化率(速度)表示为  $V_i^k = (v_{i1}^k, v_{i2}^k, \dots, v_{iM}^k)$ ;第  $i$  个粒子在前  $k$  步迭代中的历史最优位置为  $P_i^k$ ;整个粒子群在前  $k$  步迭代中的历史最优

位置为  $P_g^k$ 。PSO 算法的迭代公式如下,

$$V_i^{k+1} = \omega V_i^k + c_1 r_1 (P_i^k - X_i^k) + c_2 r_2 (P_g^k - X_i^k) \quad (4)$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \quad (5)$$

$P_i^k$  和  $P_g^k$  有如下的定义:

$$P_i^k = \begin{cases} P_i^k & f(X_i^k) \geq f(P_i^k) \\ X_i^k & f(X_i^k) < f(P_i^k) \end{cases} \quad (6)$$

$$P_g^k \in \{P_0^k, P_1^k, \dots, P_N^k\} \mid f(P_g^k) = \max\{f(P_0^k), f(P_1^k), \dots, f(P_N^k)\} \quad (7)$$

其中  $\omega$  称为惯性权重,当  $\omega$  较大时适于对解空间大范围搜索,当  $\omega$  较小时适于局部寻优; $c_1$  和  $c_2$  为正常数,并分别表示粒子飞向个体最优解和全局最优解的加速权重,根据经验设定其值为  $1.0 \sim 2.0$ ,称为加速因子; $r_1$  和  $r_2$  为  $[0, 1]$  之间的随机数; $f$  表示目标函数。

在对基本 PSO 算法研究的基础上,Clerc 引入了收缩因子的概念,以保证粒子群的收敛性,PSO 算法迭代公式变为

$$V_i^{k+1} = \chi [\omega V_i^k + c_1 r_1 (P_i^k - X_i^k) + c_2 r_2 (P_g^k - X_i^k)] \quad (8)$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \quad (9)$$

其中收敛因子  $\chi = \frac{2}{|2 - \phi - \sqrt{\phi^2 - 4\phi}|}$ ,  $\phi = c_1 + c_2 > 4$ 。此时,粒子运动轨迹稳定。

## 1.4 任务可靠性的预计方法

由上述分析可知,在任务可靠性分配中,预计值与初始设定的可靠度指标的差异是决定分配效果的标准,所以可靠性预计显得十分重要。目前针对航天测控系统任务可靠性预计方法有二元决策图方法<sup>[11]</sup>、马尔可夫法<sup>[12]</sup>、离散事件仿真法<sup>[13]</sup>、限制抽样法<sup>[14]</sup>等。

由于航天测控系统在任务中出现失效的情况是稀有事件,需要大量的仿真次数才能得到既定精度的仿真结果,这非常消耗时间,而限制抽样法是一种有效提高稀有事件的仿真效率的系统可靠性仿真方法,在航天测控任务的可靠性评估中有很好的适用性,因此,本文采用该方法在分配过程中进行预计。

## 2 航天测控系统任务可靠性分配的粒子群算法

### 2.1 任务可靠性分配算法思路

航天测控系统任务可靠性分配是在已建立的任务可靠性模型的基础上,给定系统的可靠度指

标,明确相关约束条件,采用十进制编码,把系统各个设备的初始可靠度设定为初始粒子群  $X^0 = (x_1^0, x_2^0, \dots, x_M^0)$ , 然后按照式(8)和式(9)进行迭代计算,基于限制抽样得到系统的可靠度预计值,迭代中以系统可靠度的预计值与系统可靠度指标的差值作为适应度指标。

本文针对粒子群算法易早熟的不足,引入自适应调整策略,选取惯性权重系数  $\omega$  设计了自适应规模粒子群算法(adaptive scale particle swarm optimization, APSO)来解决航天测控系统任务可靠性分配问题。算法在粒子群迭代过程中,根据种群中粒子多样性的变化,对粒子进行自动增删,避免陷入局部最优解。选取惯性权重系数  $\omega$  用下式进行自适应调整:

$$\omega = \omega_{\max} - (\omega_{\max} - \omega_{\min})k/k_{\max} \quad (10)$$

式中  $\omega_{\max}$ ,  $\omega_{\min}$  分别为惯性系数的最大值和最小值,通常取  $\omega_{\max} = 0.9$ ,  $\omega_{\min} = 0.4$ ;  $k_{\max}$  表示最大迭代次数,  $k$  表示当前迭代次数。分析一下  $\omega$  的取值的意义,初始时  $k$  值小,导致  $\omega$  大,更容易搜索全局解;后面  $k$  值大,导致  $\omega$  小,更容易搜索局部解。刘自发等<sup>[15]</sup>将其应用于电力系统,取得很好的效果,表明了式(10)中  $\omega$  的选取在 PSO 算法中具有良好的性能。为此,粒子群迭代公式变为

$$V_i^{k+1} = (0.9 - 0.5k/k_{\max})V_i^k + c_1 r_1 (P_i^k - X_i^k) + c_2 r_2 (P_g^k - X_i^k) \quad (11)$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \quad (12)$$

## 2.2 任务可靠性分配算法流程

航天测控系统任务可靠性分配的具体算法流程如下:

Step 1 设置初始参数  $\omega$ ,  $c_1$ ,  $c_2$ ,  $r_1$  和  $r_2$ , 输入粒子群的初始值  $X^0 = (x_1^0, x_2^0, \dots, x_M^0)$ , 给出系统的可靠性指标  $R^*$  以及初始种群的局部最优解  $P_1$  和全局最优解  $P_g$ 。

Step 2 根据限制抽样法计算出系统的预计可靠度  $R$ , 比较  $R$  与  $R^*$  的差值,从而确定各粒子群的适应度。

Step 3 根据粒子的适应度值,更新局部最优解和全局最优解。

Step 4 按照式(11)和式(12)更新粒子群的速度和位置。

Step 5 判断迭代步数是否达到设置的迭代次数( $t > T$ ),如果达到,则转到 Step 7; 否则转到 Step 6。

Step 6 判断适应度是否满足最小误差标准

( $R - R^* < \epsilon$ ), 如果满足,则转到 Step 7; 否则转到 Step 2。

Step 7 结束。

根据上述分析,可以得到自适应规模粒子群算法流程,如图 3 所示。

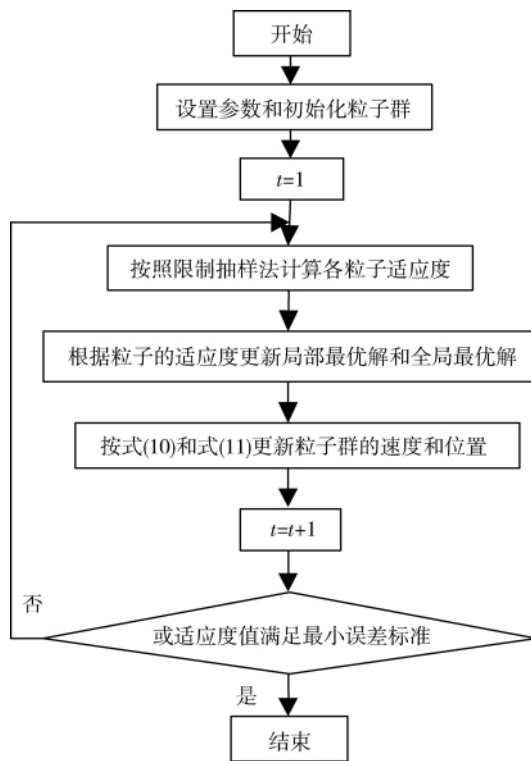


图 3 任务可靠性分配算法流程图

Fig. 3 Procedure of reliability allocation

## 3 仿真分析

针对本文研究设计了两个仿真算例:算例 1 通过对比不同的算法,证明 APSO 在解决航天测控通信系统任务可靠性分配中能够取得很好的结果;算例 2 进一步分析算法控制参数选择对结果的影响。本文采用软件 Microsoft Visual Studio 2008 c# 自行开发实验平台对仿真算例进行分析,实验平台运行的计算机硬件配置为 Pentium 4, CPU 为 3.00 GHz, 内存为 1 G。

### 3.1 仿真想定设计

为验证算法的有效性,以下选择航天测控系统某飞行阶段的某个业务为例。其任务可靠性框图模型如图 4 所示。其中  $S_{11} \sim S_{18}$  代表 8 个不同的站点,各个站点由相应的设备按照一定的逻辑关系组成,逻辑结构如表 1 所示。

其中  $S_{11} \sim S_{86}$  分别代表各个地面站完成任务所用设备、 $P_{11} \sim P_{14}$  分别代表公用的控制中

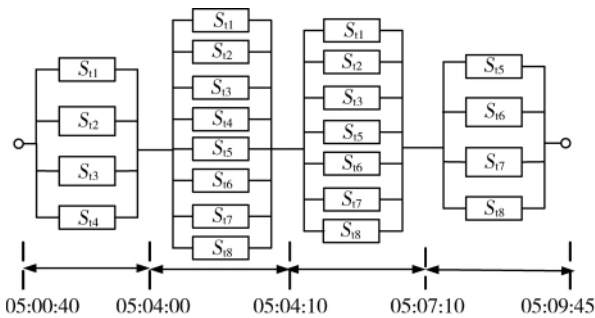


图 4 某飞行阶段的任务可靠性框图

Fig. 4 Sequential diagram of a spaceflight mission

心、通信设备和路由等. 假设系统的可靠度为  $R = 0.925$ , 各个设备的初始可靠度为  $0.5$ , 站属设备的可靠性限制条件为  $R_{Snn} > 0.8$ , 公用设备的可

靠性限制条件为  $R_{Pnn} > 0.9$ .

3.2 算例 1

本算例中最大迭代次数  $T = 40$ , 惯性权重  $\omega = 0.9 - 0.5k/k_{\max}$ , 加速常数  $c_1 = c_2 = 4$ . 对想定进行了 10 次仿真计算, 得到的分配结果如表 2 所示. 对分配效果的评价标准是分配后预计值与目标值的差距以及所需的分配时间, APSO 分配效果与启发式 (HA) 算法分配效果进行了比较, 见表 3.

由表 3 可见, APSO 计算时间比 HA 有较大幅度改善, 并且随着想定规模的增大, APSO 的优势更加明显; 虽然分配结果比 HA 稍大一点, 但结果还是非常合适.

表 1 站点设备逻辑组成表

Table 1 Logic table of equipments

| 地面站名称    | 组成设备                                     | 逻辑结构                                                                                 |
|----------|------------------------------------------|--------------------------------------------------------------------------------------|
| $S_{11}$ | $S_{11}, P_{11}, P_{12}$                 | $S_{11} \rightarrow P_{11} \rightarrow P_{12}$                                       |
| $S_{12}$ | $S_{21}, P_{11}, P_{12}$                 | $S_{21} \rightarrow P_{11} \rightarrow P_{12}$                                       |
| $S_{13}$ | $S_{31}, P_{11}, P_{12}$                 | $S_{31} \rightarrow P_{11} \rightarrow P_{12}$                                       |
| $S_{14}$ | $S_{41}, P_{11}, P_{12}$                 | $S_{31} \rightarrow P_{11} \rightarrow P_{12}$                                       |
| $S_{15}$ | $S_{51}, S_{52}, S_{53}, S_{54}, S_{55}$ | $S_{51} \rightarrow S_{52} \rightarrow S_{53} \rightarrow S_{54} \rightarrow P_{13}$ |
| $S_{16}$ | $S_{56}, P_{11}, P_{12}, P_{13}, P_{14}$ | $S_{55} \rightarrow S_{56} \rightarrow P_{14} \rightarrow P_{11} \rightarrow P_{12}$ |
| $S_{17}$ | $S_{61}, S_{62}, S_{63}, S_{64}, S_{65}$ | $S_{61} \rightarrow S_{62} \rightarrow S_{63} \rightarrow S_{64} \rightarrow P_{13}$ |
| $S_{18}$ | $S_{66}, P_{11}, P_{12}, P_{13}, P_{14}$ | $S_{65} \rightarrow S_{66} \rightarrow P_{14} \rightarrow P_{11} \rightarrow P_{12}$ |
|          | $S_{71}, S_{72}, S_{73}, S_{74}, S_{75}$ | $S_{71} \rightarrow S_{72} \rightarrow S_{73} \rightarrow S_{74} \rightarrow P_{13}$ |
|          | $S_{76}, P_{11}, P_{12}, P_{13}, P_{14}$ | $S_{75} \rightarrow S_{76} \rightarrow P_{14} \rightarrow P_{11} \rightarrow P_{12}$ |
|          | $S_{81}, S_{82}, S_{83}, S_{84}, S_{85}$ | $S_{81} \rightarrow S_{82} \rightarrow S_{83} \rightarrow S_{84} \rightarrow P_{13}$ |
|          | $S_{86}, P_{11}, P_{12}, P_{13}, P_{14}$ | $S_{85} \rightarrow S_{86} \rightarrow P_{14} \rightarrow P_{11} \rightarrow P_{12}$ |

表 2 APSO 分配结果表

Table 2 Allocation results of APSO

| 设备名称     | 第 1 次   | 第 2 次   | 第 3 次   | 第 4 次   | 第 5 次   | 第 6 次   | 第 7 次   | 第 8 次   | 第 9 次   | 第 10 次  |
|----------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| $S_{11}$ | 0.87997 | 0.90831 | 0.99999 | 0.99999 | 0.93578 | 0.94988 | 0.92233 | 0.98972 | 0.99999 | 0.86361 |
| $S_{21}$ | 0.99999 | 0.89172 | 0.99999 | 0.99999 | 0.97074 | 0.96051 | 0.82721 | 0.99999 | 0.99999 | 0.89847 |
| $S_{31}$ | 0.99709 | 0.90199 | 0.99999 | 0.91584 | 0.88261 | 0.92018 | 0.95687 | 0.98370 | 0.99999 | 0.90479 |
| $S_{41}$ | 0.99999 | 0.96935 | 0.99999 | 0.99999 | 0.96001 | 0.99999 | 0.99999 | 0.95735 | 0.99999 | 0.92269 |
| $S_{51}$ | 0.99999 | 0.92893 | 0.99999 | 0.97407 | 0.98334 | 0.99999 | 0.89360 | 0.86943 | 0.99999 | 0.86063 |
| $S_{52}$ | 0.99862 | 0.83501 | 0.99999 | 0.96504 | 0.98273 | 0.99999 | 0.87541 | 0.96751 | 0.99999 | 0.90853 |
| $S_{53}$ | 0.99999 | 0.95842 | 0.99999 | 0.99205 | 0.99999 | 0.98987 | 0.96730 | 0.96802 | 0.99999 | 0.94867 |
| $S_{54}$ | 0.99750 | 0.87764 | 0.99999 | 0.99999 | 0.99404 | 0.97480 | 0.99999 | 0.90226 | 0.99999 | 0.87893 |
| $S_{55}$ | 0.97348 | 0.82248 | 0.89542 | 0.98225 | 0.96279 | 0.89359 | 0.90924 | 0.99999 | 0.98004 | 0.95713 |
| $S_{56}$ | 0.99999 | 0.92189 | 0.92735 | 0.97111 | 0.97993 | 0.89329 | 0.92155 | 0.91117 | 0.99999 | 0.97220 |

续表

| 设备名称     | 第 1 次   | 第 2 次   | 第 3 次   | 第 4 次   | 第 5 次   | 第 6 次   | 第 7 次   | 第 8 次   | 第 9 次   | 第 10 次  |
|----------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| $S_{61}$ | 0.91293 | 0.89745 | 0.99999 | 0.83848 | 0.94047 | 0.97069 | 0.96357 | 0.95721 | 0.98026 | 0.90642 |
| $S_{62}$ | 0.97865 | 0.85655 | 0.99999 | 0.96639 | 0.95010 | 0.83248 | 0.95500 | 0.99999 | 0.99999 | 0.98532 |
| $S_{63}$ | 0.96013 | 0.98895 | 0.99999 | 0.99999 | 0.95615 | 0.99999 | 0.92607 | 0.99999 | 0.99999 | 0.93205 |
| $S_{64}$ | 0.97157 | 0.82003 | 0.99999 | 0.92438 | 0.82958 | 0.92923 | 0.99999 | 0.96641 | 0.99999 | 0.88108 |
| $S_{65}$ | 0.99718 | 0.96177 | 0.94471 | 0.99999 | 0.96466 | 0.87484 | 0.92396 | 0.86489 | 0.99999 | 0.90717 |
| $S_{66}$ | 0.96132 | 0.96441 | 0.97010 | 0.96855 | 0.99999 | 0.89125 | 0.91810 | 0.99513 | 0.99999 | 0.93148 |
| $S_{71}$ | 0.85123 | 0.89658 | 0.99999 | 0.95843 | 0.87628 | 0.93243 | 0.99999 | 0.92023 | 0.86339 | 0.87551 |
| $S_{72}$ | 0.99999 | 0.93820 | 0.85745 | 0.98987 | 0.86264 | 0.96618 | 0.94224 | 0.87807 | 0.99999 | 0.87540 |
| $S_{73}$ | 0.99999 | 0.91043 | 0.94661 | 0.99119 | 0.92945 | 0.98300 | 0.99999 | 0.99999 | 0.99999 | 0.96198 |
| $S_{74}$ | 0.99999 | 0.85405 | 0.99999 | 0.99912 | 0.95839 | 0.99139 | 0.88659 | 0.96117 | 0.99999 | 0.83534 |
| $S_{75}$ | 0.99999 | 0.94566 | 0.99999 | 0.94374 | 0.86324 | 0.98157 | 0.97725 | 0.99999 | 0.99999 | 0.86624 |
| $S_{76}$ | 0.97039 | 0.97927 | 0.99999 | 0.99999 | 0.91543 | 0.91686 | 0.99999 | 0.87390 | 0.99999 | 0.98024 |
| $S_{81}$ | 0.99065 | 0.96770 | 0.99999 | 0.98173 | 0.96736 | 0.96343 | 0.91692 | 0.99999 | 0.99999 | 0.92030 |
| $S_{82}$ | 0.99999 | 0.91429 | 0.99999 | 0.99999 | 0.87813 | 0.99999 | 0.95023 | 0.94495 | 0.99999 | 0.89680 |
| $S_{83}$ | 0.93848 | 0.96705 | 0.99999 | 0.89208 | 0.99999 | 0.95068 | 0.99647 | 0.95795 | 0.99999 | 0.81252 |
| $S_{84}$ | 0.99999 | 0.89142 | 0.97039 | 0.96861 | 0.99999 | 0.97921 | 0.92849 | 0.99999 | 0.99999 | 0.99999 |
| $S_{85}$ | 0.89070 | 0.88949 | 0.99999 | 0.99999 | 0.99999 | 0.99999 | 0.93934 | 0.99999 | 0.99999 | 0.90556 |
| $S_{86}$ | 0.99160 | 0.86205 | 0.99999 | 0.92751 | 0.87098 | 0.99999 | 0.93792 | 0.99999 | 0.92746 | 0.94621 |
| $P_{11}$ | 0.99999 | 0.95651 | 0.92643 | 0.92517 | 0.96324 | 0.96958 | 0.93089 | 0.95611 | 0.93379 | 0.96190 |
| $P_{12}$ | 0.93026 | 0.97033 | 0.99999 | 0.99999 | 0.96130 | 0.95857 | 0.99999 | 0.97487 | 0.99670 | 0.96818 |
| $P_{13}$ | 0.93864 | 0.99214 | 0.99999 | 0.93785 | 0.99446 | 0.93799 | 0.97947 | 0.95281 | 0.99414 | 0.97192 |
| $P_{14}$ | 0.93183 | 0.95330 | 0.99999 | 0.99999 | 0.98008 | 0.99999 | 0.96909 | 0.97184 | 0.98625 | 0.92412 |
| 预计值      | 0.9251  | 0.9256  | 0.9264  | 0.9271  | 0.9275  | 0.9294  | 0.92980 | 0.9302  | 0.9304  | 0.9307  |
| 时间/s     | 18.85   | 16.0625 | 19.00   | 9.34    | 19.29   | 29.61   | 15.94   | 12.81   | 16.06   | 30.39   |

表 3 分配结果对比  
Table 3 Comparison of allocation results

| APSO        |        |          | HA          |        |          |
|-------------|--------|----------|-------------|--------|----------|
| 分配后系统的可靠度指标 |        | 平均运行时间/s | 分配后系统的可靠度指标 |        | 平均运行时间/s |
| 最优值         | 最差值    |          | 最优值         | 最差值    |          |
| 0.9251      | 0.9307 | 18.85    | 0.9251      | 0.9264 | 215.49   |

选取想定某次计算结果,给出粒子群迭代过程中可靠性预计值的变化情况,如图 5 所示.由图 5 可见,APSO 在迭代过程前期的全局探索过程中,使可靠度的预计增长较大(从 0.1029 增加到 0.8276);后期局部寻优过程中,算法在保证预计值增加的同时,使得可靠度预计值与指标值最大

限度接近,最终达到近似最优解.同时,APSO 能在 40 步迭代中获得较优的结果,具有很好的收敛性.

3.3 算例 2

在仿真的基础上,调整算法控制参数进

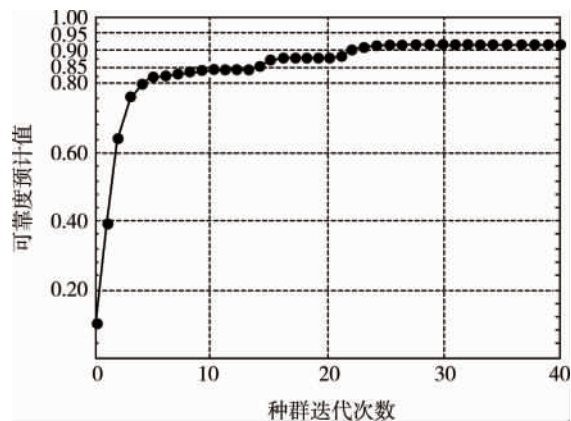


图 5 可靠度预计值变化图

Fig. 5 Relationship diagram between predicted value and population iterating times

行仿真计算,进一步分析控制参数取值对调度结果和计算时间的影响.

加速因子  $c_1$  和  $c_2$  较大时,由于收缩因子  $\chi$  的作用,可以增强 APSO 算法收敛性,并有助于局部寻优.本文中为保证粒子历史最优解和整体历史最优解对粒子更新速度影响相同,取  $c_1$  和  $c_2$  相同.当  $c_1 = c_2 > 4$  时,收缩因子  $\chi$  随加速因子增大变化较小,此时,对算法结果优劣和计算时间影响较小(如表 4 所示).因此,加速因子一般在区间  $[2, 6]$  取值即可.

表 4 加速因子对分配结果的影响

Table 4 Accelerating index effects on allocation results

| $c_1$ (或 $c_2$ ) | 可靠度预计值 | 运行时间/s |
|------------------|--------|--------|
| 2                | 0.9251 | 30.42  |
| 3                | 0.9264 | 29.64  |
| 4                | 0.9287 | 19.34  |
| 5                | 0.9315 | 15.86  |
| 6                | 0.9338 | 12.39  |

4 结 论

针对航天测控系统可靠性分配模型的特点,本文提出了一种自适应粒子群算法,算法能根据粒子群惯性权重系数  $\omega$  进行自适应调整,从而有效地搜索到全局解和局部解.通过仿真算例,验证了自适应粒子群算法在解决航天测控系统可靠性分配问题过程中,有运算速度快、鲁棒性好、分配结果质量高等优点,表明自适应粒子群算法可以很好地解决航天测控系统可靠性分配问题.

参考文献:

[1] 于志坚. 航天测控系统工程[M]. 北京:国防工业出版社,2008.  
YU Zhijian. Space TT&C system engineering[M]. Beijing:Defenses Industry Press,2008. (in Chinese)

[2] 马顺南,张士峰,刘秋辉. 靶场测控通信系统可靠性分析[J]. 飞行器测控学报,2005,24(1):11-16.  
MA Shunnan,ZHANG Shifeng,LIU Qiuhui. Reliability analysis for TT&C system of missile test range[J]. Journal of Spacecraft TT&C Technology,2005,24(1):11-16. (in Chinese)

[3] Demircioglu E,Nefes M M. Reliability-based TT&C subsystem design methodology for complex spacecraft missions[C]// Proceedings of IEEE Information Sciences and Systems Conference. Princeton: Princeton University, 2008:1268-1272.

[4] Khoshbakht A F,Raoofat M. Optimal allocation of DGs and RCSs to improve distribution network reliability and network energy loss[C]// Proceedings of 2nd IEEE International Conference on Power and Energy (PECon 08). Malaysia:Johor Baharu,Malaysia; IEEE,2008:1586-1591.

[5] Vidyarthi D P,Tripathi A K. Maximizing reliability of distributed computing system with task allocation using simple genetic algorithm[J]. Journal of System Architecture, 2001,47(6):549-554.

[6] Kartik S,Murthy C S R. Task allocation algorithms for maximizing reliability of distributed computing systems [J]. IEEE Transactions on Computers, 1997, 46(6): 719-724.

[7] Tom P A,Murthy C S R. Algorithms for reliability-oriented module allocation in distributed computing systems [J]. Journal Systems Software,1998,40(2):125-138.

[8] Yalaoui A,Chatelet E,CHU Chengbin. A new dynamic programming method for reliability & redundancy allocation in a parallel-series system[J]. IEEE Transactions on Reliability,2005,54(2):254-262.

[9] Lyu M R,Rangarajan S, Van Moorsel A P A. Optimal allocation of test resources for software reliability growth modeling in software development[J]. IEEE Transactions on reliability,2002,51(2):183-193.

[10] Kennedy J,Eberhart R. Particle swarm optimization[C]// Proceedings of IEEE International Conference on Neural Networks. Piscataway, New Jersey, USA; IEEE Service Center,1995:1492-1498.

[11] MENG Li,WANG Gang,WU Xiaoyue. Mission reliability analysis of TT&C system using BDD algorithm[C]// Proceeding of the 7th International Conference on "Mathematical Method in Reliability". Beijing:Beijing Institute of Technology Press,2011:682-686.

[12] Alam M, Al-Saggaf U M. Quantitative reliability evaluation of repairable phased-mission systems using Markov approach[J]. IEEE Transaction on Reliability, 1986, 35

- (5):498-503.
- [13] YANG Xiaosong, WU Xiaoyue. Mission reliability evaluation of TT&C system using discrete-event simulation[C] // Proceedings of the 7th International Conference on "Mathematical Method in Reliability". Beijing: Beijing Institute of Technology Press, 2011:672-676.
- [14] XU Shuangwei, WU Xiaoyue, YAN Hua. An efficient and approximate evaluation method for mission reliability of TT&C system[C] // Proceedings of the 7th International Conference on "Mathematical Method in Reliability". Beijing: Beijing Institute of Technology Press, 2011:687-692.
- [15] 刘自发, 葛少云, 余贻鑫. 基于混沌粒子群优化方法的电力系统无功最优潮流[J]. 电力系统自动化, 2005, 29(7): 53-57.
- LIU Zifa, GE Shaoyun, YU Yixin. Optimal reactive power dispatch using chaotic particle swarm optimization algorithm[J]. Automation of Electric Power Systems, 2005, 29(7):53-57. (in Chinese)