

基于 Pareto 前沿与粒子群优化的卫星资源调度算法

郑义成¹, 袁 茵², 邓 勇², 李 军¹, 王海鸿¹

(1. 91635 部队, 北京 102249; 2. 中国科学院软件研究所, 北京 102249)

摘 要: 针对多空间目标的卫星资源调度问题, 设计动态矩阵群编码方法, 在此基础上提出一种结合 Pareto 前沿与粒子群优化(PSO)的卫星资源调度算法。利用 Pareto 前沿保存一组当前最优解引导粒子群进化, 扩大搜索范围并避免陷入局部最优, 同时得到一组在不同指标上均有优势的差异化解集, 便于根据用户偏好和实时需求选择最优解。实验结果表明, 与基于传统整数编码的卫星调度算法相比, 该算法能降低粒子群进化过程中试探、判断和调整的时间消耗, 并且具有较高的资源利用率及稳定性。

关键词: 卫星调度; 多目标; 粒子群优化; 动态矩阵群; 编码

中文引用格式: 郑义成, 袁 茵, 邓 勇, 等. 基于 Pareto 前沿与粒子群优化的卫星资源调度算法[J]. 计算机工程, 2016, 42(1): 193-198.

英文引用格式: Zheng Yicheng, Yuan Yin, Deng Yong, et al. Satellite Resource Scheduling Algorithm Based on Pareto Front and Particle Swarm Optimization[J]. Computer Engineering, 2016, 42(1): 193-198.

Satellite Resource Scheduling Algorithm Based on Pareto Front and Particle Swarm Optimization

ZHENG Yicheng¹, YUAN Yin², DENG Yong², LI Jun¹, WANG Haihong¹

(1. Troop 91635, Beijing 102249, China; 2. Institute of Software, Chinese Academy of Sciences, Beijing 102249, China)

[Abstract] Aiming at the satellite resource scheduling problem of multi-space target, this paper designs Dynamic Matrix Cluster(DMC) encoding method, and proposes a satellite resource scheduling algorithm based on Pareto front and Particle Swarm Optimization(PSO). It uses Pareto front to keep a set of optimal solutions, avoids getting stuck in local optimization, also leads to more optimal solutions diverse in different index optimization priority. Then it selects the optimal solution according to user preferences and real-time requirements. Experimental results indicate that the algorithm can reduce the time consumption of temptation, judgment, and adjustment during particle swarm evolution process, and it has high resource utilization and stability compared with satellite resource scheduling algorithm based on traditional integer encoding.

[Key words] satellite scheduling; multi-object; Particle Swarm Optimization(PSO); Dynamic Matrix Cluster(DMC); encoding

DOI: 10.3969/j.issn.1000-3428.2016.01.034

1 概述

低轨卫星可应用于监视空间目标, 但由于空间目标高速飞行, 且其出现具有不可预知性, 因此卫星资源的高效调度显得尤为重要^[1]。当前关于多优化目标、多空间目标情况下的卫星资源调度研究的时间复杂度随空间目标增多迅速增长^[2]。文献[3]指出粒子群优化是一种高效的近似优化方

法。文献[4]将其引入传感器管理领域, 提出基于二进制粒子群优化的传感器管理算法。针对目前的资源调度编码方法不能有效解决资源冲突的问题, 本文提出动态矩阵群(Dynamic Matrix Cluster, DMC)编码方式, 通过在编码阶段化解资源冲突, 降低进化过程中的时间消耗。在动态矩阵群编码的基础上, 进一步提出 Pareto 前沿^[5]驱动思想改进粒子群优化算法。

作者简介: 郑义成(1976-), 男, 高级工程师、硕士, 主研方向为空间资源调度; 袁 茵, 工程师、博士; 邓 勇, 高级工程师、博士; 李 军, 高级工程师、硕士; 王海鸿, 工程师、硕士。

收稿日期: 2014-10-30 **修回日期:** 2015-02-11 **E-mail:** zyc0218@yeah.net

2 问题描述

假设每个低轨卫星携带一个传感器,监视一个空间目标需要 w 个传感器。在观测空间目标时没有干扰,并忽略不同时间段之间卫星切换的时间消耗。卫星资源调度问题描述如下:(1)有 m 个卫星,表示为集合 $S = \{S_1, S_2, \dots, S_m\}$,把星座中的卫星位置抽象为一维,使得标号相邻的卫星位置相邻;(2)有 n 个空间目标,表示为集合 $O = \{O_1, O_2, \dots, O_n\}$,使得标号相邻的目标位置相邻;(3)调度过程分为 K 个时间段,表示为集合 $T = \{T_1, T_2, \dots, T_k\}$,在第 k 个时间段中,有 n_k 个空间目标待监视, m_k 个卫星待分配;(4)调度策略表示为 $\Phi = \{\varphi(i, j, k)\}, i \in [1, m], j \in [1, n], k \in [1, K]$,其中, $\varphi(i, j, k)$ 表示在第 k 个时间段,第 i 个空间目标由第 j 个卫星监视。

优化指标为^[6]:

(1)可观测率 Q_s ,用每个时间段被监视的空间目标数除以总目标数表示,再对所有时间段求平均得到 Q_s ,该 Q_s 值越高越好。设时间段 p 内被监视的空间目标集合为 $O_p = \{O_1, O_2, \dots, O_p\}$,其成员数量为 p 。由于总的目标集合 $O = \{O_1, O_2, \dots, O_n\}$ 的成员数量为 n ,则该时间段内的可观测率为: $Q_p = \frac{p}{n}$ 。总的可观测率为: $Q_s = \frac{\sum_i Q_{total}}{t_{total}}$,其中, t_{total} 是总的时间段数。

(2)卫星切换次数 Q_η ,用所有时间段间的卫星-目标监视对切换次数求和表示,因为不同时段间的卫星-目标监视对的切换会带来不必要的能量、时间损耗,所以 Q_η 越低越好。

设在时间段 k 内的卫星-目标监视对切换次数为 Q_k ,则卫星切换次数 Q_η 为:

$$Q_\eta = \sum_1^{sum} Q_k$$

(3)松弛度 Q_μ ,用卫星工作时间的方差表示,因为均衡的工作时间会保证整个星座的稳定性,所以 Q_μ 越低越好。

设卫星工作时间的平均值是 t_{ave} ,设卫星工作时间的集合为 $T = \{t_1, t_2, \dots, t_n\}$,其工作的总时间段数为 n ,则根据方差计算公式得到:

$$Q_\mu = \frac{1}{s_{sum}} [(t_1 - t_{ave})^2 + (t_2 - t_{ave})^2 + \dots + (t_n - t_{ave})^2]$$

卫星资源调度需满足如下约束:

(1)每个目标在任意时刻需要 w 颗卫星监视。将在任意 t 时刻监视空间目标 $O_{(i-t)}$ 的卫星集合设

为 $Sat_{(i-t)} = \{S_1, S_2, \dots, S_z\}$,其中, $z \leq m$,其成员数量为 q_{sat} ,为满足该约束条件,需有 $q_{sat} \geq w$ 。

(2)每个卫星任意时刻只能监视一个目标。将在任意 t 时刻第 i 颗卫星 Sat_{it} 监视的目标集合设为 O_{it} ,其成员数量为 q_{obj} ,为满足该约束条件,需有 $q_{obj} = 1$ 。

(3)卫星只能监视其可视范围内的目标。在任意 t 时刻第 j 颗卫星 Sat_{jt} 只能监视一个目标,设被分配的监视目标为 O_α ,可视范围内的目标集合设为 $O_u = \{O_1, O_2, \dots, O_\eta\}$,为满足该约束条件,需有 $O_\alpha \in O_u$,并且 $O_u \subseteq O$ 。

(4)监视时间要长于系统要求的最短监视时间,且在监视过程中不可被打断。设任一监视过程的监视时间为 t_{suv} ,系统要求最短监视时间为 t_{min} ,为满足该约束条件,需有 $t_{suv} > t_{min}$ 。

3 动态矩阵群编码方式

传统粒子群整数编码方式如文献[7]中所述, Φ 是一个 $K \times wn$ 矩阵,行表示时间段序号,列表示目标编号,矩阵元素表示卫星编号, $\Phi[k, j] = i$ 表示在第 k 个时间段使用卫星 i 观测目标 j 。传统粒子群整数编码方式虽然简单,但是粒子群编码方式本身并没有对监视目标的数量和范围做出约束,因此,将传统粒子群整数编码方式应用于卫星监视所得结果并不一定为最优,这将会给后续粒子进化增加负担。

本文提出了满足所有约束条件的动态矩阵群编码方式。 K 个矩阵形成一个簇群来表示卫星资源调度问题的解 $\Phi = \{\phi_k\}, k = 1, 2, \dots, K$,其中, ϕ_k 是一个 $m_k \times n_k$ 的矩阵,表示第 k 个时间段的调度方案, $k \in [1, K]$ 。在该矩阵中,第 i 行代表编号为 i 的卫星 $S_i, i \in [1, m_k]$;第 $\{(w \times (j - 1) + x), x = 1, 2, \dots, w\}$ 列代表编号为 j 的空间目标;第 i 行、第 $\{(w \times (j - 1) + x), x = 1, 2, \dots, w\}$ 列相交处的元素表示第 j 个空间目标是否被第 i 个卫星监视:“1”表示“是”,任意行列中均只有一个“1”出现;“0”表示“否”;“x”表示该时段内第 j 个目标不被第 i 个卫星可见。动态矩阵群编码方式如图1所示。

0	0	x	x	0	0	1	0	x	x	0	0	0	1	0	0	0	0
0	0	0	0	0	1	0	0	0	0	1	0	x	x	0	x	0	1
1	0	x	x	0	0	0	1	0	0	x	x	0	0	1	0	0	0
0	0	0	0	1	0	x	x	0	1	0	0	1	0	x	x	x	x
0	1	x	x	x	x	x	x	1	0	x	x	0	0	0	1	x	x

(a)第1个阶段 (b)第2个阶段 (c)第3个阶段

图1 动态矩阵群编码方式

动态矩阵群编码需满足以下约束:

(1)对于第 j 个目标,第 $\{(w \times (j - 1) + x), x$

$= 1, 2, \dots, w$ 列中的数据表明该目标由哪些卫星监视。因为每一列中最多只有一个元素为“1”,且每一行中最多只有一个元素为“1”,所以有且仅有多于 w 个不同的卫星监视目标 j 。

(2) 每一行中最多只有一个元素为“1”,所以,每个卫星任意时刻只能监视一个目标。

(3) 对没有可见关系的卫星-目标对用“×”标注其对应的矩阵元素,资源分配过程中不能在“×”位置建立监视关系。

(4) 把小于标准监视时长的时段向前或向后合并。

动态矩阵群编码方法具有较好的适应性,如低轨卫星星座为获得更清晰准确的空间目标监视结果,需要使用多个卫星同时监视同一目标,此时改变 w 值即可满足此项变化。在某个卫星失效的情况下,删除各个矩阵中该卫星所代表的行即可满足此项变化。

4 Pareto 前沿驱动的粒子群优化算法

4.1 Pareto 前沿驱动思想

在粒子群优化算法中使用 Pareto 前沿引导粒子算法中各优化指标的进化是很必要的,其原因为:
 (1) 在已有的粒子群优化算法中,粒子适应度由多个优化指标共同组成,通过所有优化指标的线性或非线性加和表示,这些优化指标的权重会影响粒子群的进化方向,但却没有统一的设置标准;
 (2) 优化指标通常分布在不同维度,且单位不同,简单加和没有物理意义;
 (3) 在实际决策过程中,最优策略不一定是具有最高适应度的解,而是所有指标都处在可行域中的一个解。

在 Pareto 前沿驱动的粒子群优化过程中,每一代都保存一组粒子作为 Pareto 前沿。前沿中粒子必须满足至少有一个优化指标优于其他所有粒子的条件。在进化过程中,Pareto 前沿会向坐标系的某个角度进行进化。在进化结束后,Pareto 前沿提供一组在不同优化指标上各具优势的解决方案,决策者根据实时优化指标的优先级和可行域选取最终解决方案。

Pareto 前沿驱动的粒子群优化算法在整个进化过程中提供多个最优解,使优化算法不易错过最优解。同时粒子随机向前沿中任意一个位置运动,降低优化算法陷入局部最优解的概率。

在动态粒子群编码的基础上,使用 Pareto 前沿引导粒子群优化算法的进化过程。在每一代粒子群中用 Pareto 前沿代替全局最优粒子,通过不断更新 Pareto 前沿实现粒子群最优化,具体流程如图 2 所示。

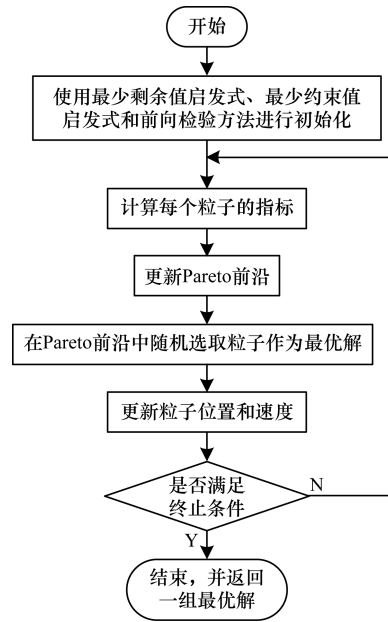


图 2 粒子群优化算法流程

4.2 位置和速度定义

位置和速度定义具体如下:

(1) 位置定义。 $\Phi = \{\phi_k\}, k = 1, 2, \dots, K$, 其中, K 个矩阵表示粒子群优化算法的解决方案,即粒子位置。为了节省存储空间, ϕ_k 分为 2 个矩阵,每个矩阵中的元素用布尔变量表示: ϕ_k^1 称为约束矩阵,若 ϕ_k^1 中“×”元素的位置和 ϕ_k 保持一致,则用“0”表示,其他元素置为“1”; ϕ_k^2 称为决策矩阵,先使其复制 ϕ_k 中所有元素,再将所有“×”元素置为“0”,其他元素保持不变。约束矩阵 $\Phi^1 = \{\phi_k^1\}, k = 1, 2, \dots, K$ 在初始化之前已经存在,并在整个进化过程中保持不变。在含有 q 个粒子的粒子群中,第 l 个粒子的位置为 $x_l = (x_{l1}, x_{l2}, \dots, x_{lk}), l = 1, 2, \dots, q, x_{lk}$ 代表 ϕ_k^2 , 决策矩阵 $\Phi^2 = \{\phi_k^2\}, k = 1, 2, \dots, K$ 。

(2) 速度定义。当粒子从一个位置移动到另一个位置,将原位置做行列变换得到新的位置。行变换和列变换是等价的,本文以行变换为例说明粒子的位置移动。如果某次随机抽取到 i_1, i_2 这 2 行进行交换, $\phi_k^2[i_1, j_1] = 1, \phi_k^2[i_2, j_2] = 1$, 这 2 行的其余元素为“0”,则交换后 $\phi_k^2[i_1, j_2] = 1, \phi_k^2[i_2, j_1] = 1$, 这 2 行的其余元素为“0”。如果在随机抽取到的 2 行中, $\phi_k^2[i_1, j_1] = 1, \phi_k^2[i_2, j_2] = 0$, 这 2 行其余元素为“0”,则交换后 $\phi_k^2[i_1, j_2] = 0, \phi_k^2[i_2, j_1] = 1$ 。行交换的前提是约束矩阵满足 $\phi_k^1[i_1, j_2] = 1, \phi_k^1[i_2, j_1] = 1$ 。在含有 q 个粒子的粒子群中,第 l 个粒子的速度为 $v_l = (v_{l1}, v_{l2}, \dots, v_{lk}), l = 1, 2, \dots, q, v_{lk}$ 表示需要在决策矩阵 ϕ_k^2 上进行 v_{lk} 次行交换才能使粒子到达下一个位置。

4.3 初始化

本文使用最少剩余值(Minimum Remaining Value, MRV)启发式、最少约束值启发式^[8-10]和前向检验方法来处理初始化问题,即先给被最少卫星可见的目标分配卫星资源,在可分配的卫星资源中,优先分配可视范围最小的卫星。每给一个目标分配卫星或分配出一个卫星,即在等待列表或资源列表中将其删除,不再参与后续分配中的排序。

4.4 Pareto 前沿计算

在粒子群更新位置后,检查每个粒子的位置以更新 Pareto 前沿。由于行列交换获得新的位置,因此可观测率 Q 保持在初始值。前沿中只保存 Q_η 或者 Q_μ 全局最优的粒子位置。如果 Pareto 前沿中的一个位置在 Q_η 和 Q_μ 上均劣于某个粒子,则将前沿中该位置删除;如果某个粒子在 Q_η 或者 Q_μ 上全局最优,则把该粒子的位置加入 Pareto 前沿。

4.5 位置和速度更新

在传统粒子群优化算法中,粒子位置按如下公式更新:

$$\mathbf{x}_{ik}^{t+1} = \mathbf{x}_{ik}^t + \mathbf{v}_{ik}^{t+1} \quad (1)$$

$$\mathbf{v}_{ik}^{t+1} = \omega \cdot \mathbf{v}_{ik}^t + c_1 \cdot \psi_1 \cdot (p_{ik}^t - \mathbf{x}_{ik}^t) + c_2 \cdot \psi_2 \cdot (p_{gk}^t - \mathbf{x}_{ik}^t) \quad (2)$$

$$\omega = \omega_{\max} - ((\omega_{\max} - \omega_{\min}) / \text{iter}_{\max}) \times \text{iter} \quad (3)$$

其中, \mathbf{x}_{ik}^t 代表第 t 代、第 k 个时间段的决策矩阵; \mathbf{v}_{ik}^t 代表对第 t 代、第 k 个时间段的决策矩阵进行 \mathbf{v}_{ik} 次有效的行互换,有效变换根据约束矩阵来确定; iter 表示当前迭代次数; iter_{\max} 表示迭代次数上限; p_{ik}^t 为全局最优位置; p_{gk}^t 是从 Pareto 前沿中随机选取的最优解; ω 为惯性权因子,控制前一时刻速度对当前速度的影响; c_1 为自身经验学习因子,经验取值为 2; c_2 为社会经验学习因子,经验取值为 2; ω 在进化初期被赋予较大的值以扩展搜索范围,随着粒子群进化接近尾声而逐渐变小,从而进行更精细的最优解挖掘,此时速度可以看成是一个绝对值逐渐变小的矢量。

在基于动态矩阵群编码的 Pareto 前沿驱动的粒子群优化算法中, \mathbf{v} 一直保持正值。如果使用式(2)对速度进行更新,会导致粒子速度发散到正无穷。根据动态矩阵群的编码方式,速度更新公式为:

$$\mathbf{v}_{ik}^{t+1} = c_1 \cdot \psi_1 \cdot (p_{ik}^t - \mathbf{x}_{ik}^t) + c_2 \cdot \psi_2 \cdot (p_{gk}^t - \mathbf{x}_{ik}^t) \quad (4)$$

其中, $\psi_1, \psi_2 \in [0, 1]$, 为随机数; $p_{ik}^t - \mathbf{x}_{ik}^t$ 表示全局最优位置 p_{ik}^t 和粒子位置 \mathbf{x}_{ik}^t 在编码矩阵中的不同行; p_{gk}^t 是从 Pareto 前沿中随机选取的最优解,用以扩大搜

索范围和防止陷入局部最优。

粒子的位置更新是用对决策矩阵做 \mathbf{v}_{ik}^t 次行交换实现的,每次做交换的 2 行在等待池中随机挑选。对于每个粒子来说有 2 个等待池, $Pool_1$ 和 $Pool_2$ 。 $Pool_1$ 中存储 ϕ_{ik}^2 所有的行, $Pool_2$ 存储 ϕ_{ik}^2 中与 p_{gk}^t 不一致的行。从 $Pool_2$ 中挑选做交换行的概率为:

$$p_2 = \text{iter} / \text{iter}_{\max} \quad (5)$$

其中, iter 表示当前迭代次数; iter_{\max} 表示迭代次数上限。在进化初期 p_2 值很小,做交换的行从 $Pool_1$ 中挑选的概率较大,可以有效地扩展搜索最优解的范围。随着粒子群进化, p_2 逐渐变大,做交换的行以越来越大的概率从 $Pool_2$ 中选取,使粒子能更精确地搜索最优解周边区域。此时速度可以看成是一个方向性越来越强的矢量,如图 3 所示。

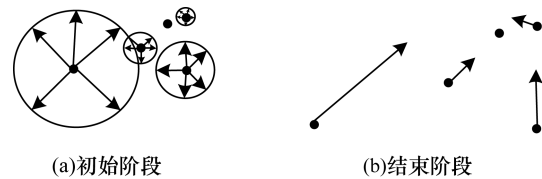


图3 速度矢量在算法初始和结束阶段的变化情况

4.6 终止条件

如果迭代周期达到最大值 iter_{\max} 或者最优值在连续若干代中没有进化,则算法终止。

5 仿真实验与结果分析

5.1 实验条件

仿真场景选取 Walker^[12] 星座对空间目标进行观测,星座参数取值为 24/3/1/1600/50。星座中共有 24 颗卫星,每颗卫星上配备一台传感器,每个目标需要 2 个卫星传感器才能完成观测。另外,含有 3 个轨道面,初始相位因子为 1,高度为 1 600 km。粒子群优化算法的参数为:粒子群规模为 50,最大迭代次数为 1 000。算法用 C++ 编程语言实现,在 Pentium 4[®] 2.6 GHz 的 CPU、2 GB 内存的平台上运行。

5.2 可观测率分析

本文实验分析比较传统整数编码的 Pareto 前沿驱动粒子群算法和 DMC 编码的 Pareto 前沿驱动粒子群算法。两者具有相同的编码方式、可观测率数据、初始化方式,并且从同一个位置开始进化。随着空间目标数的增加,可观测率随目标数的变化情况如图 4 所示。由图 4 可得,在一定的卫星数下,当空间目标数 n_k 增大到 m_k/w 时,可观测率低于 100% 且开始显著下降。

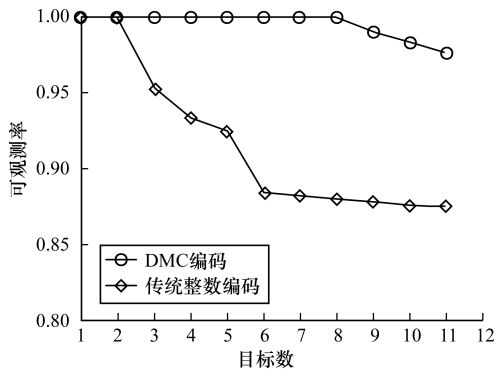


图 4 可观测率随目标数的变化

在 DMC 编码的粒子群进化过程中,因为决策矩阵行交换导致粒子位置的变化,所以可观测率保持不变。而使用传统整数编码的粒子,其运动是由对当前位置矢量做算数运算而得来的,会改变可观测率。而本文使用的动态矩阵群编码方式相对于传统整数编码方式具有可观测率优势,并且随着目标数的增加愈加明显。

5.3 松弛度和切换频率分析

在热点区域分别随机生成 3 个、6 个、11 个目标,在不同编码情况下,卫星切换次数和松弛度随时间进化。如图 5 ~ 图 7 所示,DMC 编码比传统整数编码方式具有更优的 Pareto 前沿,这是因为 DMC 编码与 Pareto 前沿有更紧密的关系。其中,首代表示多目标优化迭代的初始值;末代表示多目标优化的最后一次迭代。DMC 编码使速度在初始阶段只有大小、没有方向,极大地扩展了粒子的搜索范围,使各维度的指标均衡发展。同时,Pareto 前沿又使得在任何一个指标上有绝对优势的粒子位置都得以保存,导致更多的粒子参与竞争从而得到新的前沿。而传统整数编码通过叠加当前位置和全局最优位置的差值矢量更新位置,在 Pareto 前沿位置聚集的情况下不利于搜索空间的扩展。所以,DMC 编码能获得更优的结果。

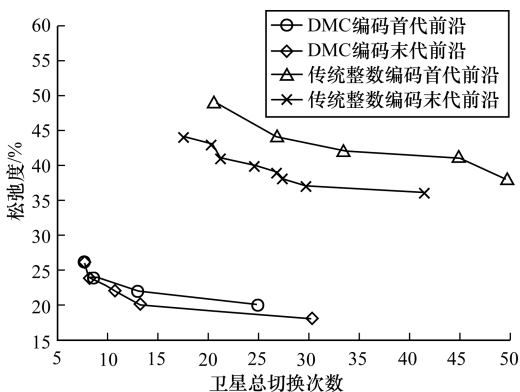


图 5 3 个目标情况下的 Pareto 前沿分布

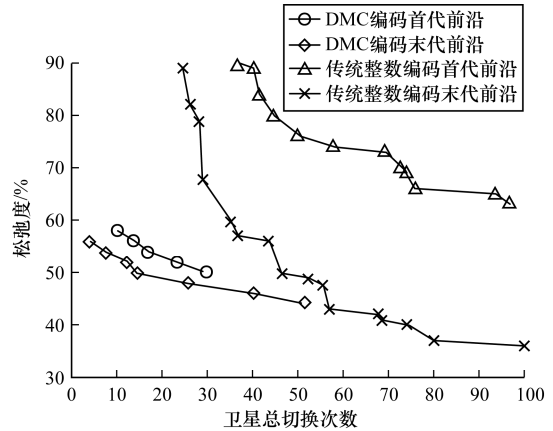


图 6 6 个目标情况下的 Pareto 前沿分布

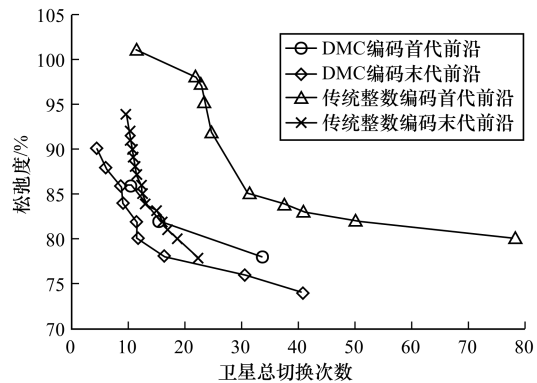


图 7 11 个目标情况下的 Pareto 前沿分布

5.4 时间复杂度分析

本文设 p 为粒子群规模、 K 为时段数、 n 为空间目标数、 ν 为粒子平均速度即行交换次数、 N_e 为每次行交换时找到合适的 2 行所需尝试的行交换次数、 $N_{attempts}$ 为给某目标配置合适的卫星所需尝试的行交换次数。在 DMC 编码下,粒子群优化算法的时间复杂度为 $O(p \cdot K \cdot \nu \cdot N_e)$,其中, $\nu \cdot N_e$ 表示每个粒子获得新位置时各矩阵所需尝试行交换次数; $p \cdot K \cdot \nu \cdot N_e$ 表示所有粒子更新位置时所需尝试行交换次数。

若某 2 行可以进行行交换的条件为:在决策矩阵中,2 行中至少有一行中含有“1”,并且交换后仍然满足约束矩阵。对既定的卫星、目标可观测率数据,则 2 行进行行交换的概率是确定值,对应的 N_e 也是常量。另外,由于速度 ν 表示需要进行的行交换次数,且每行代表一个卫星,因此 ν 的最大值即卫星个数为 24。由以上分析可知,在粒子群规模、调度时间段数一定时,DMC 编码的粒子群优化算法的时间复杂度上限是一个常数。

在传统整数编码中,粒子群优化算法的时间复杂度为 $O(p \cdot K \cdot n \cdot N_{attempts})$,其中, $n \cdot N_{attempts}$ 表示每个粒子获得新位置时各矩阵所需尝试行交换次数; $p \cdot K \cdot n \cdot N_{attempts}$ 表示所有粒子更新位置时所需

尝试行交换次数, N_{attempts} 是由系统设置的常数, 一般取值为 10~20。由此可知, 在粒子群规模、调度时间段一定时, 传统整数编码的时间复杂度随着空间目标数成线性增长。因此, DMC 编码在时间复杂度上优于传统整数编码。图 8 验证了上述理论, 其中运算时间表示粒子群平均每 500 次迭代消耗的时间。

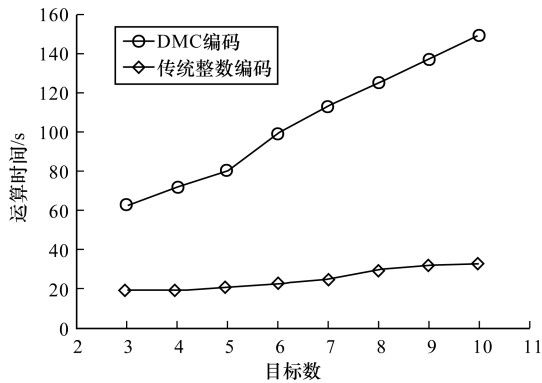


图 8 目标数不同对运算时间的影响

5.5 适应性分析

本文实验比较了 DMC 编码的 Pareto 前沿驱动粒子群算法和传统整数编码的 Pareto 前沿驱动粒子群算法, 两者具有相同的编码方式、可观测率数据、初始化方式, 并且从同一个位置开始进化。在热点区域内随机生成 3 个、6 个、11 个目标, 实验结果表 1 所示。

表 1 Pareto 前沿驱动的粒子群算法适应度对比

目标数	DMC 编码 $Fitness_1$	传统整数编码 $Fitness_2$
3	66.738 5	32.762 5
6	109.636 9	95.128 2
11	114.416 6	100.656 9

在表 1 中, DMC 编码的 Pareto 前沿驱动粒子群算法和传统整数编码的 Pareto 前沿驱动粒子群算法的适应度 $Fitness_1$, $Fitness_2$ 分别为:

$$Fitness_1 = Q_{\mu(\text{globalbest})} + Q_{\eta(\text{globalbest})} + 1/Q_{s(\text{globalbest})} \quad (6)$$

$$Fitness_2 = \max \{ Q_{\mu_k} + Q_{\eta_k} + 1/Q_{s_k}, k = 1, 2, \dots, \text{sizeof}(\text{ParetoFront}) \} \quad (7)$$

由于 DMC 编码的 Pareto 前沿驱动的单一性, 会导致粒子群搜索范围有限, 因此容易错过最优解, 同时易于陷入局部最优停止进化。DMC 编码的 Pareto 前沿驱动通过保存一组在不同维度上各具优势的最优解避免上述问题, 从而获得更优的解决方案, 具有较好的适应性。

6 结束语

本文提出一种用于多空间目标卫星调度的动态矩阵群编码方式, 解决了卫星资源冲突的问题, 并将

Pareto 前沿和粒子群优化算法相结合, 使各优化指标在不同维度分别进化, 不仅符合多目标优化的特点, 也避免了粒子陷入局部最优。仿真实验结果证明, 在典型的多目标跟踪系统中, 当粒子群规模和调度时间一定时, DMC 编码的粒子群算法以常数时间复杂度运行, Pareto 前沿驱动的粒子群算法由于在粒子群进化初期扩大了搜索范围, 并在进化后期迅速收敛, 从而获得了更优的适应度。因此, 基于动态矩阵群编码的 Pareto 前沿驱动粒子群优化算法能有效地解决多空间目标、多优化指标情况下的卫星资源调度问题, 同时为其他资源冲突、多指标优化问题提供了解决思路。

参考文献

- [1] 程洪玮, 王博, 安玮. 一种基于信息决策树的低轨星座传感器调度方法[J]. 电子学报, 2010, 38(11): 2630-2634.
- [2] 杨维, 李岐强. 粒子群优化算法综述[J]. 中国工程科学, 2004, 6(5): 87-94.
- [3] Eberhart R C, Shi Yuhui. Particle Swarm Optimization: Developments, Applications and Resources [C]// Proceedings of Congress on Evolutionary Computation. Washington D. C., USA: IEEE Press, 2001: 27-30.
- [4] Maheswararajah S, Halgamuge S. Sensor Scheduling for Target Tracking Using Particle Swarm Optimization [C]// Proceedings of VTC'06. Washington D. C., USA: IEEE Press, 2006: 7-10.
- [5] 曾三友, 蔡振华, 张青, 等. 一种评估近似 Pareto 前沿多样性的方法[J]. 软件学报, 2008, 19(6): 1301-1308.
- [6] 谢恺, 韩裕生, 薛模根, 等. 天基红外低轨星座的传感器管理方法[J]. 宇航学报, 2007, 28(5): 1331-1336.
- [7] 王博, 周一宇, 鲁建华, 等. 基于实值粒子群优化的 STSS 系统传感器管理算法研究[J]. 系统仿真学报, 2009, 21(22): 7287-7292.
- [8] 刘越畅, 姜云飞, 钱红. 基于问题结构的启发式策略在析取时态问题求解中的应用[J]. 计算机研究与发展, 2008, 45(11): 1840-1849.
- [9] Tsamardinos I, Pollack M E. Efficient Solution Techniques for Disjunctive Temporal Reasoning Problems [J]. Artificial Intelligence, 2003, 151(1/2): 43-49.
- [10] Alessandro A, Castellini C, Giunchiglia E, et al. An Open Platform for Satisfiability Modulo Theories [C]// Proceedings of PDPAR'04. Amsterdam, Holland: Elsevier Science Publishers, 2004: 25-36.
- [11] Kennedy J, Eberhart R. Particle Swarm Optimization [C]// Proceedings of IEEE International Conference Neural Networks. Washington D. C., USA: IEEE Press, 1995: 1942-1948.
- [12] Budianto I A, Olds J R. Design and Deployment of a Satellite Constellation Using Collaborative Optimization [J]. Journal of Spacecraft and Rockets, 2004, 41(6): 956-963.