

文章编号:1001-2486(2011)01-0132-06

电子侦察卫星联合侦察的混合调度算法*

黄小军,马满好,邱涤珊,祝江汉

(国防科技大学^{C1}ISR技术国防科技重点实验室,湖南长沙 410073)

摘要:针对电子侦察卫星的使用约束,及不同任务的调度需求,建立了电子侦察卫星联合侦察的多目标混合整数规划模型。利用进化算法的全局搜索能力和变邻域搜索的局部优化能力,提出了一种多目标进化算法和变邻域搜索相结合两阶段混合调度算法 MOEA-VNS。针对问题多时间窗组合优化特点,设计了进化算子与邻域移动算子,在确保解多样性的同时使算法得到的非劣解集尽可能接近问题的 Pareto 最优集。仿真实验表明,MOEA-VNS 能有效解决电子侦察卫星联合侦察调度问题。

关键词:混合整数规划;多目标进化算法;变邻域搜索;卫星调度

中图分类号:TP391 **文献标识码:**A

Solving the Cooperative Reconnaissance of Electronic Reconnaissance Satellite with a Hybrid Scheduling Algorithm

HUANG Xiao-jun, MA Man-hao, QIU Di-shan, ZHU Jiang-han

(National Key Laboratory of Information Systems Engineering, National Univ. of Defense Technology, Changsha 410073, China)

Abstract: Taking the capabilities and usage restrictions of the electronic reconnaissance satellite (ERS) as well as the scheduling requirement of different tasks into account, a multi-objective mixed integer programming model for ERS cooperative reconnaissance problem (ERSCRIP) was presented. Then a hybrid algorithm named MOEA-VNS was put forward. MOEA-VNS made full use of the outstanding global search capability of multi-objective evolutionary algorithm (MOEA) and the local search performance of variable neighborhood search (VNS). Taking advantage of the combination characteristic of multiple reconnaissance windows, the evolutionary operators and the neighborhood search operators were designed, which not only ensured the diversity of the non-dominated solutions acquired by MOEA-VNS, but also made them as close to the Pareto optimal solutions of the ERSCRIP as possible. The experiment results show that MOEA-VNS can solve the problem effectively.

Key words: mixed integer programming; multi-objective evolutionary algorithm; variable neighborhood search; satellite scheduling

电子侦察卫星 (Electronic Reconnaissance Satellite, ERS) 是用于侦收电子设施电磁辐射信号以获取情报信息的侦察卫星。ERS 可以不受地域或天气条件的限制,大范围、连续性地长期监视和跟踪敌方雷达、通信等系统的传输信号,成为现代军事情报侦察中不可或缺的侦察手段。由于卫星调度结果的优劣直接影响 ERS 资源的侦察效益,电子侦察任务调度问题成为国内外的一个研究热点。

ERS 是一类特殊的对地观测卫星,ERS 联合侦察任务调度需要考虑不同类型任务(包括普查任务与高时效应急任务)的调度需求,及卫星的存储容量、圈次工作时间、任务转换时间等约束,是一个复杂的多目标组合优化问题,且被证明为

NP—hard 问题^[1]。目前,对卫星调度问题的研究,绝大多数都集中于成像侦察卫星调度:Lin^[2] 基于单机加工问题建立了 ROCSAT-II 卫星调度模型,并提出了基于拉格朗日松弛的求解方法; Bianchessi^[3] 使用禁忌搜索及随机贪婪算法解决了 COSMO-SkyMed 星座调度问题; Soojeon^[4] 研究了海洋及通信卫星调度问题并使用任务调度算法进行了求解;郭玉华^[5] 提出了一种迭代修复求解技术用于多星联合任务规划,并通过算法比较证明了其有效性;王钧^[6] 对成像卫星调度问题建立了基于约束满足的多目标模型,并使用 SPEA2 进行求解。

上述工作主要针对卫星的使用约束,对卫星调度问题的建模与算法进行研究,为本文工作提

* 收稿日期:2010-05-11

基金项目:国家安全重大基础研究资助项目(6136101)

作者简介:黄小军(1981—),男,博士生。

供了很好的借鉴;然而面向不同类型的任务需求,其调度方法有所区别。本文针对电子侦察卫星联合侦察任务调度问题,在考虑普通任务与高时效性应急任务的不同调度需求,及卫星的工作时间、存储容量和任务转换时间等多种约束的条件下开展研究。首先建立了问题的多目标混合整数规划模型;然后提出了一种基于进化算法和变邻域搜索的混合调度算法 MOEA-VNS,根据问题的启发式信息,量身设计了用于问题求解的进化算子和邻域移动算子;最后以仿真实验验证了算法的有效性。

1 电子侦察卫星联合侦察调度问题描述

电子侦察卫星运行在高地球轨道上,并通过星载高灵敏度接收天线收集卫星覆盖范围内的电磁信号。为了提高这一稀缺资源的利用率,在一定调度周期内的多星联合任务调度,成为电子侦察卫星指挥控制过程的关键环节。由于电子侦察卫星调度问题为一类过度订购问题,仅有部分信号接收任务能够完成。我们的工作就是从用户提交的任务集中选取一个满足卫星使用约束的子集,并确定每个已选任务的执行时间,以获得最优或近似最优的收益。为准确描述电子侦察卫星联合侦察调度问题,首先定义以下符号。

1.1 集合

- $Task$: 任务集
- $UTask$: 高时效性应急任务集, $UTask \subset Task$
- Sat : 电子侦察卫星集合, $|Sat| = N$
- R^s : 电子侦察卫星 s 可执行的任务子集, $s \in Sat$
- S_i : 可执行任务 i 的卫星子集, $|S_i| = m_i$, $i \in Task$, $S_i \subseteq Sat$
- TW_s^j : 卫星 s 在第 j 圈次的侦察窗口集合, $j \in \{1, 2, \dots, |c_s|\}$, $s \in Sat$

- TW_{si}^j : 卫星 s 对任务 i 的时间窗口集合, $i \in Task$, $s \in Sat$
- c_s : 规划周期内卫星 s 的侦察圈次集合, $s \in Sat$

1.2 参数

- $Span$: 调度周期 $[t_s, t_e]$, t_s 和 t_e 分别为调度开始时间和结束时间
- a_i : 任务 i 的最早开始时间, $i \in Task$
- v_i : 任务 i 的侦察收益, $i \in Task$

- n_{si} : 卫星 s 对任务 i 的侦察窗口数量, $i \in Task$, $s \in Sat$
- TW_{si}^k : 卫星 s 对任务 i 的第 k 个时间窗口 $[stw_{si}^k, etw_{si}^k]$, 其中 stw_{si}^k 和 etw_{si}^k 分别为窗口开始和结束时间
- rdt_i^s : 卫星 s 对任务 i 的侦察时间, $i \in Task$, $s \in Sat$
- q^i : 卫星 s 执行任务 i 收集的数据量
- Q^s : 卫星 s 的存储容量, $s \in Sat$
- t_{ij}^s : 卫星 s 连续执行任务 i, j 所需的转换时间
- t_e^s : 卫星 s 单次最长开机时间, $s \in Sat$
- t_{cir}^s : 卫星 s 单圈最长工作时间

1.3 决策变量

- $x_{ij}^s = \begin{cases} 1 & \forall i, j \in Task, \text{如果卫星 } s \text{ 连续执行任务 } i, j \\ 0 & \text{其他} \end{cases}$
- $x_{is}^k = \begin{cases} 1 & \forall i \in Task, \text{如果任务 } i \text{ 在它} \\ & \text{与卫星 } s \text{ 的第 } k \text{ 个窗口执行} \\ 0 & \text{其他} \end{cases}$
- rst_i : 任务 i 的开始侦察时刻

1.4 约束

(1) 每个任务只执行一次, 额外侦察不产生任何收益

$$\sum_{k \in \{1, 2, \dots, n_{si}\}} \sum_{s \in S_i} x_{is}^k \leq 1 \quad \forall i \in Task$$

(2) 信号收集任务必须在卫星与任务的可视窗口内

$$x_{is}^k (rst_i - stw_{is}^k) \geq 0 \quad \forall s \in Sat, i \in R^s, k \in \{1, 2, \dots, n_{si}\}$$

$$x_{is}^k (rst_i + rdt_i^s - etw_{is}^k) \leq 0 \quad \forall s \in Sat, i \in R^s, k \in \{1, 2, \dots, n_{si}\}$$

(3) 任何连续执行的任务必须满足转换时间约束

$$x_{ij}^s \cdot (rst_i + rdt_i^s + t_{ij}^s - rst_j) = 0 \quad \forall s \in Sat, i, j \in R^s$$

(4) 单圈累积工作时间不能超过 t_{cir}^s

$$\sum_{k \in TW_s^j} x_{is}^k \cdot rdt_i^s \leq t_{cir}^s \quad \forall s \in Sat, i \in R^s, j \in \{1, 2, \dots, |c_s|\}$$

(5) 单次最长开机时间不能超过 t_e^s

$$\max(x_{is}^k \cdot rdt_i^s) \leq t_e^s \quad \forall s \in Sat, i \in R^s, k \in \{1, 2, \dots, n_{si}\}$$

(6)星载存储容量约束

$$\sum_{k \in TW_s^j} x_{is}^k \cdot q^i \leq Q^s \quad \forall s \in Sat, i \in R^s, \\ j \in \{1, 2, \dots, |c_s|\}$$

1.5 目标函数

卫星调度是一种过度订购问题,难以做到安排执行所有提交的任务,因此调度计划应尽量最大化任务侦察收益。此外,对于 $UTask$ 中的应急任务,由于其高时效性要求,需要尽量将其安排在最早的侦察窗口,以减少任务延迟。基于上述考虑,定义问题的优化目标函数如下。

(1)最大化侦察收益

$$\max f_1 = \sum_{k \in \{1, 2, \dots, n_{si}\}} \sum_{s \in Sat} \sum_{i \in Task} x_{is}^k \cdot v_i$$

(2)最小化应急任务平均延时

if $\exists x_{is}^k \neq 0, i \in UTask$:

$$\min f_2 = \sum_{k \in \{1, 2, \dots, n_{si}\}} \sum_{s \in Sat} \sum_{i \in UTask} x_{is}^k \cdot (rst_i - a_i) / \\ \sum_{k \in \{1, 2, \dots, n_{si}\}} \sum_{s \in Sat} \sum_{i \in UTask} x_{is}^k$$

2 MOEA-VNS 算法设计

ERS 联合侦察调度问题本质上是一个多目标、多约束的复杂多目标优化问题(Multi-Objective Optimization Problem, MOP)。多目标进化算法^[7-10](Multi-objective Evolutionary Algorithm, MOEA)在求解 MOP 问题时,无需偏好信息即可对多个目标同时进行优化,且其并行机制使算法运行一次就能得到问题的 Pareto 最优集或近似集。虽然 MOEA 具有较强的全局优化能力,但存在容易过早收敛的缺陷,当问题规模和复杂度的增大,其优化性能往往不够理想。然而基于算法混合的思想,通过不同优化技术的优势互补,已成为提高算法优化性能的一种重要且有效的途径^[11]。

本文提出了一种多目标进化算法和变邻域搜索相结合的两阶段搜索算法 MOEA-VNS,通过 MOEA 的全局搜索能力得到问题的近似 Pareto 最优集;变邻域搜索以此为输入,利用其局部优化能力实现解集的进一步优化。以下部分对模型求解的 MOEA-VNS 算法设计进行具体阐述。

2.1 MOEA 算法设计

2.1.1 染色体编码

电子侦察卫星联合侦察调度问题的数学模型为典型的混合整数规划模型。模型中共有三类决策变量: $x_{ij}^s, x_{is}^k, rst_i$, 其中 x_{ij}^s, x_{is}^k 为 0-1 变量, rst_i

为区间变量,其取值范围为卫星对任务的侦察时间窗口。根据侦察任务与侦察窗口、侦察窗口与侦察卫星的对应关系,通过确定 x_{is}^k 即可确定 x_{ij}^s , 而确定 x_{ij}^s 即可通过约束传播确定 rst_i 并进而验证 x_{ij}^s 的可行性。因此,MOEA-VNS 的染色体只需针对决策变量 x_{is}^k 进行编码。如图 1 所示,对于给定的染色体 $Ch = \{S^1, S^2, \dots, S^N, Re\}$, $S^s = \{S_1^s, S_2^s, \dots, S_{c_s}^s\}$, $s = \{1, 2, \dots, N\}$ 是卫星 s 的任务序列, $S_i^s = \{t_{i1}, t_{i2}, \dots, t_{im}\}$, $i = \{1, 2, \dots, c_s\}$, 是卫星 s 第 i 圈次任务序列, Re 是未成功调度任务集。每颗卫星的任务序列和卫星的圈次任务序列均用标志位隔开。为保证染色体的有效性,仅任务集中的任务才能出现在染色体中并且每个任务仅能出现一次。

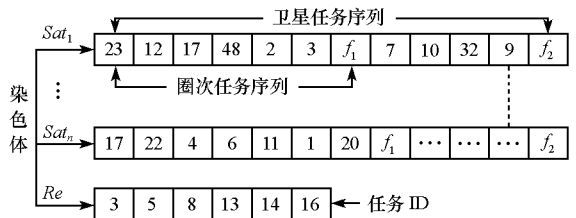


图 1 染色体数据结构

Fig.1 Data Structure of chromosome

2.1.2 初始种群构造

初始种群的构造是影响算法性能的一个重要因素。虽然完全随机生成初始解可以实现解的多样性,但会增加进化代数,进而延长进化时间;另一方面,单纯采用构造算法,虽然可以避免初始种群中不可行解太多引起的收敛过慢,但是容易导致算法陷入局部最优。因此,根据卫星与任务的对应关系及卫星使用约束,采用随机选取和构造相结合的方式产生初始种群。初始种群的随机构造流程如图 2 所示。

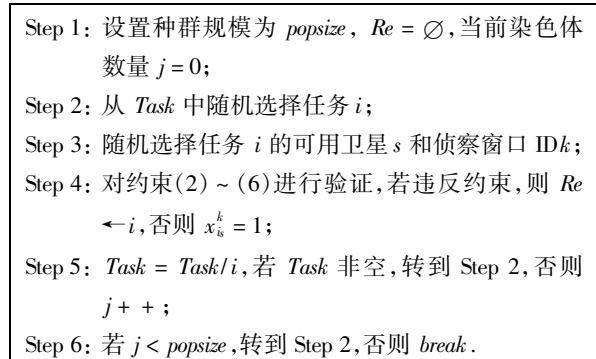


图 2 初始种群构造流程

Fig.2 Construction flow of initial population

2.1.3 交叉算子

本文提出了一种具有问题特色的序列交叉算子来搜索解空间。序列交叉算子基本操作包括染色体序列交换、冗余任务删除和缺失任务的插入。给定父染色体 $F_1 = \{S_1^1, S_1^2, \dots, S_1^N, Re_1\}$ 和 $F_2 = \{S_2^1, S_2^2, \dots, S_2^N, Re_2\}$, 随机产生非负整数 s 和 j , 其中 $s < N$ 并且 $j < c_s$ 。通过交换卫星 s 第 j 圈次安排的任务序列 S_{1j}^s 和 S_{2j}^s , 可以得到子代个体 $C_1 = \{S_1^1, S_1^2, \dots, S_{11}^s, \dots, S_{2j}^s, \dots, S_{1c_s}^s, \dots, S_1^N, Re_1\}$ 和 $C_2 = \{S_2^1, S_2^2, \dots, S_{21}^s, \dots, S_{1j}^s, \dots, S_{2c_s}^s, \dots, S_2^N, Re_2\}$ 。为了保证染色体的有效性, 构造了序列差异函数 $Dis(S_j^s, S_j^s)$ 来找出子代中的冗余任务和缺少的任务。序列差异函数计算如下:

$$Dis(S_{1j}^s, S_{2j}^s) = \{i \in Task \mid i \in S_{1j}^s, i \notin S_{2j}^s\},$$

$$Dis(S_{2j}^s, S_{1j}^s) = \{i \in Task \mid i \in S_{2j}^s, i \notin S_{1j}^s\}$$

显然, $Dis(S_{1j}^s, S_{2j}^s)$ 代表 C_1 的缺少的任务集, 而 $Dis(S_{2j}^s, S_{1j}^s)$ 是冗余任务集。这样删除操作和插入操作可对应完成。

2.1.4 变异算子

变异利于进化算法跳出搜索空间中的固定位置, 实现种群的多样性。但是变异算子的设计应当以尽量不破坏好的进化模式为前提。定义两种变异算子: 插入变异以概率 p_{in} 从 Re 中随机选择 n 个任务将其插入当前任务序列中; 移出变异以概率 p_{out} 从任务序列中随机选择 m 个任务, 将其移入 Re 中。这两种变异算子针对问题的多窗口组合特性, 能够实现尽可能多的解空间进行搜索。

2.1.5 外部种群更新逻辑

外部种群 (*Archive*) 的主要目的是用于存储历代进化过程中得到的非劣解, 并实现非劣解集的多样性。本文采用基于邻域密度的方法来实现解的多样性。对于容量为 *ArchiveSize* 的外部种群, 密度评估根据个体 S 对应的目标函数向量 $f(S)$ 评价其在外种群中的邻域密度, 并以一个量化指标 $D(S) \in R^{\oplus}$ 表示评估结果。密度评估值 $D(S)$ 越大, 表示个体的邻域密度越大。当非劣解数量超出外部种群规模, 更新逻辑将把具有最大邻域密度的解从外部种群中剔除, 以尽量保证解的多样性。外部种群更新逻辑如图 3 所示。

2.2 VNS 算法设计

2.2.1 直接插入邻域

在不改变当前调度方案任务安排的资源与顺序的前提下, 按照任务综合收益的高低顺序, 依次

判断 Re 中所有的任务能否在满足约束条件的情况下, 插入到某个资源的任务序列中, 使得各优化目标进一步改进。直接插入移动算子流程如图 4 所示。

```

Step 1: 对非劣解  $S$ , 若  $\exists S_i \in Archive$ , 且  $S_i > S$ , break;
Step 2:  $\forall S_i \in Archive$ , 若  $S_i < > S$ ;
    ◆ 如果  $Archive$  中非劣解数量小于  $ArchiveSize$ , 则  $Archive \leftarrow S$ ;
    ◆ 如果  $Archive$  中非劣解数量等于  $ArchiveSize$ , 计算  $Archive \cup \{S\}$  中个体邻域密度;
    • 如果  $D(S) \geq \max_{S_i \in Archive} D(S_i)$ , break;
    • 如果  $D(S) < \max_{S_i \in Archive} D(S_i)$ , 则剔除个体  $S_j$ ,  $D(S_j) = \max_{S_j \in Archive} D(S_j)$ , 且  $Archive \leftarrow S$ ;
Step 3: 若  $\exists S_i \in Archive$ , 使得  $S_i < S$ , 则剔除所有的  $S_i$ ,  $Archive \leftarrow S$ .
  
```

图 3 外部种群更新逻辑

Fig.3 Update logic of external archive

```

Step 1: 设置直接插入不能安排的任务集  $DT' = \emptyset$ , 待插入任务集  $Re$ , 初始解为  $S_0$ , 当前卫星已安排的任务为  $\bigcup_{s \in Sat} Q(s)$ ;
Step 2: 计算  $Re$  中各任务的综合收益  $v'_i = v_i / \sum_{s \in Sat} n_{si}$ , 并以高低顺序进行插入任务排序;
Step 3: 计算当前待插入任务  $i$  的所有插入点所在卫星的圈次负载  $L_j^s = \sum_{k \in TW_s^j, j \in c_s} x_k^k \cdot rdt_i^s / t_{ar}^s$ , 并按负载由低到高的顺序对插入点进行排序;
Step 4: 依次遍历各个插入点:
    • 若存在卫星  $s$  使得任务能直接插入, 且任务插入得到的解  $S^* > S_0$ , 则更新  $Q(s)$ ,  $Re = Re \setminus i$ ;
    • 若不存在可行的插入点, 则  $DT' \leftarrow i$ ,  $Re = Re \setminus i$ .
  
```

图 4 直接插入算子流程

Fig.4 Directly insert tasks operator

2.2.2 移位插入邻域

在不违反任何约束的情况下, 移位插入尝试把与插入任务直接冲突的任务从当前窗口移动到另一个无冲突的窗口, 从而实现任务插入。移位插入移动算子流程如图 5 所示。

2.2.3 替换插入邻域

若上述两种方法无法实现任务插入, 此时可通过替换插入的方式, 删除冲突的低收益任务, 从而提升调度方案的总体收益。替换插入移动算子流程如图 6 所示。

Step 1: 设置移位插入不能安排的任务集 $DT' = \emptyset$, 初始解为 S_0 , 搜索深度为 n , 当前卫星已安排的任务为 $\cup_{s \in Sat} Q(s)$;

Step 2: 对 $\forall i \in DT'$, 随机选择 i 的一个未处理插入点 j ;

Step 3: 记录任务 i 当前插入点的冲突任务集 $conf_i^j$, 对冲突集中的每一任务分别进行移位操作, 若存在冲突也记录各自的冲突任务, 以此递推;

Step 4: 若搜索深度内存在解 $S^* > S_0$, 则更新 $\cup_{s \in Sat} Q(s)$, 并使 $DT' = DT' / i$, 否则 $DT' \leftarrow i$.

图5 移位插入算子流程

Fig.5 Move and insert tasks operator

Step 1: 设置初始解为 S_0 , 替换插入任务集为 DT'' ;

Step 2: 对 $\forall i \in DT''$, 记录任务 i 的插入点 j 的冲突任务集 $conf_i^j$, 计算任务插入的净收益 $Val_i^j = v_i - \sum_{k \in conf_i^j} v_k$ 和应急任务延时差异 Δ_j ;

Step 3: 若存在 $Val_i^j > 0$ 且 $\Delta_j \leq 0$, 则选择此插入点进行替换插入操作;

Step 4: 更新 $\cup_{s \in Sat} Q(s)$, 并使 $DT'' = DT'' / i$.

图6 替换插入算子流程

Fig.6 Replace tasks operator

3 仿真实验

为验证算法的有效性, 设计了4个场景 Sc1、Sc2、Sc3、Sc4 开展仿真实验。任务调度周期为24h。所有场景均包含4颗电子侦察卫星, 卫星的轨道参数来自于STK的卫星数据库。4个场景的任务数量分为100个, 150个, 180个和200个, 其中应急任务数量分别为20个, 30个, 35个和40个, 普通任务的侦察收益为1, 应急任务收益为3。各任务对应的侦察目标位置均随机生成。算法使用的参数是: 种群规模100, 外部种群50, 交叉概率0.8, 两种变异概率0.05, 移位插入的搜索深度为10层。对于模型的两个优化指标, 若连续50代最优值保持不变, 则进化算法终止。所有的程序在VC++ 2003框架下完成, 并在一台 Celeron 3.2 GHZ CPU, 1G DDR RAM 的电脑上运行。

3.1 Sc1 - Sc4 计算结果

表1给出了在4个不同场景下, 采用MOEA和MOEA-VNS得到的调度结果。评价指标主要包括优化目标的最优函数值(f_1 和 f_2 的最优值并非对应同一个解)和CPU计算时间。从表1可以

看出, 在MOEA计算结果的基础上, MOEA-VNS用较小的计算开销使优化目标 f_1 平均提高了5.9%; 而对于 f_2 , 随着任务安排数量增加以及任务与卫星间的强时间窗口约束, 其最优目标值在 f_1 提高时没有得到明显改善。

表1 MOEA、MOEA-VNS得到的计算结果

Tab.1 Result of MOEA and MOEA-VNS

Sc	f_1 (min)		f_2 (min)		CPU Time(s)	
	MOEA	MOEA-VNS	MOEA	MOEA-VNS	MOEA	MOEA-VNS
Sc1	117	126	112	119	842	864
Sc2	172	185	121	121	1395	1433
Sc3	208	226	95	95	1561	1602
Sc4	236	251	124	124	1612	1665

3.2 MOEA-VNS 收敛性能

收敛性是衡量进化算法性能的一个重要方面。以场景Sc4为例, 在仿真实验中, 分别记录了进化过程中每一代的精英个体中, 目标函数 f_1 和 f_2 的最优值, 图7分别给出了两个目标函数随计算时间的进化曲线。随着进化过程的演变, 两个目标函数值都得到了优化, 并且收敛到一个相对稳定的数值。这说明MOEA-VNS能够同时有效地实现对这两个目标函数的优化。

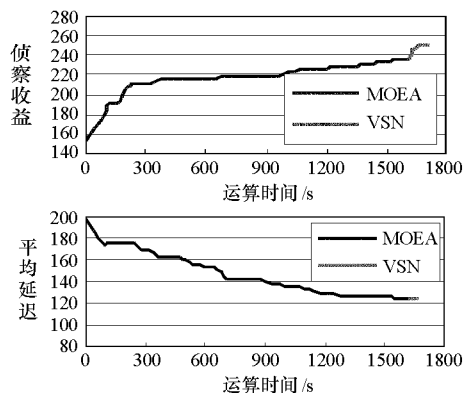


图7 Sc4中优化目标2收敛曲线

Fig.7 Evolution curve of objective f_2 in Sc4

MOEA-VNS的一个主要特点在于其致力于逼近问题的Pareto最优面, 而不是寻找一个最优解。为了验证该算法的多目标优化效果, 图8给出了进化过程中, 初始非劣解、MOEA优化解及VNS优化解的Pareto解面。从图8的结果可以看出, 在进化过程中, 非劣解面不断向性能提升的方向演化。

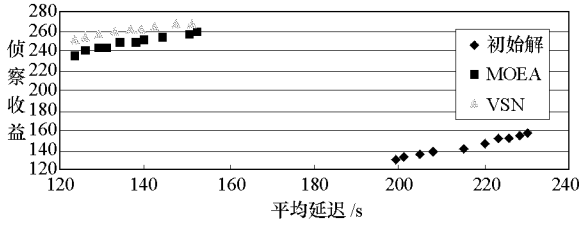


图 8 Sc4 中非劣解面进化过程

Fig.8 Evolution process of Pareto front in Sc4

4 结论

电子侦察卫星联合侦察调度问题是一类复杂的组合优化问题,随着问题规模(卫星数量、任务数量及单个任务的侦察窗口数)增加,解空间规模呈指数增长。结合卫星调度问题特点,提出了一种多目标进化算法和变邻域搜索相结合的两阶段混合搜索算法 MOEA-VNS。本文工作经过了一系列实验的检验,算法的作用和效率也得到了证明。

参考文献:

- [1] Bensana E, Verfaillie G, Bataillie N, et al. Exact and Approximate Methods for the Daily Management of an Earth Observing Satellite [C]//Proceedings of SpaceOPS, Germany: Munich, 1996.
- [2] Lin W C, et al. Daily Imaging Scheduling of an Earth Observation Satellite[J]. IEEE Transaction on Systems, Man, and Cybernetics Part A: System and Humans, 2005, 35(2): 213 - 223.
- [3] Bianchessi N, Righini G. Planning and Scheduling Algorithms for the COSMO-SkyMed Constellation [J]. Aerospace science and Technology, 2008, 12(7): 535 - 544.
- [4] Soojeon Lee. Task Scheduling Algorithm for the Communication, Ocean, and Meteorological Satellite[J]. ETRI Journal, 2008, 30(1): 1 - 12.
- [5] 郭玉华,李军,赵珂,等.多星联合任务规划的迭代修复求解技术[J].系统工程与电子技术, 2009,31(8): 1958 - 1962.
- [6] 王钧,李军,陈健,等.多目标 EOSs 联合成像调度方法[J].宇航学报, 2007,28(2): 354 - 359.
- [7] Deb K, Pratap A, Agrawal S, T. Meyarivan. A Fast and Elitist Multi-objective Genetic Algorithm: NSGA-II [J]. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182 - 197.
- [8] Zitzler E, Thiele L. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. IEEE Transactions on Evolutionary Computation, 1999, 3(4): 257 - 271.
- [9] Zitzler E, Laumanns M, L. Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm For Multiobjective Optimization[C]// Proceedings of Evolutionary Methods for Design, Optimization and Control, Barcelona, Spain, 2002, 1 - 6.
- [10] Laumanns M, Thiele L, et al. Combining Convergence and Diversity in Evolutionary Multi-Objective Optimization [C]// Evolutionary Computation, 2002, 10(3): 1 - 21.
- [11] Behnamian J, Fatemi G S M T, Zandieh M. A Multi-phase Covering Pareto-optimal Front Method to Multi-objective Scheduling in a Realistic Hybrid Flowshop Using a Hybrid Metaheuristic[J]. Expert Systems with Applications, 2009.