第 27 卷第 4 期
2010 年 4 月

控 制 理 论 与 应 用
Control Theory & Applications

Vol. 27 No. 4
Apr. 2010

文章编号: 1000−8152(2010)04−0407−08

置换表示方法求解多卫星多地面站调度问题

靳肖闪, 李 军, 王 钧, 景 宁

(国防科学技术大学 电子科学与工程学院, 湖南 长沙 410073)

摘要: 针对多卫星成像和多地面站数传并存的对地成像调度问题, 从置换空间到调度解空间的映射方法和置换空间的搜索算法两方面进行了研究. 提出了一种数传时间窗优先的置换序列映射算法, 并证明该映射算法可以将置换序列映射到调度解空间上的最优解. 提出了一种遗传随机搜索算法, 基于有记忆随机邻域搜索, 在置换空间上进行搜索. 仿真计算表明, 随机邻域搜索可以增强遗传算法的局部搜索能力, 搜索结果平均获得了4.64%的改进.

关键词: 多卫星多地面站调度; 置换表示; 遗传算法; 随机邻域搜索

中图分类号: TP79 **文献标识码**: A

On permutation-based integrated scheduling for earth observing system

JIN Xiao-shan, LI Jun, WANG Jun, JING Ning

(College of Electronics Science and Engineering, National University of Defense Technology, Changsha Hunan 410073, China)

Abstract: The integrated scheduling for earth observing system deals with multiple imaging satellites and ground stations simultaneously; it is a NP-hard oversubscribed scheduling problem involving lots of constraints. Permutation-based methods are presented to solve this complicated optimization problem. First, the integrated scheduling is expressed as permutation sequences, and a data-transfer-time-window preempted algorithm is put forward as the schedule builder, which ensures the searching for optimal schedules. Next, a genetic stochastic search method is designed to search for optimal permutations by using the permutation schedule builder and the stochastic neighborhood search algorithm with memory. Finally, the computational results on several scheduling instances show that our method enhances the local search ability for genetic search with an improvement-rate of 4.64%.

Key words: integrated scheduling for earth observing system; permutation; genetic algorithm; neighborhood search

1 引言(Introduction)

对地成像是地球观测系统的一项重要任务, 涉及到对成像卫星和数传地面站的综合调度. 对地成像包含两方面的调度需求: 一是在满足侧摆成像、存储器容量和星上电源等成像约束条件下, 为地面目标分配成像卫星; 二是在满足地面站数传约束条件下, 为成像卫星分配数传地面站, 以使星上成像数据下传到地面. 在已有研究中, Globus[1], Bianchessi[2], 李菊芳[3]和王钧[4]等对多卫星成像调度问题中的成像数据下传进行了简化, Soma[5], Marinelli[6]和Barbulescu[7]等的研究局限于卫星和地面站之间的数据传输调度. 由于卫星的可用存储器容量随成像过程而减少, 经地面站数传之后又增加, 因此成像过程与数传过程需要作为一个整体进行调度. 已有的研究难以从整体上优化利用卫星成像资源和地面站数传资源, 因此需要研究整体调度方法.

Kramer[8]指出多卫星多地面站调度是NP难解的超额订购(oversubscribed)调度问题,

编码及邻域表示方式、搜索算法是影响调度结果的两个重要因素; 置换表示方法的编码长度短, 易于处理大量约束, 适用于解决此类过度调度问题. 好的邻域结构可以改善局部搜索的质量, 交换(swap)和插入(insert)邻域具有均衡的局部搜索性能和邻域结构复杂度, 是求解类似调度问题常用的两种邻域[9]. Wolfe[10], Globus[1]和Barbulescu[7]等对卫星和地面站调度问题的研究表明, 遗传算法具有全局优化的特点, 但在局部寻优上较弱. 本文基于遗传算法的全局寻优优点, 提出了一种遗传随机搜索算法, 通过随机交换/插入邻域搜索来增强遗传算法的局部寻优能力, 在置换空间上优化多卫星多地面站调度问题.

2 多卫星多地面站调度问题(Integrated scheduling for earth observing system)

多卫星多地面站调度, 是同时为对地成像任务分

收稿日期: 2008−05−08; 收修改稿日期: 2009−05−31.
基金项目: 国家自然科学基金资助项目(60604035); 国家863高技术研究发展项目(2007AA12Z229); 国家科技支撑计划课题(2006BAG01A07).

配成像卫星和数传地面站的过程. 调度的目标是在满足各类成像约束和数传约束时, 最大化可完成的成像任务. 本文假设调度时间内每颗卫星对地面目标的过顶访问次数最大为1, 且每一个地面站对应一套数传设备. 该调度问题具体包括:

1) s颗成像卫星$S_1, S_2, \cdots, S_s$. S_i可以表示为$\langle M_i(t), E_i(t), T_i^{\rm adv}, T_i^{\rm del}, T_i^{\rm slew}, M_i^{\rm shot}, E_i^{\rm shut}, E_i^{\rm shot}, E_i^{\rm slew}\rangle$. 其中: $M_i(t)$是t时刻星上可用存储器容量, $E_i(t)$是t时刻用于成像的星上可用电能, $T_i^{\rm adv}$为成像前需要的准备时间, $T_i^{\rm del}$为成像后需要的稳定时间, $T_i^{\rm slew}$为侧摆单位角度需要的时间, $M_i^{\rm shot}$为成像时单位时间内需要的存储器容量, $E_i^{\rm shut}$为单次开关机需要的能量, $E_i^{\rm shot}$为成像时单位时间内需要的能量, $E_i^{\rm slew}$为侧摆单位角度时需要的能量.

2) g个数传地面站$G_1, G_2, \cdots, G_g$. G_i可以表示为$\langle v_i, T_i^{\rm gsa}, T_i^{\rm gst}\rangle$. 其中: v_i为等效数传倍率, $T_i^{\rm gsa}$是数传前准备时间, $T_i^{\rm gst}$是数传切换时间. $T_i^{\rm gst}$为从一颗卫星切换到另一颗卫星数传的时间, 包含了$T_i^{\rm gsa}$.

3) n个对地成像任务$J_1, J_2, \cdots, J_n$. J_i可以表示为$\langle d_i, t_i^{\rm s}, t_i^{\rm e}, p_i\rangle$. 其中: d_i是任务需要的成像时间, $t_i^{\rm s}$和$t_i^{\rm e}$是任务有效时间窗的起始和结束时间, 即在此时段内成像和数传才有效, p_i是任务的优先级.

4) 卫星S_j对J_i的成像时间窗可以表示为$\langle J_i, S_j, t_{i,j}^{\rm is}, t_{i,j}^{\rm ie}, {\rm agl}_{i,j}\rangle$. 其中: $t_{i,j}^{\rm is}$和$t_{i,j}^{\rm ie}$分别为成像起始时间和成像结束时间, $t_{i,j}^{\rm ie} - t_{i,j}^{\rm is} = d_i$; ${\rm agl}_{i,j}$为成像时的侧摆角度.

5) 卫星S_i在地面站G_j上第k次过站时的数传时间窗可以表示为$W_{i,j,k} = \langle S_i, G_j, k, t_{i,j,k}^{\rm ds}, t_{i,j,k}^{\rm de}, t_{i,j,k}\rangle$. 其中: $1 \leqslant k \leqslant K_{i,j}$, $K_{i,j}$是卫星S_i在地面站G_j上的最大过站次数; 受天线最小数传仰角限制, 数传时间窗是一个有限时段, $t_{i,j,k}^{\rm ds}$和$t_{i,j,k}^{\rm de}$分别为数传时间窗的起始时间和结束时间; $t_{i,j,k}$是该时间窗的实际开始数传时间, 由$t_{i,j,k}^{\rm ds}, T_i^{\rm gsa}, T_i^{\rm gst}$以及该地面站上一次执行的数传任务等确定.

定义调度决策变量$x_{i,j,k,l}$. $x_{i,j,k,l}=1$表示卫星S_j对J_i成像后, 在地面站G_k对卫星S_j的第l个数传时间窗上数传; 否则$x_{i,j,k,l} = 0$.

为对地成像任务分配成像卫星的过程必须满足如下的成像约束:

C1 成像有效时间约束. 成像时间不能超出对地成像任务的有效时间窗:

$$\begin{aligned}\forall\, i,j,k,l:\ &(t_i^{\rm s} - t_{i,j}^{\rm is})x_{i,j,k,l} \leqslant 0;\\ &(t_{i,j}^{\rm ie} - t_i^{\rm e})x_{i,j,k,l} \leqslant 0.\end{aligned}$$

C2 成像时序约束. 定义$F_{i,j}^{\rm para}$, 对$\forall\, i,j,k,l,m$, $i \neq j$, 如果$t_{i,m}^{\rm is} \neq t_{j,m}^{\rm is}$, 则$F_{i,j}^{\rm para} = 0$, 否则$F_{i,j}^{\rm Para} = 1$. 同一卫星不能对两个及以上的任务同时成像:

$$F_{i,j}^{\rm para}(x_{i,m,k,l} + x_{j,m,k,l}) - 1 \leqslant 0.$$

C3 成像侧摆时间约束. 定义$F_{i,j}^{\rm slew}$, 对$\forall\, i,j,k,l,m$, $i \neq j$, $t_{j,m}^{\rm is} > t_{i,m}^{\rm is}$, 如果$t_{i,m}^{\rm ie} + T_m^{\rm adv} + T_m^{\rm del} + |{\rm agl}_{i,m}|T_m^{\rm slew} + |{\rm agl}_{j,m}|T_m^{\rm slew} \leqslant t_{j,m}^{\rm is}$, 则$F_{i,j}^{\rm slew} = 0$, 否则$F_{i,j}^{\rm slew} = 1$. 卫星对一个任务成像之后, 必须有足够的时间供卫星完成侧摆动作, 以对下一个任务成像(卫星成像后首先侧摆到0°, 再侧摆以成像):

$$F_{i,j}^{\rm slew}(x_{i,m,k,l} + x_{j,m,k,l}) - 1 \leqslant 0.$$

C4 可用存储器约束. 卫星成像前必须有足够的可用存储器:

$$\forall\, i,j,k,l:(M_j^{\rm shot}(t_{i,j}^{\rm ie} - t_{i,j}^{\rm is}) - M_j t_{i,j}^{\rm is})x_{i,j,k,l} \leqslant 0.$$

C5 可用能量约束. 卫星成像前必须有足够的可用能量, 供其完成侧摆、开关机、成像等动作:

$$\begin{aligned}\forall\, i,j,k,l:[&E_j^{\rm shut} + E_j^{\rm shot}(t_{i,j}^{\rm ie} - t_{i,j}^{\rm is}) +\\ &2|{\rm agl}_{i,j}|E_j^{\rm slew} - E_j(t_{i,j}^{\rm is})]x_{i,j,k,l} \leqslant 0.\end{aligned}$$

为对地成像任务分配数传时间窗的过程必须满足如下的数传约束:

C6 数传有效时间约束. 数传时间不能超出对地成像任务的有效时间窗:

$$\begin{aligned}&\forall\, i,j,k,l:(t_i^{\rm s} - t_{j,k,l})x_{i,j,k,l} \leqslant 0;\\ &(t_{j,k,l}^{\rm de} - t_i^{\rm e})x_{i,j,k,l} \leqslant 0.\end{aligned}$$

C7 数传时间窗约束. 任一数传时间窗上执行的数传任务, 不能超过该时间窗的数传能力:

$$\forall\, j,k,l: t_{j,k,l} + \sum_{i=1}^{n} d_i x_{i,j,k,l}/v_k - t_{j,k,l}^{\rm de} + t_{j,k,l} \leqslant 0.$$

C8 数传切换约束. 同一地面站执行完一个数传时间窗上的数传任务后, 需要足够的切换时间, 以执行下一数传时间窗上的数传任务.

$$\begin{aligned}&\forall\, j,k,l,m; l \neq m; t_{j,k,l}^{\rm is} \leqslant t_{j,k,m}^{\rm is}:\\ &t_{j,k,l} + \sum_{i=1}^{n} d_i x_{i,j,k,l}/v_k + T_k^{\rm gst} - t_{j,k,m} \leqslant 0.\end{aligned}$$

为对地成像任务分配卫星及地面站的过程, 必须满足如下的联合约束条件:

C9 资源使用限制. 每个任务J_i最多只需一次成像, 一次数传:

$$\forall\, i: \sum_j \sum_k \sum_l x_{i,j,k,l} - 1 \leqslant 0.$$

C10 因果性约束. 对任务J_i成像后, 才能对J_i的成像数据进行数传:

$$\forall\, i,j,k,l:(t_{i,j}^{\rm ie} - t_{j,k,l})x_{i,j,k,l} \leqslant 0.$$

多卫星多地面站调度问题, 是在同时满足约束C1∼C10时, 最大化可完成的对地成像任务:

$$\max \sum_{i}^{n}\sum_{j}^{s}\sum_{k}^{g}\sum_{l}^{K_{j,k}} x_{i,j,k,l}p_i. \qquad (\otimes)$$

基于置换表示的调度方法求解多卫星多地面站调度问题, 重点需要解决: ①建立置换空间到调度解空间的映射算法, 使得每一置换序列对应一个可行调度, 并藉此计算置换序列的评价值(第2节); ②在置换空间上优化多卫星多地面站调度问题(第3节).

3 置换空间到调度空间的映射(Mapping from permutation space to schedule space)

置换序列是所有对地成像任务编号的一个全排列. 置换序列不能直接表示调度结果, 需要建立置换空间到调度解空间的映射算法, 根据各类约束确定每个对地成像任务是否可分配卫星和地面站, 或分配哪一个成像时间窗和数传时间窗.

首先, 为每个数传时间窗$W_{i,j,k}$设置一个序号$\mathrm{idx}^{\mathrm{win}}_{i,j,k}$, 设置原则为:

$$\forall\, i,i',j,j',k,k'\colon k<k' \Rightarrow \mathrm{idx}^{\mathrm{win}}_{i,j,k}<\mathrm{idx}^{\mathrm{win}}_{i',j',k'};$$

$$\forall\, i,i',j,j',k\colon t^{\mathrm{ds}}_{i,j,k}<t^{\mathrm{ds}}_{i',j',k} \Rightarrow \mathrm{idx}^{\mathrm{win}}_{i,j,k}<\mathrm{idx}^{\mathrm{win}}_{i',j',k}.$$

定义 1　任务置换序列 π_k : $\pi_k = \langle o_{k,1}, o_{k,2}, \cdots, o_{k,n}\rangle$, 是所有对地成像任务的一个全排列, $o_{k,i}$表示对地成像任务$o_{k,i}$在π_k中的序号为i.

定义 2　任务调度序列$\theta(\pi_k)$: $\theta(\pi_k) = \langle\langle o_{k,1}, \mathrm{sat}_{k,1}, \mathrm{gs}_{k,1}, \mathrm{win}_{k,1}\rangle, \langle o_{k,2}, \mathrm{sat}_{k,2}, \mathrm{gs}_{k,2}, \mathrm{win}_{k,2}\rangle, \cdots, \langle o_{k,n}, \mathrm{sat}_{k,n}, \mathrm{gs}_{k,n}, \mathrm{win}_{k,n}\rangle\rangle$, 是按照某种规则为$\pi_k$生成的一个调度结果. 其中: $\mathrm{sat}_{k,i} \geqslant 0$, 表示为任务$o_{k,i}$分配的成像卫星, $\mathrm{sat}_{k,i} = 0$表示未分配成像卫星; $\mathrm{gs}_{k,i}, \mathrm{win}_{k,i} \geqslant 0$, 表示为任务$o_{k,i}$分配的数传时间窗, $\mathrm{gs}_{k,i} = \mathrm{win}_{k,i} = 0$表示未分配数传时间窗. $\mathrm{sat}_{k,i}\mathrm{gs}_{k,i}\mathrm{win}_{k,i} = 0$时, $\mathrm{sat}_{k,i} = \mathrm{gs}_{k,i} = \mathrm{win}_{k,i} = 0$, 即同时取0.

定义 3　饱和调度s_k : $s_k = \langle\langle o_{k,1}, \hat{\mathrm{sat}}_{k,1}, \hat{\mathrm{gs}}_{k,1}, \hat{\mathrm{win}}_{k,1}\rangle, \langle o_{k,2}, \hat{\mathrm{sat}}_{k,2}, \hat{\mathrm{gs}}_{k,2}, \hat{\mathrm{win}}_{k,2}\rangle, \cdots, \langle o_{k,n}, \hat{\mathrm{sat}}_{k,n}, \hat{\mathrm{gs}}_{k,n}, \hat{\mathrm{win}}_{k,n}\rangle\rangle$, 其含义与$\theta(\pi_k)$类似, 且满足:

当$\hat{\mathrm{sat}}_{k,i}\hat{\mathrm{gs}}_{k,i}\hat{\mathrm{win}}_{k,i} > 0, \hat{\mathrm{sat}}_{k,j}\hat{\mathrm{gs}}_{k,j}\hat{\mathrm{win}}_{k,j} > 0$时,

$$i<j \Rightarrow \mathrm{idx}^{\mathrm{win}}_{\hat{\mathrm{sat}}_{k,i},\hat{\mathrm{gs}}_{k,i},\hat{\mathrm{win}}_{k,i}} < \mathrm{idx}^{\mathrm{win}}_{\hat{\mathrm{sat}}_{k,j},\hat{\mathrm{gs}}_{k,j},\hat{\mathrm{win}}_{k,j}};$$

当$\hat{\mathrm{sat}}_{k,i}\hat{\mathrm{gs}}_{k,i}\hat{\mathrm{win}}_{k,i} = 0$时,

$$\hat{\mathrm{sat}}_{k,j}\hat{\mathrm{gs}}_{k,j}\hat{\mathrm{win}}_{k,j} > 0 \Rightarrow j<i.$$

对$\forall\, i,j$, 当$i<j, \hat{\mathrm{sat}}_{k,i}\hat{\mathrm{gs}}_{k,i}\hat{\mathrm{win}}_{k,i} > 0$时, 如果令$\langle\hat{\mathrm{sat}}_{k,j}, \hat{\mathrm{gs}}_{k,j}, \hat{\mathrm{win}}_{k,j}\rangle = \langle\hat{\mathrm{sat}}_{k,i}, \hat{\mathrm{gs}}_{k,i}, \hat{\mathrm{win}}_{k,i}\rangle$, 则违反成像约束, 就称$s_k$是饱和调度.

根据定义, 可知饱和调度中无法为未安排成像的任务分配数传时间窗, 因此是处于可行解和不可行解边界上的可行调度. 可行调度集合中的非饱和调度, 总可以为其中部分未安排成像的任务分配数传时间窗. 因此, 以饱和调度集合组成搜索空间, 可使搜索算法获得更高的搜索效率.

定义 4　置换序列调度算子$\Lambda(\cdot)$: 将卫星和地面站分配给置换序列的操作, 即$\Lambda(\pi_k) \to \theta(\pi_k)$.

定义 5　置换序列评价值函数fitness(·): 是对任务置换序列施加调度操作$\Lambda(\cdot)$后, 所得调度结果的评价值, 与调度模型的目标函数(⊗)对应.

定义 6　调度序列距离函数:

$$d(\theta(\pi_i),\theta(\pi_j)) = \sum_{k=1}^{n}\sum_{l=1}^{n}\mathrm{differ}(\langle\mathrm{sat}_{i,k},\mathrm{gs}_{i,k},\mathrm{win}_{i,k}\rangle, \langle\mathrm{sat}_{j,l},\mathrm{gs}_{j,l},\mathrm{win}_{j,l}\rangle),$$

用以度量两个调度序列的相似度. 当$o_{i,k}=o_{j,l}, \mathrm{sat}_{i,k}=\mathrm{sat}_{j,l}, \mathrm{gs}_{i,k}=\mathrm{gs}_{j,l}, \mathrm{win}_{i,k}=\mathrm{win}_{j,l}$时,

$$\mathrm{differ}(\langle\mathrm{sat}_{i,k},\mathrm{gs}_{i,k},\mathrm{win}_{i,k}\rangle,\langle\mathrm{sat}_{j,l},\mathrm{gs}_{j,l},\mathrm{win}_{j,l}\rangle)=0;$$

否则

$$\mathrm{differ}(\langle\mathrm{sat}_{i,k},\mathrm{gs}_{i,k},\mathrm{win}_{i,k}\rangle,\langle\mathrm{sat}_{j,l},\mathrm{gs}_{j,l},\mathrm{win}_{j,l}\rangle)=1.$$

定义 7　约束检测函数 $\mathrm{check}(\theta(\pi_k))$: 如果$\theta(\pi_k)$满足所有的约束, 则$\mathrm{check}(\theta(\pi_k)) = 1$; 否则, $\mathrm{check}(\theta(\pi_k)) = 0$. 在进行约束检测前, 更新每个数传时间窗的实际开始数传时间.

引理 1　存在饱和调度与最优调度$\theta(\pi^*_k)$具有相同评价值.

证　将最优调度$\theta(\pi^*_k)$按照饱和调度的定义方式重排序, 生成具有一定顺序的调度序列

$$\theta(\pi^*_k) = \langle\langle o^*_{k,1}, \mathrm{sat}^*_{k,1}, \mathrm{gs}^*_{k,1}, \mathrm{win}^*_{k,1}\rangle, \langle o^*_{k,2}, \mathrm{sat}^*_{k,2}, \mathrm{gs}^*_{k,2}, \mathrm{win}^*_{k,2}\rangle, \cdots, \langle o^*_{kn}, \mathrm{sat}^*_{k,n}, \mathrm{gs}^*_{k,n}, \mathrm{win}^*_{k,n}\rangle\rangle,$$

其对应的置换序列为$\pi^*_k = \langle o^*_{k,1}, o^*_{k,2}, \cdots, o^*_{k,n}\rangle$. 假设$\langle\mathrm{sat}^*_{k,i}, \mathrm{gs}^*_{k,i}, \mathrm{win}^*_{k,i}\rangle$还可以容纳任务$o^*_{k,j}(j > i$且$\langle\mathrm{sat}^*_{k,j}, \mathrm{gs}^*_{k,j}, \mathrm{win}^*_{k,j}\rangle \neq \langle\mathrm{sat}^*_{k,i}, \mathrm{gs}^*_{k,i}, \mathrm{win}^*_{k,i}\rangle)$, 则令

$$\mathrm{sat}^*_{k,j} = \mathrm{sat}^*_{k,i}, \mathrm{gs}^*_{k,j} = \mathrm{gs}^*_{k,i}, \mathrm{win}^*_{k,j} = \mathrm{win}^*_{k,i},$$

且不改变其他任务分配的卫星与地面站. 由于最优调度是可行调度, 则修改后的调度仍然是可行调度. 如果$\mathrm{sat}^*_{k,j} = \mathrm{gs}^*_{k,j} = \mathrm{win}^*_{k,j} = 0$, 这说明修改后的调度结果优于最优调度, 显然是不可能的, 所以必然有$\mathrm{sat}^*_{k,j} > 0, \mathrm{gs}^*_{k,j} > 0, \mathrm{win}^*_{k,j} > 0$. 按照以上方法, 使得所有的$\langle\mathrm{sat}^*_{k,i}, \mathrm{gs}^*_{k,i}, \mathrm{win}^*_{k,i}\rangle$不能容纳$j > i$的任务, 从而可将最优调度转换为饱和调度. 由于转换的过程不改变调度结果的评价值, 则转换后的饱和调度必然与$\theta(\pi^*_k)$有相同的评价值.　　证毕.

引理1说明 饱和调度集合中存在最优调度. 只要将置换序列映射为饱和调度, 即可保证置换空间上的搜索算法从理论上可以搜索到最优调度.

多卫星多地面站调度问题同时涉及多卫星之间的成像任务分配和多地面站之间的数传任务分配, 已有的研究[1,7]还缺乏此类调度问题的映射算法. 本文提出了一种数传时间窗优先的置换序列映射算法(算法1), 建立了从置换空间到饱和调度空间的完全映射关系. 该算法的基本思想是根据饱和调度的定义原则, 按照任务置换序列中任务的出现顺序, 依次同时为各任务分配成像时间窗和数传时间窗, 算法步骤描述如下:

算法 1 输入: 置换序列π_h;输出: 调度序列$\theta(\pi_h)$.

for $i=1\leftarrow s$

for $j=1\leftarrow g$

for $k=1\leftarrow k_{i,j}$

$t_{i,j,k}=t^{\rm ds}_{i,j,k}+T^{\rm gsa}_k$ //初始化实际开始数传时间

for $i=1\leftarrow n$

let $\mathrm{sat}_{h,i}=0,\mathrm{gs}_{h,i}=0,\mathrm{win}_{h,i}=0$ //初始化每个任务待分配的对象

for $k=1\leftarrow\max(K_{i,j})$ //按照过站次数分配成像

for each i,j in $\mathrm{idx}^{\rm win}_{i,j,k}$ //按照$\mathrm{idx}^{\rm win}_{i,j,k}$的降序顺序

for $l=1\leftarrow n$

if $(\mathrm{sat}_{h,l}=\mathrm{gs}_{h,l}=\mathrm{win}_{h,l}=0)$ //未分配时间窗

let $\mathrm{sat}_{h,l}=i,\mathrm{gs}_{h,l}=j,\mathrm{win}_{h,l}=k$ //尝试分配

if $(\mathrm{check}(\theta(\pi_h))==0)$ //如果违反约束

let $\mathrm{sat}_{h,l}=0,\mathrm{gs}_{h,l}=0,\mathrm{win}_{h,l}=0$ //撤销分配的时间窗

else

for each $W_{i',j,k'}\neq W_{i,j,k}$

if $(t^{\rm ds}_{i',j,k'}\geqslant t^{\rm ds}_{i,j,k},t^{\rm ds}_{i',j,k'}<t^{\rm de}_{i,j,k})$

let $t_{i',j,k'}=\max(t_{i',j,k'},t_{i,j,k}+\sum\limits_{m=1}^{n}v_jd_{o_{h,m|\mathrm{sat}_{h,m}=i,\mathrm{gs}_{h,m}=j,\mathrm{win}_{h,m}=k}})$

输出$\theta(\pi_h)$.

算法1中, 每次为一个待成像任务分配成像时间窗和数传时间窗后, 更新每个数传时间窗的实际开始数传时间, 以进行后续待处理对地成像任务的成像时间窗和数传时间窗分配.

引理 2 数传时间窗优先置换序列映射算法, 所生成的任务调度序列是饱和调度.

证 数传时间窗优先置换序列映射算法中, 为对地成像任务分配成像时间窗和数传时间窗的过程符合饱和调度的定义, 因此生成的任务调度序列必然是饱和调度. 证毕.

引理 3 任意一种饱和调度均可由数传时间窗优先置换序列映射算法对某任务置换序列实施调度操作而得到.

证 对任意的饱和调度

$$\begin{aligned}s_k=\langle\langle&o_{k,1},\hat{\mathrm{sat}}_{k,1},\hat{\mathrm{gs}}_{k,1},\hat{\mathrm{win}}_{k,1}\rangle,\\&\langle o_{k,2},\hat{\mathrm{sat}}_{k,2},\hat{\mathrm{gs}}_{k,2},\hat{\mathrm{win}}_{k,2}\rangle,\cdots,\\&\langle o_{k,n},\hat{\mathrm{sat}}_{k,n},\hat{\mathrm{gs}}_{k,n},\hat{\mathrm{win}}_{k,n}\rangle\rangle,\end{aligned}$$

将该饱和调度对应的置换序列, 按照各个任务分配的成像时间窗和数据传输时间窗重新排序, 排序的原则是当$\hat{\mathrm{sat}}_{k,i}>0,\hat{\mathrm{gs}}_{k,i}>0,\hat{\mathrm{win}}_{k,i}>0$时, 根据$\mathrm{idx}^{\rm win}_{\hat{\mathrm{sat}}_{k,i},\hat{\mathrm{gs}}_{k,i},\hat{\mathrm{win}}_{k,i}}$的取值进行升序排序, $\hat{\mathrm{sat}}_{k,i}=\hat{\mathrm{gs}}_{k,i}=\hat{\mathrm{win}}_{k,i}=0$的任务排在最后, 从而生成具有一定顺序的置换序列$\pi_k=\langle o'_{k,1},o'_{k,2},\cdots,o'_{k,n}\rangle$. π_k对应的饱和调度为

$$\begin{aligned}s'_k=\langle\langle&o'_{k,1},\hat{\mathrm{sat}}'_{k,1},\hat{\mathrm{gs}}'_{k,1},\hat{\mathrm{win}}'_{k,1}\rangle,\\&\langle o'_{k,2},\hat{\mathrm{sat}}'_{k,2},\hat{\mathrm{gs}}'_{k,2},\hat{\mathrm{win}}'_{k,2}\rangle,\cdots,\\&\langle o'_{k,n},\hat{\mathrm{sat}}'_{k,n},\hat{\mathrm{gs}}'_{k,n},\hat{\mathrm{win}}'_{k,n}\rangle\rangle.\end{aligned}$$

对π_k施加时间窗优先置换序列映射操作后的调度序列为

$$\begin{aligned}\theta(\pi_k)=\langle\langle&o'_{k,1},\mathrm{sat}_{k,1},\mathrm{gs}_{k,1},\mathrm{win}_{k,1}\rangle,\\&\langle o'_{k,2},\mathrm{sat}_{k,2},\mathrm{gs}_{k,2},\mathrm{win}_{k,2}\rangle,\cdots,\\&\langle o'_{k,n},\mathrm{sat}_{k,n},\mathrm{gs}_{k,n},\mathrm{win}_{k,n}\rangle\rangle.\end{aligned}$$

由于饱和调度是可行调度, 则数传时间窗优先置换序列映射算法为每个任务$o'_{k,i}$所分配的成像时间窗和数传时间窗, 必有: 当$\hat{\mathrm{sat}}'_{k,i}>0,\hat{\mathrm{gs}}'_{k,i}>0,\hat{\mathrm{win}}'_{k,i}>0$时, 必然有$\mathrm{sat}_{k,i}>0,\mathrm{gs}_{k,i}>0,\mathrm{win}_{k,i}>0$, 且$\mathrm{idx}^{\rm win}_{\hat{\mathrm{sat}}'_{k,i},\hat{\mathrm{gs}}'_{k,i},\hat{\mathrm{win}}'_{k,i}}\geqslant\mathrm{idx}^{\rm win}_{\mathrm{sat}_{k,i},\mathrm{gs}_{k,i},\mathrm{win}_{k,i}}$. 又因为$s'_k$是饱和调度, 则当$\mathrm{sat}_{k,i}>0,\mathrm{gs}_{k,i}>0$及$\mathrm{win}_{k,i}>0$时, 有$\hat{\mathrm{sat}}'_{k,i}>0,\hat{\mathrm{gs}}'_{k,i}>0$和$\hat{\mathrm{win}}'_{k,i}>0$, 且

$$\mathrm{idx}^{\rm win}_{\hat{\mathrm{sat}}'_{k,i},\hat{\mathrm{gs}}'_{k,i},\hat{\mathrm{win}}'_{k,i}}\leqslant\mathrm{idx}^{\rm win}_{\mathrm{sat}_{k,i},\mathrm{gs}_{k,i},\mathrm{win}_{k,i}}.$$

因此必然有, $\mathrm{sat}_{k,i}>0,\mathrm{gs}_{k,i}>0$及$\mathrm{win}_{k,i}>0\Leftrightarrow\hat{\mathrm{sat}}'_{k,i}>0,\hat{\mathrm{gs}}'_{k,i}>0$和$\hat{\mathrm{win}}'_{k,i}>0$, 以及$\mathrm{sat}_{k,i}=\mathrm{gs}_{k,i}=\mathrm{win}_{k,i}=0\Leftrightarrow\hat{\mathrm{sat}}'_{k,i}=\hat{\mathrm{gs}}'_{k,i}=\hat{\mathrm{win}}'_{k,i}=0$, 并且已分配的成像时间窗和数传时间窗满足

$$\mathrm{idx}^{\rm win}_{\mathrm{sat}_{k,i},\mathrm{gs}_{k,i},\mathrm{win}_{k,i}}=\mathrm{idx}^{\rm win}_{\hat{sat}'_{k,i},\hat{\mathrm{gs}}'_{k,i},\hat{\mathrm{win}}'_{k,i}}.$$

即对$\forall\,i$, 有

$$\langle\mathrm{sat}_{k,i},\mathrm{gs}_{k,i},\mathrm{win}_{k,i}\rangle=\langle\hat{\mathrm{sat}}'_{k,i},\hat{\mathrm{gs}}'_{k,i},\hat{\mathrm{win}}'_{k,i}\rangle,$$

从而任意一个饱和调度均可由时间窗优先置换序列映射算法施加于某一置换序列而得到. 证毕.

定理 1　数传时间窗优先置换序列映射算法对应的调度算子$\Lambda(\cdot)$, 建立了任务置换序列集合到饱和调度集合S的多对一映射.

证　分配相同成像时间窗和数传时间窗的任务间任意改变顺序, 都不会改变数传时间窗优先置换序列映射算法为各个任务所分配的成像时间窗和数传时间窗. 意即, 有不少于一个置换序列, 经数传时间窗优先置换序列映射算法处理后, 生成的调度结果与特定饱和调度相同. 结合引理2和3可知, 数传时间窗优先置换序列映射算法的调度算子$\Lambda(\cdot)$, 建立了任务置换序列集合到饱和调度集合的多对一映射.　　证毕.

定理1说明　基于数传时间窗优先置换序列映射算法计算置换序列的评价值, 确定置换空间搜索算法的搜索方向, 则当该搜索算法收敛到某置换序列时, 根据该置换序列可以生成理论上的最优调度.

4　置换搜索(Search optimal permutations)

4.1　邻域搜索算法(Neighborhood search method)

假设π和π'是长度均为n的置换序列, $\pi(k)$是π的第k个元素, 交换邻域和插入邻域的定义为[9]:

1) π'是π在i和j两点上的交换邻域, 记作$\pi'=\mathrm{swap}(\pi,i,j)$, 且$\pi'(k)=\begin{cases}\pi(k), & k\neq i,k\neq j,\\ \pi(j), & k=i,\\ \pi(i), & k=j.\end{cases}$

因为$\mathrm{swap}(\pi,i,j)=\mathrm{swap}(\pi,j,i)$, 所以$\pi$有$(n^2-n)/2$个不同的交换邻域解.

2) π'是π上第i个元素插入位置j的插入邻域, 记作$\pi'=\mathrm{insert}(\pi,i,j)$, 且

a)

$$i<j:\pi'(k)=\begin{cases}\pi(k), & k<i\text{或者}k>j,\\ \pi(k+1), & i\leqslant k<j,\\ \pi(i), & k=j;\end{cases}$$

b)

$$i>j:\pi'(k)=\begin{cases}\pi(k), & k<j\text{或者}k>i,\\ \pi(k-1), & j<k\leqslant i,\\ \pi(i), & k=j.\end{cases}$$

因为$\mathrm{insert}(\pi,i,i+1)=\mathrm{insert}(\pi,i+1,i)$, 所以$\pi$有$(n-1)^2$个不同的插入邻域解. 因此交换邻域和插入邻域具有$O(n^2)$的邻域结构复杂度.

传统的邻域搜索一般采取两种具有确定搜索顺序的邻域搜索策略[11]: ①最优邻域搜索策略(best neighborhood search, BNS), 从当前解的所有邻域中选择最好的解; ②就近邻域搜索策略(fastest neighborhood search, FNS), 按照一定搜索顺序, 从邻域中就近选择优于当前解的解. 由于交换和插入具有$O(n^2)$的邻域结构复杂度, 所以在调度问题的对地成像任务数量n较大时, 最优邻域选择策略将因为迭代耗时过长而变得不可取.

随机邻域搜索(stochastic neighborhood search, SNS)从邻域中随机选择更好的解替代当前解. 由于随机邻域搜索的随机特性, 在当前解的邻域搜索过程中可能会出现重复的邻域搜索. 本文提出一种有记忆的随机邻域搜索(stochastic neighborhood search with memory, MSNS)算法(算法2), 为随机邻域搜索增加记忆功能, 避免在当前解的邻域搜索过程中的重复搜索, 以提高搜索效率. MSNS算法描述如下:

算法 2　输入: 给定置换序列π_0, 最大迭代次数maxStep; 输出: π_0的优化解.

初始解为$\pi_1=\pi_0$, π_1的邻域解集为$N(\pi_k)$, π_k的邻域解集记录$\mathrm{NSH}=\varnothing$.

for $k=1\leftarrow$ maxStep

$\pi'_k\leftarrow\mathrm{Randomselect}(N(\pi_k)\backslash\mathrm{NSH})$

//从$N(\pi_k)\backslash$NSH中随机选择

if (fitness(π'_k) $\geqslant$ fitness (π_k))//邻域替换

$\pi_k\leftarrow\pi'_k,\mathrm{NSH}\leftarrow\varnothing$

else $\mathrm{NSH}\leftarrow\mathrm{NSH}\cup\{\pi'_k\}$//否则更新邻域搜索记录

输出π_k.

MSNS算法基于“不劣于”的邻域替换方法, 也可以基于“优于”的替换方法, 但是后者更易于陷入局部最优, 实验结果也支持了这一结论.

4.2　基于随机邻域搜索的遗传算法(Genetic algorithm based on MSNS)

本文基于遗传进化思想搜索置换空间. 遗传算法通过交叉和变异生成新个体, 使得种群向着更优化的方向进化. 但是如果新个体距离其局部最优解的距离较远, 则所生成新个体的质量较差. 为了克服遗传算法的局部寻优弱点, 本文提出了算法3—遗传随机搜索算法(genetic stochastic search algorithm, GSSA), 在遗传操作过程中基于随机邻域搜索增强局部寻优能力.

遗传随机搜索算法的遗传操作基于Whitley[12]提出的稳态遗传算法框架, 算法描述如下:

算法 3　输入: 最大外循环迭代次数outStep, 最大内循环迭代次数innerStep; 输出: 优化调度π^*.

$\Pi_1=\varnothing$, $\mathrm{fitness}(\pi^*)=-\infty$, $\mathrm{fitness}(\pi^{\#})=+\infty$, while ($i=1\leqslant P$) //初始化种群

FLAG1: $\pi\leftarrow$RandomPermutation(n) //随机生成新序列

$\pi_i \leftarrow \mathrm{MSNS}(\pi, \mathrm{innerStep})$ //用MSNS算法改进π

if $(\exists\, \pi_j \in \Pi_1, d(\theta(\pi_i), \theta(\pi_j)) == 0)$

go to FLAG1 //不是新解, 则丢弃

$\Pi_1 = \Pi_1 \cup \{\pi_i\}$//新解, 则加入种群

if (fitness(π^*) <fitness(π_i))

$\pi^* \leftarrow \pi_i$//记录最优个体

if (fitness($\pi^{\#}$) <fitness(π_i))

$\pi^{\#} \leftarrow \pi_i$//记录最劣个体

$i = i + 1$//产生下一个序列

for $k = 1$ ←outerStep//进行遗传迭代

FLAG2: π_α, π_β ←RankBasedSelection(Π_k)//基于Rank方法的选择操作

if $(\mathrm{rnd}(\,) \leqslant P_\mathrm{c})$

π_c←Crossover(π_α, π_β)//以概率P_c进行交叉操作

if $(\mathrm{rnd}(\,) \leqslant P_\mathrm{m})$

π_c' ←Mutation(π_c)//以概率P_m进行变异操作

π_c'' ← MSNS (π_c', innerStep)//用MSNS算法改进π_c

if $(\exists\, \pi_i \in \Pi_k, d(\theta(\pi_i), \theta(\pi_c'')) == 0)$

go to FLAG2 //不是新解, 则丢弃

$\Pi_k \leftarrow \Pi_k \setminus \{\pi^*, \pi^{\#}\}$//从$\Pi_k$中移去$\pi^*$和$\pi^{\#}$

if (fitness(π_c'') < fitness(π^*))

$\pi^* \leftarrow \pi_c''$//更新最优个体

if (fitness(π_c'') > fitness($\pi^{\#}$))

$\pi^{\#} \leftarrow \pi_c''$//更新最劣个体

$\Pi_k \leftarrow \Pi_k \cup \{\pi^*, \pi^{\#}\}$//更新种群

输出π^*.

其中, 交叉操作基于循环交叉[13], 变异操作基于随机两点元素交换. 与基本稳态遗传算法相比, GSSA算法以MSNS算法替代了变异操作.

5 仿真实验及算法性能对比(Experiments)

仿真实验包括了5颗光学成像卫星(太阳同步极轨道)、5个地面站(分布在境内, 每站一套数传设备). 每颗卫星的最大侧摆角度45°, 每个地面站数传天线最小数传仰角5°. 根据每颗卫星的2行星历数据TLE (two-line element)文件、地面站的经纬度和海拔高度、地面目标的经纬度等数据计算成像时间窗和数传时间窗. 地面目标的优先级在1∼3之间. 每颗卫星的可用能量根据其每圈的太阳光照时间确定(每颗卫星的绕地周期约为95 min, 约63.3%的太阳光照时间). 调度时间为24 h. 共4个调度实例(P404, P820, P1251, P1575), 分别包括404, 820, 1251和1575个对地成像任务. 每个任务被分配一个1∼3之间的数字作为表示其重要性的权值, 数值越大, 该任务越重要; 优化目标为最大化可成像任务的权值总和(式(⊗), 无量纲评价值). 每个调度实例上计算多次取平均值. 测试平台: 2.0 GHz CPU+2048 MB内存. 使用C语言编程实现.

首先, 在各个调度实例上测试有限次迭代就近邻域搜索(FNS)与随机邻域搜索(SNS)在交换邻域和插入邻域上的搜索性能差别, 最大迭代次数10^4次. 在随机邻域搜索算法上分别测试是否采用记忆搜索和“不劣于/优于”替换方法对搜索结果的影响. 计算结果见表1. 表1中: “$\bar{\Sigma}$”为平均调度结果, “$\bar{T}$”为平均算法耗时.

表 1 不同邻域搜索方法的对比结果

Table 1 Neighborhood search results

邻域搜索测试项目				调度实例计算结果							
邻域结构	搜索策略	记忆搜索	替换方法	P404		P820		P1215		P1576	
				$\bar{\Sigma}$	$\bar{T}$/s	$\bar{\Sigma}$	$\bar{T}$/s	$\bar{\Sigma}$	$\bar{T}$/s	$\bar{\Sigma}$	$\bar{T}$/s
交换邻域	FNS	—	—	237.00	48.26	351.90	184.02	393.60	399.11	406.30	633.72
	SNS	否	不劣于	247.00	53.83	402.70	191.23	460.60	407.88	484.10	654.25
		否	优于	243.70	52.87	378.90	185.39	435.70	405.39	451.40	638.28
		是	不劣于	248.00	53.52	403.70	190.15	463.00	407.86	489.40	665.31
		是	优于	244.30	52.68	384.10	185.84	433.20	399.23	452.30	632.63
插入邻域	FNS	—	—	227.10	53.18	337.60	184.34	376.10	403.65	381.80	629.37
	SNS	否	不劣于	247.50	52.98	402.30	185.69	461.70	409.18	485.10	652.69
		否	优于	242.10	53.50	378.90	181.80	435.90	400.63	455.00	630.60
		是	不劣于	247.80	52.36	403.00	187.50	462.60	406.69	484.50	658.27
		是	优于	243.60	53.26	381.40	188.04	436.40	402.56	444.70	630.73

表1说明　在限制最大迭代次数的条件下, 随机邻域搜索优化结果明显优于就近邻域搜索. 基于交换邻域的邻域搜索效果优于基于插入邻域的搜索效果. 对于随机邻域搜索而言, 增加记忆机制对随机邻域搜索的计算耗时影响很小, 却可以带来小幅的优化结果提升; 采用“不劣于”的邻域替换策略时, 邻域搜索结果较“优于”替换策略有所改进.

其次, 测试变异概率对稳态遗传算法的影响. 在遗传算法进化过程中, 变异操作主要是保持群体一定程度的多样性. Barbulescu在求解AFSCN问题[7]时, 所用稳态遗传算法未包括变异操作. 本文实验部分将交换邻域操作作为稳态遗传算法的变异操作, 变异概率取0.01, 0.1和1.0, 种群规模100, 迭代次数10^5. 在各个调度实例上不同变异概率对稳态遗传算法进化过程的影响如图1所示.

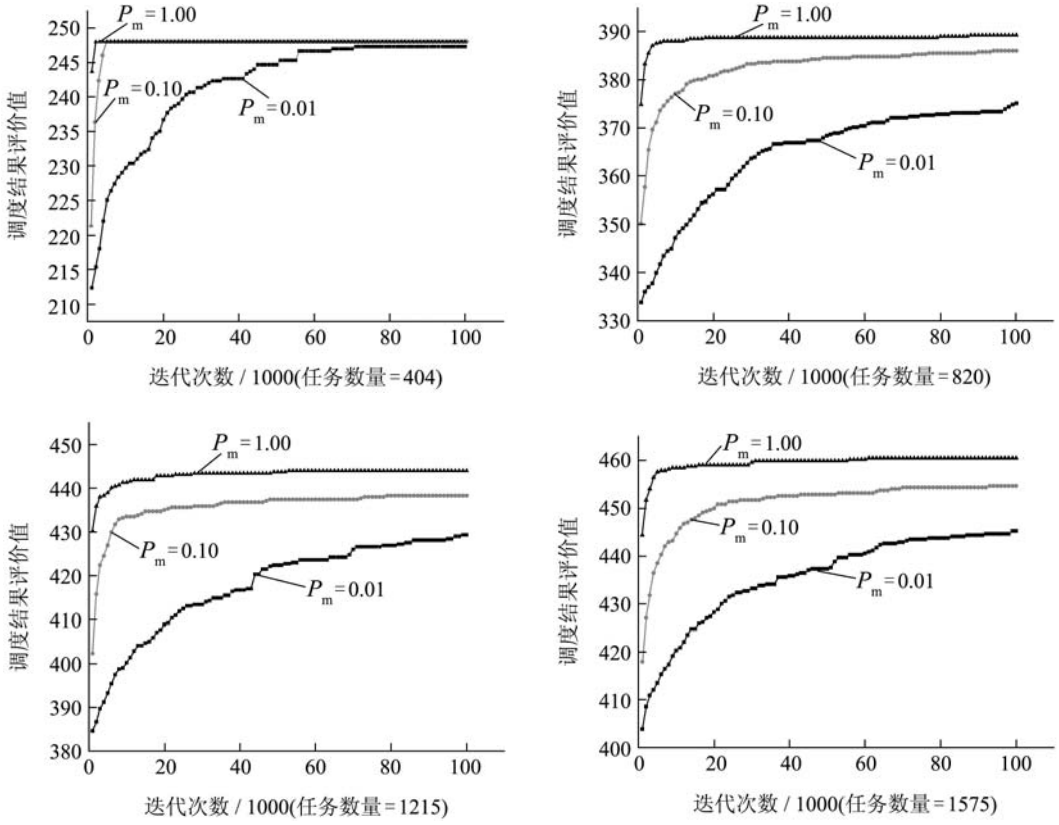

图 1　变异概率对遗传算法进化过程的影响

Fig. 1　GA's evolutionary curve at different mutation rate

图1说明　求解多卫星多地面站调度问题时, 增加变异概率可加快稳态遗传算法的收敛速度, 并改善优化结果. 变异是保持种群多样性的有效手段. 一般来说, 变异概率过小容易导致早熟收敛, 变异概率过大会因为算法随机性过高而导致收敛缓慢. 在标准遗传算法中, 如果种群规模为N, 编码长度为L, 子个体等位基因变异概率为$p_{\rm m}$, 则该个体发生变异的概率为$1-(1-p_{\rm m})^L$. 以

$$N=100, L=200, p_{\rm m}=0.01$$

为例, 则有超过86.6%的概率发生个体变异; 当$L=1000$时, 则有超过99.99%的概率发生个体变异. 因此稳态遗传算法中取值为1.0的变异概率并不算太高. 由于稳态遗传算法具备“保优”作用, 可以将优秀的新个体保留在种群中, 因此变异就可以起到一定的局部改进作用, 这点在图1中得到了验证.

再次, 测试随机邻域搜索对遗传算法的局部寻优能力的改进. 未采用随机邻域搜索的遗传算法(GA)的世代更替次数为10^5, 遗传随机搜索算法(GSSA)的世代更替次数为10^3, 其内嵌的随机邻域搜索迭代次数为10^2(因此对比计算时GA与GSSA具有相同的等效迭代次数). GA基于交换邻域操作方式进行变异, 变异概率取1.0, GSSA中的随机邻域搜索基于交换邻域. 其他参数如种群规模(100)、选择方式、交叉方式, GSSA与GA相同. 计算结果见表2. 表2中“$\bar{\Sigma}$”为平均调度结果, “σ”为调度结果标准差, “$\bar{T}$”为平均算法耗时.

表2说明　随机邻域搜索可以改善遗传算法的局部寻优能力, 在4个调度实例上平均改进幅度为4.64%, 算法耗时有类似幅度的增加. 另一方面, GA调度结果的标准差平均为9.21, GSSA则为2.45, 这说明GSSA有更好的稳定性和抗随机性.

表 2 GSSA与GA的搜索性能对比

Table 2 Computational results of GSSA and GA

调度实例	GA			GSSA			$\frac{\text{GSSA}-\text{GA}}{\text{GA}}$	
	$\bar{\Sigma}$	σ	$\bar{T}$/s	$\bar{\Sigma}$	σ	$\bar{T}$/s	$\bar{\Sigma}$	$\bar{T}$
P404	244.88	1.25	552.51	248.00	0.00	553.19	0.94%	0.12%
P820	388.80	9.96	1755.07	407.80	2.28	1922.50	4.89%	9.54%
P1215	445.80	11.69	4143.11	472.60	3.21	4265.34	6.01%	2.95%
P1575	463.75	13.96	6200.71	495.00	4.32	6697.61	6.74%	8.01%
平均值		9.21			2.45		4.64%	5.16%

6 结论(Conclusion)

本文基于置换搜索方法求解多卫星多地面站调度问题, 可以得出以下几点结论:

1) 数传时间窗优先置换序列映射算法确保可从理论上搜索到最优调度, 并为置换空间上的启发式搜索算法提供了评价值计算方法; 该方法将有约束优化转化为无约束优化, 降低了优化算法对具体调度问题的依赖性.

2) 在置换序列集合的局部搜索过程中, 交换邻域的邻域搜索结果优于插入邻域.

3) 有限次迭代条件下, 随机邻域搜索结果优于就近邻域搜索, 采用随机邻域搜索可更好的改进遗传算法中生成的新个体.

4) 有记忆搜索机制和“不劣于”邻域替换方法可改善随机邻域搜索的结果, 主要体现在可在一定程度上防止局部搜索过程中出现的“回溯”和减少陷入局部最优的概率.

5) 将随机邻域搜索和遗传算法结合在一起的混合搜索算法, 在保留单纯遗传算法全局寻优特性时, 增强了局部寻优能力.

综上所述, 遗传随机搜索算法是基于置换表示方式求解多卫星多地面站调度问题的一种比较有效的优化算法.

参考文献(References):

[1] GLOBUS A, CRAWFORD J, LOHN J, et al. A comparison of techniques for scheduling earth observing satellites[C] //*Proceedings of the 16th Innovative Applications of Artificial Intelligence*. San Jose, CA, USA: AAAI Press, 2004: 836 – 843.

[2] BIANCHESSI N. *Planning and scheduling problems for earth observation satellites: Models and algorithms*[D]. Milan, Italy: University of Milan, 2006.

[3] 李菊芳. 航天侦察多星多地面站任务规划问题研究[D]. 长沙: 国防科学技术大学, 2004.

[4] 王钧. 成像卫星综合任务调度模型与优化方法研究[D]. 长沙: 国防科学技术大学, 2007.

[5] SOMA P, VENKATESWARLU S, SANTHALAKSHMI S, et al. Multi-satellite scheduling using genetic algorithms[C] //*Proceedings of the 8th International Conference on Space Operations*. Motreal, Canada: [s.n.], 2004.

[6] MARINELLI F, NOCELLA S, ROSSI F, et al. *A Lagrangian Heuristic for Satellite Range Scheduling with Resource Constraints*[R]. Italy: Universit‘a di L’ Aquila, 2005.

[7] BARBULESCU L, HOWE A, WHITLEY D. AFSCN scheduling: how the problem and solution have evolved[J]. *Mathematical and Computer Modelling*, 2006, 43(9/10): 1023 – 1037.

[8] KRAMER L, BARBULESCU L, SMITH S. Analyzing basic representation choices in oversubscribed scheduling problems[C] //*Proceedings of the 3rd Multidisciplinary International Conference on Scheduling: Theory and Application*. Paris, France: [s.n.], 2007.

[9] ERGUN O, ORLIN J B. Fast neighborhood search for the single machine total weighted tardiness problem[J]. *Operations Research Letters*, 2006, 34(1): 41 – 45.

[10] WOFFE W J, SORENSEN S E. Three scheduling algorithms applied to the earth observing systems domain[J]. *Management Science*, 2000, 46(1): 148 – 166.

[11] 孙元凯, 刘民, 吴澄. 变邻域结构Tabu搜索算法及其在Job Shop调度问题上的应用[J]. 电子学报, 2001, 29(5): 622 – 625.
(SUN Yuankai, LIU Min, WU Cheng. Tabu search algorithm with varying neighborhood and its application to job shop scheduling problem[J]. *Acta Electronic Sinica*, 2001, 29(5): 622 – 625.)

[12] WHITLEY D, KAUTH J. Genitor: A different genetic algorithm[C] //*Proceedings of the Rocky Mountain Conference on Artificial Intelligence*. Denver, Colorado, USA: [s.n.], 1988: 118 – 130.

[13] OLIVER I, SMITH D, HOLLAND J. A study of permutation crossover operators on the traveling salesman problems[C] //*Proceedings of 2nd International Conference on Genetic Algorithms and Their Applications*. Cambridge, USA: Lawrence Erlbaum Associates In. 1987: 224 – 230.

作者简介:

靳肖闪 (1978—), 男, 博士生, 目前研究方向为遥感与地理信息集成, E-mail: jinxiaoshan@nudt.edu.cn;

李 军 (1973—), 男, 副教授, 硕士生导师, 主要研究遥感与地理信息集成;

王 钧 (1978—), 男, 讲师, 主要研究遥感与地理信息集成;

景 宁 (1963—), 男, 教授, 博士生导师, 主要研究地理信息系统与数据库技术.