

文章编号: 1001- 2486(2009) 01- 0058- 06

考虑随机回放的卫星数传调度问题的一种求解方法*

靳肖闪, 李 军, 王 钧, 景 宁

(国防科技大学 电子科学与工程学院, 湖南 长沙 410073)

摘要: 针对考虑随机回放的卫星数传调度问题, 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, 基于有记忆功能的随机邻域搜索, 在置换空间上搜索产生优化调度的置换序列。仿真计算表明, 遗传随机搜索算法可以增强遗传算法的局部搜索能力, 在搜索结果上平均获得了 2.72% 的改进。

关键词: 卫星数传调度; 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

A New Scheduling Method for Satellites' Randomized Data Transmission

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

(College of Electronics Science and Engineering, National Univ. of Defense Technology, Changsha 410073, China)

Abstract: Our work focuses on permutation scheduling and permutation searching methods for the scheduling problem of satellites' randomized data transmission. A time window preempted permutation scheduling algorithm is put forward, which can ensure searching for optimal scheduling. A genetic stochastic search method is designed to carry out the search procedure for optimal permutations, using a stochastic neighborhood search algorithm with memory. Computational results show that our method can enhance the local search ability for genetic search, and makes an average improvement of about 2.72%.

Key words: satellites' data transmission scheduling; permutation-based representation; genetic algorithm; stochastic neighborhood search

地面站对地球观测卫星观测数据的接收, 是对地观测活动中的重要组成部分, 影响到观测数据对最终用户的可用性及时效性评价。当卫星具有随机回放数传能力时, 地面站对卫星的一次数传活动, 可对多个任务进行选择数传。同一数传设备的数传时间窗的重叠, 以及同一数传任务可选择多个数传时间窗的情形, 是优化调度的过程需要同时考虑的因素。已有的研究大多基于星上数据顺序回放, 把卫星的一次数传活动作为一个完整的数传任务, 或者未考虑数传时间窗的重叠影响^[1-5]。

卫星的数传调度是包含了大量约束的过度 (Oversubscribed) 调度问题, 即在满足各类数传约束的条件下, 最大化利用地面站的数传能力。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 (Permutation) 表示方法的编码长度短, 可有效处理大量约束, 适用于解决此类过度调度问题^[6]。好的邻域结构可以有效改善局部搜索的质量, 交换 (Swap) 邻域和插入 (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表示方式^[7]。Barbulescu 等在求解 AFSCN 调度问题时, 采用稳态遗传算法获得了较好的优化结果^[3]。虽然遗传算法具有较强的全局搜索能力, 但在局部搜索性能上较差。本文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。

* 收稿日期: 2008- 04- 09

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 (2007AA 12Z229); 国家科技支撑计划课题 (2006BAG01A07)

作者简介: 靳肖闪 (1978-), 男, 博士生。

1 考虑随机回放的卫星数传调度问题

考虑随机回放的卫星数传调度问题可以简单描述为:卫星每次处于地面站接收范围内进行数据下传时,对星上存储的多个数传任务有选择地进行执行。调度的目标是在满足各类数传约束条件下,最大化可执行的数传任务。具体包括:

(1) s 个数传设备 G_1, G_2, \dots, G_s , 每个数传设备 G_i 的等效数传倍率为 v_i , 数传切换时间为 Δ_i 。此处假定同一数传设备对所有数传任务具有相同的数传倍率。

(2) n 个卫星数传任务 J_1, J_2, \dots, J_n , 每个数传任务 J_i 可以表示 $\langle d_i, t_i^s, t_i^e, p_i \rangle$ 。其中, d_i 是任务需要的数据下传时间(数传倍率为 1 时); t_i^s 和 t_i^e 是任务时间窗起始和结束时间, 该任务必须在此时间窗内执行; p_i 是任务的优先级。数传任务可能来自单颗卫星, 也可能来自多颗卫星。

(3) m 个数传时间窗 W_1, W_2, \dots, W_m , 受地球曲率及天线设备等的使用特性的影响, 卫星只有在经过地面站可数传空域时, 才可以把数据下传给地面站。每个数传时间窗 W_i 可表示为 $\langle T_i^s, T_i^e, D_i, G_i \rangle$ 。其中, T_i^s 和 T_i^e 是可以进行数传的起始和结束时间; D_i 是该时间窗内可执行的数传任务集合, 即 $D_i \subseteq \{J_1, J_2, \dots, J_n\}$; G_i 为该数传时间窗对应的数传设备。

(4) 数传调度决策变量 $x_{i,j}$, 当任务 J_i 在时间窗 W_j 上执行时, $x_{i,j} = 1$; 否则, $x_{i,j} = 0$ 。

(5) 连续时间变量 t_i , 为 W_i 实际开始数传的时间。 W_i 在数传设备 G_i 上不与其他数传时间窗重叠时, $t_i = T_i^s + \Delta_{G_i}$; 否则, 根据数传设备 G_i 上一次数传活动的结束时间确定 t_i 。

数传过程需满足的约束条件包括:

• 任务时间窗约束。数传任务在其任务时间窗内完成:

$$\forall 1 \leq i \leq n, 1 \leq j \leq m: x_{i,j} (t_i^s - T_j^s) \leq 0, x_{i,j} (t_i^e - T_j^e) \geq 0 \quad (1)$$

• 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:

$$\forall 1 \leq j \leq m: t_j + \sum_{\substack{i \in D_j \\ j}} x_{i,j} v_{G_j} d_i \leq T_j^e \quad (2)$$

• 数传唯一性约束。每个数传任务仅需数传一次:

$$\sum_{j=1}^m x_{i,j} \leq 1 \quad (3)$$

• 数传设备切换时间约束。同一数传设备执行完时间窗 W_k 上的数传任务后, 需要足够的切换时间, 以执行下一时间窗 W_l 上的数传任务, 此约束确保同一数传设备上的数传任务不会在时间上冲突:

$$G_k = G_l, \exists J_i \in D_k, J_j \in D_l, x_{i,k} = x_{j,l} = 1: t_k + \sum_{\substack{i \in D_k \\ k}} x_{i,k} v_{G_k} d_i \leq t_l \quad (4)$$

考虑随机回放的卫星数传调度问题, 在同时满足上述 4 类约束时, 最大化可完成的数传任务, 即:

$$\max \sum_{i=1}^n \sum_{j=1}^m x_{i,j} p_i \quad (5)$$

Barbulescu 等证明此类调度问题是 NP 难解问题, 需要采用近似算法求解^[3]。本文基于置换表示的调度方法求解考虑随机回放的卫星数传调度问题, 有两个问题需要解决: (1) 建立置换空间到调度解空间的映射算法, 使得每一个数传任务置换序列可以对应到某一个可行调度, 并藉此计算置换序列的适应值; (2) 建立置换空间的搜索算法, 以替代直接地在调度解空间上的搜索。置换序列的适应值确定搜索算法在置换空间的搜索方向, 以指导搜索算法搜索可生成优化调度结果的置换序列。

2 置换序列的调度算法

2.1 相关定义

定义 1 组合数传时间窗 W'_k 是数传时间窗集合的一个子集 $W'_k \subseteq \{W_1, W_2, \dots, W_m\}$, 共 m' 个, 且满足:

$$\left\{ \begin{array}{l} \forall W_i \in W'_k, \forall W_j \in W'_k, G_i = G_j \Rightarrow T_i^s < T_j^s \text{ 且 } T_i^e \leq T_j^e \text{ 或 } T_i^s > T_j^s \text{ 且 } T_i^e \geq T_j^e \\ |W'_k| > 1 \Rightarrow \forall W_i \in W'_k, \exists W_j \in W'_k \setminus \{W_i\}, G_i = G_j, T_i^s \leq T_j^s \text{ 且 } T_i^e > T_j^e \text{ 或 } T_i^s > T_j^s \text{ 且 } T_i^e < T_j^e \\ \forall i \neq j, W'_i \cap W'_j = \emptyset; \quad W'_1 \cup W'_2 \cup \dots \cup W'_m = \{W_1, W_2, \dots, W_n\} \end{array} \right. \quad (6)$$

全部的组合数传时间窗是数传时间窗集合的一个划分。每个组合数传时间窗或者只包含一个数传时间窗,或者包含若干个在时间上重叠的数传时间窗(对应同一数传设备)。

所有的组合数传时间窗满足如下的排列顺序:

$$\forall W'_i, \forall W'_j, \quad \min(T_k^s) | W_k \in W'_i < \min(T_l^s) | W_l \in W'_j \Rightarrow i < j \quad (7)$$

所有的数传时间窗满足如下的排列顺序:

$$\forall W_i \in W'_k, \quad \forall W_j \in W'_l, \quad k < l \Rightarrow i < j; \quad \forall W_i, W_j \in W'_k, \quad T_i^s < T_j^s \Rightarrow i < j \quad (8)$$

定义2 任务置换序列 π_k 是所有数传任务的一个全排列, $\pi_k = \langle o_{k,1}, o_{k,2}, \dots, o_{k,n} \rangle$, $o_{k,i}$ 表示任务 $o_{k,i}$ 在 π_k 中的序号为 i 。 $\Pi = \{ \pi_1, \pi_2, \dots, \pi_m \}$ 是所有任务置换序列的集合。

定义3 任务置换调度序列 $\theta(\pi_k) = \langle \langle o_{k,1}, q_{k,1} \rangle, \langle o_{k,2}, q_{k,2} \rangle, \dots, \langle o_{k,n}, q_{k,n} \rangle \rangle$, 是按照某种分配规则为 π_k 生成的一个调度结果。其中, $0 \leq q_{k,i} \leq m$, 表示为任务 $o_{k,i}$ 分配的数传时间窗, $q_{k,i} = 0$ 表示未分配数传时间窗。

定义4 饱和调度 $s_k = \langle \langle o_{k,1}, p_{k,1} \rangle, \langle o_{k,2}, p_{k,2} \rangle, \dots, \langle o_{k,n}, p_{k,n} \rangle \rangle$, $o_{k,i}$ 的含义同定义2, $p_{k,i}$ 的含义与定义3中的 $q_{k,i}$ 相同。 s_k 满足: $p_{k,i} > 0$ 时, $p_{k,i} < p_{k,j} \Rightarrow i < j$; $p_{k,i} = 0$ 时, $p_{k,j} > 0 \Rightarrow j < i$ 。对于 $\forall j > i, p_{k,i} > 0$, 如果令 $p_{k,j} = p_{k,i}$, 则违反成像约束, 就称 s_k 是饱和调度。饱和调度是每一颗卫星的成像能力得到最大利用(不能承担本属于更大序号数传时间窗或未分配数传时间窗的数传任务)的调度结果。如果限制搜索优化调度的过程只包含饱和调度, 则可以提高搜索的效率和效果。 $S = \{ s_1, s_2, \dots, s_m \}$ 是所有饱和调度的集合。

定义5 置换序列调度算子 $\Lambda(\cdot)$: 将数传时间窗分配给置换序列的操作, 即 $\Lambda(\pi_k) \rightarrow \theta(\pi_k)$ 。

定义6 置换序列评价函数 $fitness(\cdot)$: $\Pi \rightarrow Z^+$, 是对任务置换序列施加调度操作 $\Lambda(\cdot)$ 后所得数传调度结果的评价值, 与调度模型的目标函数(\hookrightarrow)对应。

定义7 置换序列调度距离函数 $d(\theta(\pi_k), \theta(\pi_j)) = \sum_{k=1}^n \sum_{l=1}^n differ(q_{i,k}, q_{j,l})$, 用以度量置换序列施加调度操作后调度结果的相似度。其中, 当 $o_{i,k} = o_{j,l}$ 且 $q_{i,k} = q_{j,l}$ 时, $differ(q_{i,k}, q_{j,l}) = 0$; 否则, $differ(q_{i,k}, q_{j,l}) = 1$ 。

定义8 约束检测函数 $check(\theta(\pi_k))$: 如果 $\theta(\pi_k)$ 满足所有的数传约束, 则 $check(\theta(\pi_k)) = 1$; 否则, $check(\theta(\pi_k)) = 0$ 。

2.2 置换序列映射算法

性质1 存在饱和调度与最优数传调度具有相同评价值。

证明 对最优数传调度 $\theta(\pi_k^*)$, 将该调度按照各个任务分配的数传时间窗重排序, 分配第一个数传时间窗的任务排在最前, 然后依次是分配第2个数传时间窗、……、第 m 个数传时间窗的任务, 未分配数传时间窗的任务排在最后, 从而生成具有一定顺序的任务置换序列 $\pi_k^* = \langle o_{k,1}, o_{k,2}, \dots, o_{k,n} \rangle$, 其对应的最优调度序列为 $\theta(\pi_k^*) = \langle \langle o_{k,1}, q_{k,1}^* \rangle, \langle o_{k,2}, q_{k,2}^* \rangle, \dots, \langle o_{k,n}, q_{k,n}^* \rangle \rangle$ 。假设时间窗 $q_{k,i}^*$ 还可以执行数传任务 $o_{k,j}$ ($j > i$), 则令 $q_{k,j}^* = q_{k,i}^*$, 且不改变其他任务的时间窗分配结果。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 $q_{k,j}^* = 0$, 这说明修改后的调度结果优于最优调度, 显然这是不可能的, 所以必然有 $q_{k,j}^* > q_{k,i}^*$ 。按照以上方法, 依次使得时间窗 W_1, W_2, \dots, W_m 不能容纳其他的数传任务, 从而可将最优调度转换为饱和调度 $\theta(\pi_k)$ 。由于转换的过程不改变整个调度结果的评价值, 且 $\theta(\pi_k)$ 仍然是可行调度, 则 $\theta(\pi_k)$ 必然与 $\theta(\pi_k^*)$ 有相同的评价值。

性质1说明, 饱和调度集合包含了最优调度(达到最优调度评价值)。

已有的研究^[3]采用了基于贪婪思想的任务优先置换序列调度方法: 按照数传时间窗的早晚顺序, 为

置换序列中的任务逐次进行最大可能的数传时间窗分配操作, 直到置换序列中的任务全部处理完毕。一旦理论最优置换序列中所有已经分配了数传时间窗的任务, 其分配的数传时间窗与其应该优先分配的数传时间窗不相同, 则按照上述算法流程, 这些任务分配的数传时间窗必然不同于理论最优置换序列中对应任务分配的数传时间窗, 从而使得置换空间的搜索算法最多只能搜索到生成次优调度的置换序列。为了克服以上缺陷, 本文提出了一种时间窗优先置换序列映射算法, 以建立从置换空间到饱和调度空间的完全对应关系。具体步骤描述如下:

输入: 任务置换序列 π_k

输出: 任务置换调度序列 $\theta(\pi_k)$

- ① for $i = 1 \leftarrow m$: $t_i \leftarrow T_i^s + \Delta C_i$; for $i = 1 \leftarrow n$: $q_{k,i} \leftarrow 0$
- ② for $i = 1 \leftarrow m'$
- ③ for each $W_j \in W'_i$ (in ascent order by T_j^s)
- ④ for $l = 1 \leftarrow n$
- ⑤ if $q_{k,l} = 0$: $q_{k,l} \leftarrow j$
- ⑥ if $\text{check}(\theta(\pi_k)) = 0$: $q_{k,l} \leftarrow 0$
- ⑦ else
- ⑧ for each $W_h \in W'_i \setminus \{W_j\}$
- ⑨ if $(T_h^s \geq T_j^s \ \& \ T_h < T_j^r)$: $t_h \leftarrow \max(t_h, t_j + \sum_{d|W_{k,g}=j} v_{G_j} d_{o_{k,g}})$

性质 2 时间窗优先置换序列映射算法, 所生成的任务置换调度序列是饱和调度, 且任意一种饱和调度均可由该算法对某任务置换序列实施调度操作而得到。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 Π 到饱和调度集合 S 的多对一映射。

证明 时间窗优先置换序列映射算法采用最大可能数传时间窗分配算法, 使得每个时间窗上的数传能力得到最大利用, 符合饱和调度的定义, 因此生成的任务置换调度序列必然是饱和调度。

对任意一个饱和调度 $\theta(\pi_k)$, 将该饱和调度按照各个任务分配的数传时间窗重排序, 分配第一个数传时间窗的任务排在最前, 然后依次是分配第 2 个数传时间窗、……、第 m 个数传时间窗的任务, 未分配数传时间窗的任务排在最后, 从而生成具有一定顺序的任务置换序列 $\pi_k = \langle o_{k,1}, o_{k,2}, \dots, o_{k,n} \rangle$, 其对应的饱和调度序列为 $\theta(\pi_k) = \langle \langle o_{k,1}, \dot{q}_{k,1} \rangle, \langle o_{k,2}, \dot{q}_{k,2} \rangle, \dots, \langle o_{k,n}, \dot{q}_{k,n} \rangle \rangle$ 。由于饱和调度是可行调度, 则时间窗优先置换序列映射算法为每个任务所分配的时间窗 $q_{k,i}$, 满足 $q_{k,i} \geq \dot{q}_{k,i}$ 。根据饱和调度的定义, $\dot{q}_{k,i}$ 是按照数传时间窗的序号顺序, 为任务 $o_{k,i}$ 所能分配的数传时间窗的最大序号, 即 $q_{k,i} \leq \dot{q}_{k,i}$ 。因此必然有 $q_{k,i} = \dot{q}_{k,i}$, 即任意一个饱和调度均可将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。

分配同一数传时间窗的任务间任意改变顺序, 都不会改变时间窗优先置换序列映射算法为各个任务所分配的数传时间窗。意即有不少于一个置换序列, 经时间窗优先置换序列映射算法实施调度操作后, 生成的调度结果与特定饱和调度相同, 从而时间窗优先置换序列映射算法调度算子 $\Lambda(\cdot)$ 建立了任务置换序列集合 Π 到饱和调度集合的多对一映射。

性质 2 说明, 基于时间窗优先置换序列映射算法计算置换序列的评价值, 确定了置换序列搜索算法的搜索方向, 当该搜索算法最终收敛到最优置换序列时, 根据该置换序列可以生成理论上的最优调度。

3 遗传随机搜索算法

遗传算法通过交叉和变异生成新个体, 从而更新整个种群。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, 本文提出了一种遗传随机搜索算法 (Genetic Stochastic Search Algorithm, GSSA), 基于随机邻域搜索增强遗传算法的局部寻优性能。

遗传随机搜索算法的主流程基于 Whitley 提出的稳态遗传算法^[8] 框架, 主要步骤如下:

输入: 考虑随机回放的卫星数传调度问题

输出: 优化调度

(1) 初始化种群 $P = \{p_1, p_2, \dots, p_{|P|}\}$, 基于时间窗优先置换序列映射算法, 计算 P 中个体的适应值。

(2) while not stop do

- ① 基于 Rank 方法选择出两个父个体 p_{f1}, p_{f2}
- ② 以概率 P_c 对 p_{f1}, p_{f2} 施加交叉操作, 产生子个体 p_c
- ③ 以概率 P_m 对子个体 p_c 施加变异操作, $p_c \rightarrow p'_c$
- ④ 计算 p'_c 适应值
- ⑤ 如果 p'_c 与当前种群中的某个个体的调度距离为 0 时, 丢弃 p'_c , 返回 ①
- ⑥ 调用邻域搜索算法改进 $p'_c, p'_c \rightarrow p''_c$
- ⑦ 以 p''_c 替换当前种群中的最劣个体

(3) 输出最优个体的调度结果

其中, 交叉操作基于循环交叉^[9]; 变异操作基于随机两点元素交换。

邻域搜索可分为系统的邻域搜索和随机的邻域搜索两类。系统的邻域搜索包括两种选择策略: 优邻域选择(Best Neighborhood Search, BNS)策略, 从当前解的所有邻域中选择最好的解; 近邻域选择(Fast Neighborhood Search, FNS)策略, 按照一定搜索顺序, 从邻域中就近选择优于当前解的解。交换和插入具有 $O(n^2)$ 的邻域结构复杂度(最优交换邻域搜索每一次迭代需要搜索 $(n^2 - n)/2$ 个邻域解), 所以在调度问题规模较大时, 最优邻域选择策略将因为迭代耗时过长而变得不可取。

本文提出一种有记忆随机邻域搜索(Stochastic Neighborhood Search with Memory, MSNS)算法, 来改进遗传操作生成的新个体。MSNS 算法描述如下:

输入: 给定初始解 s_0

输出: 初始解的优化解

(1) 第 k 次迭代 s_k 的邻域为 $N(s_k)$, 邻域搜索记录 $NSH = \cong$, 初始解为 $s_1 = s_0$

(2) for $k = 1 \leftarrow MAX_STEP$

从 $N(s_k) \setminus NSH$ 中随机选择一个解 s'_k , 如果 s'_k 不劣于 s_k , 则接受 $s'_k: s_k \leftarrow s'_k$, 且令 $NSH = \cong$; 否则 $NSH = NSH \cup \{s'_k\}$

(3) 输出初始解的优化解

随机邻域搜索是无记忆的邻域搜索, 在当前解的邻域搜索过程中可能会出现重复的邻域搜索。本文为随机邻域搜索增加了记忆功能, 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。

4 仿真实验及算法性能对比

在 12 个不同规模的调度实例(2/5 套数传设备, 500/1000 个数传任务, 20/50/100 个数传时间窗)上验证所提算法, 每个调度实例上计算 10 次。测试平台: 2.8GHz CPU, 512MB 内存。使用 C 语言编程。

第一组实验中, 在各个调度实例上测试有限次迭代就近邻域搜索(FNS)、无记忆随机邻域搜索(SNS)和有记忆随机邻域搜索(MSNS)在交换邻域和插入邻域上的搜索性能差别, 最大迭代次数为 10^4 次。第二组实验中, 测试有记忆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 (对比计算时确保 GA 与 GSSA 具有相同的等效迭代次数)。实验结果见表 1。

表 1 实验结果
Tab. 1 Experimental results

实例 规模	第一组实验结果														第二组实验结果						
	Inset Neighborhood							Swap Neighborhood							GA			GSSA			
	FNS		SNS		MSNS		%	FNS		SNS		MSNS		%	Σ	Σ^*	T	Σ	Σ^*	T	
	Σ	T	Σ	T	Σ	T		Σ	T	Σ	T	Σ	T								
500X20X2	448.0	2.2	493.0	5.1	492.6	4.9	9.96	465.9	2.1	493.2	4.8	493.5	4.8	5.92	488.3	496	22.4	494.3	499	50.4	1.23
500X20X5	403.7	2.0	456.8	4.8	457.2	4.8	13.25	421.7	2.0	457.6	4.7	457.7	4.7	8.54	447.7	461	21.1	458.3	463	49.4	2.37
500X50X2	925.3	4.8	1026.3	7.4	1025.9	7.4	10.87	948.9	4.8	1027.8	7.2	1028.1	7.2	8.35	999.0	1038	49.2	1032.3	1044	74.9	3.33
500X50X5	994.9	4.6	1097.9	7.3	1098.2	7.3	10.38	1021.3	4.6	1098.8	7.1	1099.0	7.1	7.61	1071.3	1113	47.7	1101.7	1120	74.6	2.84
500X100X2	1497.9	8.6	1618.9	11.1	1618.2	11.1	8.03	1519.7	8.6	1617.7	10.8	1617.8	10.8	6.46	1585.0	1635	85.9	1625.0	1639	112.6	2.52
500X100X5	1555.2	8.5	1656.1	11.1	1656.4	11.1	6.51	1580.5	8.5	1657.9	10.9	1659.1	10.9	4.97	1632.0	1677	84.5	1660.0	1682	113.6	1.72
1000X20X2	453.9	3.6	503.2	11.4	504.4	11.4	11.13	470.5	3.6	505.4	11.2	505.0	11.2	7.33	498.0	514	38.3	506.3	515	117.2	1.67
1000X20X5	454.8	4.0	513.7	11.8	512.7	11.7	12.73	473.0	4.1	514.5	11.6	514.5	11.6	8.77	501.7	522	42.6	517.7	525	121.2	3.19
1000X50X2	996.1	9.4	1107.9	16.9	1109.1	16.9	11.34	1012.4	9.4	1112.0	16.6	1110.1	16.6	9.65	1073.3	1126	96.5	1117.0	1131	172.1	4.07
1000X50X5	1109.6	9.6	1210.9	17.2	1210.9	17.2	9.13	1127.4	9.6	1216.2	16.9	1216.3	16.9	7.89	1191.0	1237	99.0	1222.3	1242	175.7	2.63
1000X100X2	1740.7	19.3	1928.9	26.4	1932.4	26.4	11.01	1768.9	19.3	1936.2	25.9	1937.5	25.9	9.53	1882.3	1952	196.5	1944.7	1961	269.9	3.32
1000X100X5	1994.8	19.2	2194.0	26.4	2195.8	26.4	10.08	2014.0	19.3	2202.0	25.9	2202.6	25.9	9.36	2133.3	2231	195.4	2214.3	2237	269.8	3.80
总计	MSNS 比 FNS 平均改进: 10.37%							MSNS 比 FNS 平均改进: 7.87%							GSSA 比 GA 平均改进: 2.72%						

注: “ Σ ”为平均调度结果, “ Σ^* ”为最好调度结果, “T”为算法平均耗时, “%”为MSNS相对于FNS或GSSA相对于GA的改进百分比。

第一组实验结果说明: 随机邻域搜索优于就近邻域搜索, 采用交换邻域的随机邻域搜索优于采用插入邻域的随机邻域搜索。有记忆的随机邻域搜索在大部分的调度实例上获得了更好的搜索结果。交换和插入两种邻域具有类似的邻域规模, 体现在使用相同邻域搜索方法时搜索耗时差别不大。增加记忆机制对随机邻域搜索的计算耗时影响很小。MSNS在邻域搜索上的性能表现主要得益于两点: 一是允许具有相同评价价值的新解替换原解, 增强了跳出局部最优解的能力; 二是记忆功能可在一定程度上减少重复搜索。

第二组实验结果则说明, 有记忆随机邻域搜索可以改进遗传算法的局部寻优效果, 在12个调度实例上的平均改进幅度为2.72%, 所得最好调度结果也优于遗传算法, 而计算耗时有一定程度的增加。本文实验中GSSA的随机邻域搜索迭代次数固定为100, 增加随机邻域搜索迭代次数可以更好地改进遗传算法的局部寻优能力。

5 结论

采用置换表示方法将卫星数传调度问题映射为置换空间上的序列搜索, 便于处理类似调度问题数量众多、类型复杂的约束。对于考虑随机回放的卫星数传调度问题, 不同的邻域结构和邻域搜索策略对算法搜索结果有较大的影响: 交换邻域局部搜索性能优于插入邻域; 随机邻域搜索结果优于有限次迭代的就近邻域搜索; 在随机邻域搜索算法中引入记忆机制, 基本不增加计算耗时, 搜索结果有小的改进。

在整体搜索性能上, 基于有记忆随机邻域搜索的遗传随机搜索算法, 比单纯的遗传算法平均有所改进, 是求解考虑随机回放的卫星数传调度问题的一种有效算法。

参考文献:

- [1] 刘洋, 陈英武, 谭跃进. 基于贪婪算法的卫星地面站任务规划方法[J]. 系统工程与电子技术, 2003, 25(10): 1239-1241.
- [2] 金光, 武小悦, 高卫斌. 基于冲突的卫星地面站系统资源调度与能力分析[J]. 小型微型计算机系统, 2007, 28(2): 310-312.
- [3] Barbulescu L, Howe A, Whitley D. AFSCN Scheduling: How the Problem and Solution Have Evolved[J]. Mathematical and Computer Modelling, 2006, 43: 1023-1037.
- [4] Soma P, Venkateswarlu S, Santhalakshmi S, et al. Multi-satellite Scheduling Using Genetic Algorithms[C]//SPACE, 2004.
- [5] Marinelli F, Nocella S, Rossi F, et al. A Lagrangian Heuristic for Satellite Range Scheduling with Resource Constraints[R]. Technical Report TRCS 004, 2005.
- [6] 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, 2007.
- [7] 罗家祥, 唐立新. 带释放时间的并行机调度问题的ILS&SS算法[J]. 自动化学报, 2005, 31(6): 917-924.
- [8] Whitley D, Kauth J. Genitor: A Different Genetic Algorithm[C]//Proceedings of the Rocky Mountain Conference on Artificial Intelligence, 1988: 118-130.
- [9] 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, 1987.