

基于混合蚁群优化的卫星地面站系统任务调度方法

邢立宁¹ 陈英武¹

摘要 卫星地面站系统任务调度是一个典型的组合优化问题, 优化过程极其复杂. 鉴于此, 提出了一种有效求解该问题的基于蚁群优化算法和导向局部搜索方法的混合优化方法. 该方法将蚁群优化和导向局部搜索有效地结合在一起, 极大地提高了优化绩效. 实例计算结果表明, 该混合方法能有效地求解卫星地面站系统任务调度问题.

关键词 卫星地面站系统, 蚁群优化, 导向局部搜索, 任务规划
中图分类号 TP18, O224

Mission Planning of Satellite Ground Station System Based on the Hybrid Ant Colony Optimization

XING Li-Ning¹ CHEN Ying-Wu¹

Abstract The mission planning of satellite ground station system is a representative of combinative optimization problem, and its optimization process is very complicated. For this reason, this paper proposes a hybrid approach which combines the ant colony optimization with guided local search to solve this problem. The optimization performance is improved largely by effectively integrating guided local search with ant colony optimization. Experimental results suggest that the proposed algorithm is a viable and effective approach for the mission planning of satellite ground station system.

Key words Satellite ground station system, ant colony optimization, guided local search, mission planning

卫星地面站系统任务调度问题是一个基于约束的资源优化问题, 即在给定时间内向需要执行的任务分配地面站及执行时间. 该问题的优化目标可描述为: 在给定时间内完成最多的任务或最大化完成任务的权重值之和 (考虑任务权重时). 该问题的难点在于: 资源 (地面站) 相对于某个特定活动 (任务) 来讲, 只有一个或几个可用的时间窗口 (卫星和地面站之间满足任务要求的时间区段); 其调度过程既包括了资源指派问题, 又包含了时间窗口的分配问题.

各国学者通常采用人工智能方法来求解卫星地面站系统任务调度问题, 如贪婪算法^[1]、动态规划^[2]、启发式搜索方法^[3]、约束满足方法^[4]等. 如何更加快速有效地求解该问题, 依然是摆在人们面前的一个严峻挑战. 鉴于此, 本文提出了一种有效求解该问题的基于蚁群优化 (Ant colony optimization, ACO) 算法^[5]和导向局部搜索 (Guided local search, GLS) 方法^[6]的混合蚁群优化方法.

1 卫星地面站系统任务调度模型

卫星地面站系统任务调度问题可描述为: 在时间窗口的约束下, 考虑任务权重和地面站天线转换

时间, 以最大化完成任务权重之和为目标, 完成对多天线地面站系统的任务调度.

1) 地面站共有 m 个时间窗口 $W = \{w_1, w_2, \dots, w_m\}$, 时间窗口 w_i 的开始时间和结束时间分别为 S_i 和 E_i .

2) 有 n 个需要完成的任务 $A = \{a_1, a_2, \dots, a_n\}$, 每个任务所需时间为 $D = \{d_1, d_2, \dots, d_n\}$, 优先级为 $P = \{p_1, p_2, \dots, p_n\}$, 第 i 个任务的开始时间和结束时间分别为 s_i 和 e_i .

3) 任务决策变量为 $T = \{t_1, t_2, \dots, t_n\}$, 如果第 i 个任务能够完成, 则 $t_i = 1$; 反之, $t_i = 0$.

4) 地面站共有 l 个天线, 天线转换时间 (执行任务前的天线姿态调整时间) 为 $R = \{r_1, r_2, \dots, r_l\}$.

5) 调度开始时间为 T_s , 调度截止时间为 T_e .

卫星地面站系统任务调度模型可表述如下

$$F(T) = \max \left(\sum_{1 \leq j \leq n} t_j p_j \right) \quad (1)$$

约束条件包括

$$\sum_{k \in A_i} t_k (d_k + r_i) \leq E_i - S_i, \quad A_i \subseteq A, i \in [1, m] \quad (2)$$

$$s_j \geq S_i, \quad j \in [1, n], i \in [1, m] \quad (3)$$

$$e_j \leq E_i, \quad j \in [1, n], i \in [1, m] \quad (4)$$

$$T_s \leq s_j \leq T_e, \quad T_s \leq e_j \leq T_e, \quad j \in [1, n] \quad (5)$$

模型说明: 式 (2) 表示在一个时间窗口内完成任务

收稿日期 2006-11-16 收修改稿日期 2007-10-28
Received November 16, 2006; in revised form October 28, 2007
国家自然科学基金 (70601035) 资助
Supported by National Natural Science Foundation of China (70601035)
1. 国防科学技术大学信息系统与管理学院 长沙 410073
1. College of Information System and Management, National University of Defense Technology, Changsha 410073
DOI: 10.3724/SP.J.1004.2008.00414

的持续时间和天线转换时间之和不能超过该时间窗口的长度, 这里 A_i 表示在第 i 个时间窗口内安排的任务集合, r_i 表示在第 i 个时间窗口内执行任务时需要的天线转换时间; 式 (3) 表示如果任务 a_j 在时间窗口 w_i 内执行, 那么任务的开始时间必须在相应时间窗口的开始时间之后; 式 (4) 表示如果任务 a_j 在时间窗口 w_i 内执行, 那么任务的结束时间必须在相应时间窗口的结束时间之前; 式 (3) 和 (4) 确定了任务必须在选定的时间窗口之内完成; 式 (5) 表示所有任务的开始时间和结束时间必须在规定的时间段 $[T_s, T_e]$ 之内。

2 混合蚁群优化算法

在蚁群优化算法中, 人工蚂蚁按照状态转移规则逐步构造可行解, 它们将一些随机性引入到优化结果中; 因此, 通过蚁群优化算法很难快速地得到优化问题的全局最优解. 如果要快速地得到优化问题的全局最优解, 则需要采用局部搜索技术来辅助蚁群优化过程. 文献 [7] 将蚁群优化算法和局部搜索方法有效地集成起来, 已成功应用于作业车间调度问题和带宽最小化问题等组合优化问题. 鉴于此, 本文提出了一种基于蚁群优化算法和导向局部搜索方法的混合优化方法.

2.1 混合蚁群优化算法的优化框架

混合蚁群优化算法的优化框架如图 1 所示.

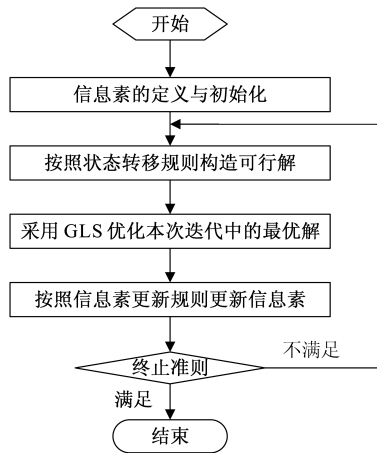


图 1 混合蚁群优化算法的优化框架

Fig. 1 Optimization framework of the hybrid ant colony optimization algorithm

2.2 信息素的定义和初始化

将“给定任务安排在每个时间窗口内的概率”定义为方案空间上的信息素. 用一个 $m \times n$ 维的矩阵 K 来记录信息素, m 代表所有任务的数目, n 代表所有时间窗口的数目. 对于任意的元素 K_{ij} , 它的含义是将任务 i 安排到时间窗口 j 内的概率. 在本

阶段中, 信息素矩阵 K 中的元素都被初始化为 λ_0 .

2.3 可行解的构造

2.3.1 时间窗口内可安排的任务集合

在构造卫星地面站系统任务调度问题的可行解之前, 首先必须确定当前时刻每个时间窗口内可以安排的任务集合. 采用 $allow(k, t)$ 表示 t 时刻第 k 个时间窗口内的可以安排的任务集合, 它的定义如下所示 (Matlab 语言).

```

allow(k, t) = Φ;           % 将 allow(k, t) 初始化为空集;
for i = 1: TaskNum       % TaskNum 表示所有任务的数目;
    Oper = 任务 i;
    if (任务 i 能在时间窗口 k 内执行)
        Time1 = 任务 i 的总执行时间;
        Time2 = t 时刻时间窗口 k 内的剩余可操作时间;
        if (Time1 不大于 Time2)
            if (任务 i 在 t 时刻还没有被安排)
                Oper ∈ allow(k, t);
            end
        end
    end
end
end
end
  
```

2.3.2 状态转移规则

对于每个时间窗口 k , 按照以下的概率分布为该时间窗口选择下一个需要执行的任务 a_i .

$$Pr(a_i, k, t) = \begin{cases} \frac{F(i, k)}{\sum_{a_j \in allow(k, t)} F(j, k)} & a_i \in allow(k, t) \\ 0 & a_i \notin allow(k, t) \end{cases} \quad (6)$$

这里,

$$F(i, k) = [K_{ik}]^a \times [\eta(i)]^b \times [\lambda(i)]^c \quad (7)$$

$Pr(a_i, k, t)$ 表示 t 时刻在时间窗口 k 内选择执行任务 a_i 的概率; K_{ik} 表示将任务 a_i 安排到时间窗口 k 内的信息素启发值; $\lambda(i)$ 表示任务 a_i 的总执行时间启发值, $\lambda(i) = 1/(d_i + r_i)$; $\eta(i)$ 表示任务 a_i 的优先级启发值, $\eta(i) = p_i$; a, b, c 分别代表以上三种不同启发值的权重. 在可行解构造过程中, 采用伪随机比例规则^[5] 替代了传统的随机状态转移规则.

2.3.3 可行解的构造机制

卫星地面站系统任务调度问题可行解的构造机制可简单描述如下: 按顺序依次在每个时间窗口内安排可以安排的任务 (按照状态转移规则从当前时间窗口内可以安排的任务集合中依次选择在该时间窗口内安排的任务), 直到该时间窗口内不能再安排

其他任何任务; 不断重复以上过程, 直到所有的时间窗口内都不能再安排其他任何任务.

2.4 导向局部搜索方法

导向局部搜索方法已被广泛地应用到各种组合优化问题的求解中^[8]. 该方法的目标函数修改机制以方案(解)特征为基础, 具有通用性好、简洁紧凑等特点. 导向局部搜索方法可针对具体问题加以变通和改进, 如采用可变的方案特征成本、改变惩罚系数的修改准则等, 使算法更切合求解具体问题的需要^[9]. 一般来讲, 采用导向局部搜索方法求解优化问题的基本流程如图 2 所示.

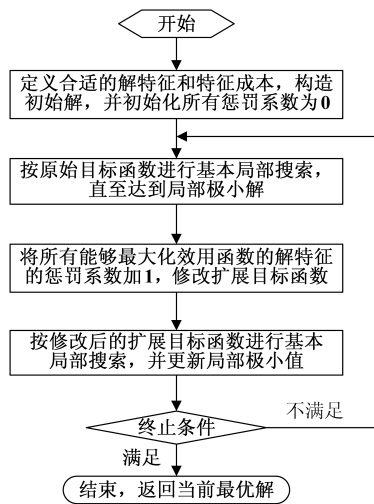


图 2 导向局部搜索方法的基本流程图

Fig. 2 Basic flow chart of guided local search

2.4.1 方案特征的选择

对应每个任务设立一个方案特征, 如果一个方案中没有安排某任务, 则该方案就具备与该任务对应的方案特征, 相应的方案特征成本就是对应任务的优先级. 按照这种方式, 对方案中出现的方案特征的惩罚, 实际上是对没有安排某个任务的惩罚. 按照上述说明, 针对每个任务 $a \in A$, 可定义如下的方案特征指标函数

$$\delta_a(s) = \begin{cases} 1 & a \notin s \\ 0 & a \in s \end{cases} \quad s \in S, a \in A \quad (8)$$

其中, S 表示问题的所有可行方案 s 的集合, A 则表示所有任务 a 的集合.

2.4.2 目标函数的扩展

卫星地面站系统任务调度问题的目标函数可转换为

$$\min_{s \in S} g(s) = \sum_{a \in A} [e_a \cdot \delta_a(s)] \quad (9)$$

其中, e_a 表示对应任务 a 的方案特征成本, 即对应任务的优先级. 选定方案特征后, 依据导向局部搜索

方法的基本思想, 原问题的目标函数将被扩展为

$$\min_{s \in S} h(s) = g(s) + \lambda \sum_{a \in A} p_a \cdot \delta_a(s) = \sum_{a \in A} (e_a + \lambda \cdot p_a) \cdot \delta_a(s) \quad (10)$$

其中, p_a 表示任务 a 所对应方案特征的惩罚系数, λ 表示规范化系数.

2.4.3 规范化系数

对于卫星地面站系统任务调度问题而言, 规范化系数取值主要与任务优先级水平的设置范围有关. 如果任务优先级水平的范围设置在 $1 \sim 10$ 之间, 规范化系数设置在 $0.5 \sim 10$ 之间都能得到较好的结果(设置为 5 左右效果最好); 如果任务优先级水平的范围设置在 $1 \sim 100$ 之间, 规范化系数设置在 $1 \sim 50$ 之间都能得到较好的结果(设置为 30 左右效果最好).

2.4.4 基本邻域搜索

本文的导向局部搜索算法采用的是快速下降的局部搜索迭代方式, 即每次迭代都将在当前解的邻域中, 选择所找到的第一个优于当前解的可行邻居进行移动. 局部搜索的邻域结构则采用的是改进型邻域, 添加未安排访问邻域和替换访问邻域结构的并集. 添加未安排访问邻域将在当前可行解的某时间窗口内的某个任务之后, 插入一个新的未安排任务. 替换访问邻域将以一个未被安排的任务来替换某个已安排任务.

2.4.5 惩罚项的修改

迭代开始之前, 所有方案特征的惩罚系数均为 0. 在迭代过程中, 当基本局部搜索过程达到一个局部极小解 s^* 时, 导向局部搜索方法将以最大化效用函数为原则选择 s^* 中出现的某些方案特征, 并将其惩罚系数加 1, 从而实现动态修改扩展目标函数的目的. 对选中的方案特征惩罚系数的增加, 将使得对应的任务在新的局部极小解中被安排的几率增加.

$$util(s^*, a) = \delta_a(s^*) \cdot \frac{e_a}{1 + p_a} \quad (11)$$

2.5 信息素的更新

2.5.1 局部更新规则

在每次迭代完成后, 本次迭代中获得最优调度的蚂蚁可应用局部更新规则来更新当前的信息素水平. 局部更新规则基于本次迭代所获得的最优调度来完成对信息素的更新. 如果任务 i 被指派到时间窗口 k 内, 则

$$K_{ik} = K_{ik} + Q_L \quad (12)$$

这里, Q_L 表示局部更新阶段信息素的增加水平.

2.5.2 全局更新规则

在蚁群优化的迭代过程中, 获得全局最优调度(自开始迭代到当前迭代过程中所获得的最优调度)的蚂蚁可以应用全局更新规则来更新当前的信息素水平. 全局更新规则基于当前的全局最优调度来完成对信息素的更新. 如果任务 i 被指派到时间窗口 k 内, 则

$$K_{ik} = K_{ik} + Q_G \quad (13)$$

这里, Q_G 表示局部更新阶段信息素的增加水平.

2.5.3 信息素挥发规则

在每次迭代完成后, 当前的信息素将在信息素挥发规则的作用下完成更新. 为了降低蚁群优化算法陷入到局部最优的可能性, 我们将信息素的水平限制到 $[\tau_{\min}, \tau_{\max}]$ 内. 信息素挥发规则的定义如下

$$K_{ik} = \min \{ \tau_{\max}, \max \{ \tau_{\min}, (1 - \rho) K_{ik} \} \} \quad (14)$$

这里, ρ ($0 < \rho < 1$) 表示信息素的挥发系数.

2.6 终止准则

终止准则主要控制混合蚁群优化算法的整体优化流程. 卫星地面站系统任务调度实例都是按照一定的规则随机生成的, 它们的最优调度(解)是未知的. 鉴于此, 我们采用预设最大迭代次数来完成对终止准则的定义.

3 仿真实例

作者随机生成了 40 个典型的卫星地面站系统任务规划实例来验证本文方法的有效性. 仿真实例的生成规则如下:

- 1) 任务数目 n 取值为 100、200、300、400 和 500.
- 2) 时间窗口数目 m 取值为 4、6、8 和 10.
- 3) 调度时间选为 0~1000 秒.
- 4) 任务权重选择考虑两种情况: a) 所有任务具有相同权重; b) 任务权重从 $[1, 50]$ 中均匀选取.
- 5) 任务执行时间的选取. 首先计算所有任务的平均执行时间 \bar{t} (式 (15)); 然后选取每个任务的基准执行时间 \bar{t}_i (表 1); 最后从 $[0.8\bar{t}_i, 1.2\bar{t}_i]$ 中均匀选取每个任务在每个时间窗口内需要的执行时间.

$$\bar{t} = (1000 \times m) / n \quad (15)$$

6) 任务可用时间窗口的选择. 首先从 $[3, 5]$ 中按均匀分布随机选取执行每个任务的可用时间窗口数目; 然后从所有时间窗口集合中按均匀分布选出当前任务的可用时间窗口集合.

7) 从 $[800, 1000]$ 中按均匀分布随机选取每个时间窗口的可用时间长度.

8) 按 n 和 m 组合生成不同规模的仿真实例.

混合蚁群优化算法中参数设置如表 2 所示, 采用 Matlab 语言实现该方法, 并在处理器为奔腾 IV、主频 2.4 GHz、内存 512 M 的电脑中求解这些仿真实例.

表 1 任务基准执行时间的生成概率表

Table 1 Probabilities of the datum executive time

| 概率 | 0.1 | 0.2 | 0.4 | 0.3 |
|-------------|-------------------|-------------------------|-------------------------|--------------------------|
| \bar{t}_i | $[1, 0.5\bar{t}]$ | $[0.5\bar{t}, \bar{t}]$ | $[\bar{t}, 1.5\bar{t}]$ | $[1.5\bar{t}, 2\bar{t}]$ |

表 2 混合蚁群优化算法中的参数设置

Table 2 Parameter setting of the proposed approach

| 符号 | 取值 | 参数的具体含义 |
|-----------------|------|----------------|
| <i>AntSize</i> | 10 | 蚁群的数目 |
| <i>a</i> | 3 | 信息素启发值的权重 |
| <i>b</i> | 5 | 优先级启发值的权重 |
| <i>c</i> | 2 | 总执行时间启发值的权重 |
| τ_0 | 0.10 | 信息素的初始化水平 |
| τ_{\max} | 1.0 | 信息素水平的上限 |
| τ_{\min} | 0.01 | 信息素水平的下限 |
| ρ | 0.02 | 信息素挥发系数 |
| Q_L | 0.02 | 局部更新阶段信息素的增量 |
| Q_G | 0.10 | 全局更新阶段信息素的增量 |
| <i>Max-Iter</i> | 20 | 预设的蚁群优化的最大迭代次数 |

采用混合蚁群优化算法求解 40 个仿真实例的优化结果如表 3(见下页)所示. 我们采用混合方法对每个相同的仿真实例均进行了 10 次优化. 从优化结果来看, 每个仿真实例的 10 次优化结果基本上非常靠近(所有实例的优化结果与优化结果均值之间的平均误差仅为 0.82%). 从优化时间来看, 每个仿真实例的 10 次优化时间基本上比较接近(所有实例的优化时间与优化时间均值之间的平均误差为 2.97%).

混合蚁群优化算法的时间复杂度的分析结果如图 3 所示. 对于不同的情形, 随着任务数目的增加, 计算时间近乎线形增加. 同时, 最复杂实例的计算时间不超过 1800 秒(30 分钟), 本方法在计算时间消耗方面是可以接受的. 进一步的实验结果还表明: 1) 在时间窗口资源有限的情况下, 优先等级较高的任务都得到了执行, 优先等级较低的任务仅有部分被执行; 2) 大部分任务都在最优或次最优(执行该任务所消耗的时间最少或较少)时间窗口内完成; 3) 各个时间窗口的利用率非常高, 并且各个时间窗口内的负载(执行任务耗费的总时间)也比较均衡. 从以上的优化结果及其分析情况来看, 混合蚁群优化算法能快速有效地求解卫星地面站系统任务调度问题.

表 3 仿真实例的优化结果

Table 3 Experimental results of testing cases

| SN | n-m | Result | Time | SN | n-m | Result | Time |
|----|--------|--------|--------|----|--------|--------|--------|
| 1 | 100-4 | 2582 | 14.1 | 21 | 100-4 | 82 | 16.1 |
| 2 | 200-4 | 5037 | 86.1 | 22 | 200-4 | 180 | 89.1 |
| 3 | 300-4 | 7315 | 274.1 | 23 | 300-4 | 255 | 336.1 |
| 4 | 400-4 | 10450 | 599.2 | 24 | 400-4 | 362 | 674.6 |
| 5 | 500-4 | 12392 | 1174.2 | 25 | 500-4 | 488 | 1357.9 |
| 6 | 100-6 | 2498 | 16.5 | 26 | 100-6 | 88 | 16.5 |
| 7 | 200-6 | 5279 | 95 | 27 | 200-6 | 171 | 107.1 |
| 8 | 300-6 | 7413 | 303.2 | 28 | 300-6 | 263 | 319.6 |
| 9 | 400-6 | 9771 | 678.1 | 29 | 400-6 | 350 | 1012.3 |
| 10 | 500-6 | 12542 | 1319.2 | 30 | 500-6 | 431 | 1603.5 |
| 11 | 100-8 | 2526 | 18 | 31 | 100-8 | 80 | 17 |
| 12 | 200-8 | 4839 | 102.9 | 32 | 200-8 | 162 | 109.4 |
| 13 | 300-8 | 7297 | 288.2 | 33 | 300-8 | 248 | 317.9 |
| 14 | 400-8 | 10373 | 653.3 | 34 | 400-8 | 346 | 825.6 |
| 15 | 500-8 | 12967 | 1322.6 | 35 | 500-8 | 441 | 1806.7 |
| 16 | 100-10 | 2319 | 18.3 | 36 | 100-10 | 73 | 19.3 |
| 17 | 200-10 | 5157 | 108.3 | 37 | 200-10 | 162 | 110.8 |
| 18 | 300-10 | 7011 | 305.4 | 38 | 300-10 | 247 | 328.7 |
| 19 | 400-10 | 10251 | 698.6 | 39 | 400-10 | 346 | 976.1 |
| 20 | 500-10 | 12479 | 1301.5 | 40 | 500-10 | 442 | 1761.3 |

注:

- 1) “SN”表示实例序号;“n-m”表示实例规模;“Result”表示计算得到的优先级之和;“Time”表示计算时间(单位:秒)。
2) 在前 20 个实例中,每个任务的权重从 1 到 50 中均匀选取;在后 20 个实例中,每个任务的权重均为 1。

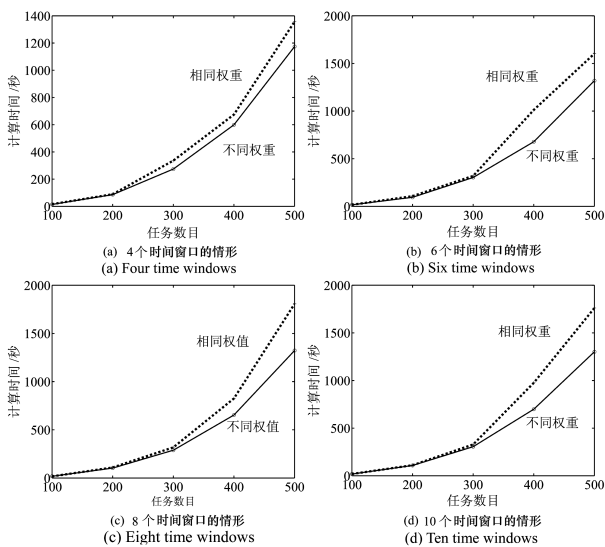


图 3 混合蚁群优化算法的时间复杂度分析

Fig. 3 Analysis of time complexity

4 结论

本文提出了一种有效求解卫星地面站系统任务规划问题的基于蚁群优化算法和导向局部搜索方法的混合优化方法.实例结果表明,该混合方法能有效地求解卫星地面站系统任务调度问题。

今后的研究方向:1)进一步考虑卫星地面站系统任务调度问题的约束条件,如任务之间的逻辑约

束、天线转换时间的随机性约束等;2)设法解决卫星地面站系统的动态任务调度问题,即在调度过程中,可以随机地增删任务;这都有待于进一步的探索和研究。

References

- Liu Yang, Chen Ying-Wu, Tan Yue-Jin. Mission planning method of the satellite ground station based on the greedy algorithm. *Systems Engineering and Electronics*, 2003, **25**(10): 1239–1241 (刘洋, 陈英武, 谭跃进. 基于贪婪算法的卫星地面站系统任务规划方法. *系统工程与电子技术*, 2003, **25**(10): 1239–1241)
- Liu Yang, Chen Ying-Wu, Tan Yue-Jin. The method of mission planning of the ground station of satellite based on dynamic programming. *Chinese Space Science and Technology*, 2005, **25**(1): 47–50 (刘洋, 陈英武, 谭跃进. 卫星地面站系统任务调度的动态规划方法. *中国空间科学技术*, 2005, **25**(1): 47–50)
- Liu Yang, Chen Ying-Wu, Tan Yue-Jin. Modeling and solution of a multi-resources dynamic scheduling problem with time windows. *Systems Engineering*, 2004, **22**(6): 8–11 (刘洋, 陈英武, 谭跃进. 一类含时间窗口的多资源动态调度问题的建模与求解方法. *系统工程*, 2004, **22**(6): 8–11)
- He Ren-Jie, Tan Yue-Jin. Apply constraint satisfaction to optimal allocation of satellite ground station resource. *Computer Engineering and Applications*, 2004, **40**(18): 229–232 (贺仁杰, 谭跃进. 基于约束满足的卫星地面站资源优化分配问题研究. *计算机工程与应用*, 2004, **40**(18): 229–232)
- Dorigo M, Stutzle T. *Ant Colony Optimization*. Cambridge, USA: MIT Press, 2004
- Voudouris C, Tsang E. Guided local search and its application to the traveling salesman problem. *European Journal of Operational Research*, 1999, **113**(2): 469–499
- Blum C. Beam-ACO-hybridizing ant colony optimization with beam search: an application to open shop scheduling. *Computers and Operations Research*, 2005, **32**(6): 1565–1591
- Tsang E, Voudouris C. Fast local search and guided local search and their application to British Telecom's workforce scheduling problem. *Operations Research Letters*, 1997, **20**(3): 119–127
- Li Ju-Fang, Tan Yue-Jin. The guided local search method for a class of work-force scheduling problems. *Systems Engineering*, 2004, **22**(9): 87–91 (李菊芳, 谭跃进. 一类人员调配问题的引导式局部搜索算法. *系统工程*, 2004, **22**(9): 87–91)



邢立宁 国防科学技术大学博士研究生. 主要研究方向为任务规划和人工智能. 本文通信作者.

E-mail: xln_2002@126.com

(XING Li-Ning Ph.D. candidate at National University of Defense Technology. His research interest covers mission planning and artificial intelligence. Corresponding author of this paper.)



陈英武 国防科学技术大学教授. 主要研究方向为管理决策和人工智能.

E-mail: ywchen@nudt.edu.cn

(CHEN Ying-Wu Professor at National University of Defense Technology. His research interest covers management decision and artificial intelligence.)