混合蚁群优化算法求解卫星数传调度问题*

孙 兵¹,陈祥国²

(1. 广东海洋大学 信息学院, 广东 湛江 524088; 2. 国防科学技术大学 信息系统与管理学院, 长沙 410073)

摘 要:为了求解卫星数传调度问题,提出了混合蚁群优化算法。算法设计了基于任务数传操作的解构造图,提出了基于解构造图的任务调度序列和资源分配序列概率决策模型,采用基于随机加权的混合策略综合利用问题的启发式信息。算法通过基于混沌变异的列信息素向量更新策略增强解构造的多样性,通过具有补偿机制的全局信息素更新策略来保证算法的收敛性。利用 STK 工具设计了五个调度场景,并利用计算机生成各场景的数传任务。仿真实验结果表明,该算法是可行、有效的,收敛性和解多样性较好。
 关键词:卫星数传;调度;蚁群优化;混合算法
 中图分类号: TP301.6 文献标志码: A 文章编号: 1001-3695(2012)11-4064-05

doi:10.3969/j.issn.1001-3695.2012.11.016

Hybrid ant colony optimization algorithm for satellite data transmission scheduling

SUN Bing1 , CHEN Xiang-guo2

(1. College of Information, Guangdong Ocean University, Zhanjiang Guangdong 524088, China; 2. College of Information Systems & Management, National University of Defense Technology, Changsha 410073, China)

Abstract: For solving satellite data transmission scheduling problem, this paper proposed a hybrid ant colony optimization algorithm. In the algorithm, it designed solution construction graph based on tasks data transmission operation, put forward probability decision-making models of task scheduling sequence and resource allocation sequence based on the construction graph, adopted heuristics mixed using strategies based on the stochastic weighted for utilizing heuristics integratedly. The algorithm strengthen the diversity of the constructed solutions through variation column pheromones vector update strategy based on chaos, and adopted the global pheromones update strategy with compensation mechanism to guarantee convergence. Using STK tool design five scheduling scenes, and generating data transmission tasks on the scenes using computer. The simulation results show that, the proposed algorithm is feasible and effective, convergence and diversity is better.

Key words: satellite data transmission; scheduling; ant colony optimization (ACO); hybrid algorithm

0 引言

卫星数传调度问题(satellite data transmission scheduling problem, SDTSP)是具有 NP-hard 性质的组合优化问题^[1],主要 是为卫星的数传任务分配满足调度约束的地面资源(地面站及 数传天线)和时间窗口,以满足设定的调度目标。随着卫星数量 的增多,数传任务大量增长,有限的地面资源和时间窗口难以满 足所有数传任务的需要,这使得卫星数传调度问题与 SRS(satellite range scheduling)问题^[2]都属于过订购问题^[3]。SRS 研究 中,提出了借鉴图着色技术(graph colouring techniques)的启发 式算法^[4]。类似问题中,还有关于多卫星侦察调度^[5]、多站多 星任务调度^[6]、地球观测卫星星座调度^[7]的研究。

在 SDTSP 研究中,曾提出基于数传任务可能冲突集分割 的遗传算法^[8]和基于任务综合优先度的启发式算法^[9]。前者 更适于求解较大规模的卫星数传调度问题,后者则在较小规模 的问题中能够获得较满意的结果。鉴于启发式算法求解大规 模卫星数传调度问题难以获得更满意的结果,遗传算法不能充 分利用基于问题的启发式信息,使得搜寻问题最优解的能力受 到局限。本文将基于蚁群优化(ACO)算法^[10]求解卫星数传调 度问题。ACO 算法是一种构造元启发式(constructive metaheuristics)算法^[11],其解构造过程中的概率决策模型扩大了解 搜索空间,信息素更新的正反馈机制使算法能够向最优解收 敛,对基于问题的启发式信息的充分利用,增强了算法寻优能 力。ACO 算法在 TSP^[12]、车辆调度^[13]、排列流水调度^[14]、车 间调度^[15]、电力系统代维护调度^[16]、天然气需求评估^[17]等问 题中获得了 广泛应用。

本文将根据卫星数传调度的问题描述(problem representation)和解构造(solution construction)方式,建立基于数传操作 的解构造图模型,并提出相应的混合蚁群优化算法,最后对模 型和算法进行仿真实验分析。

1 卫星数传调度问题

将卫星数传调度问题定义为一个七元组 P = (TimeLine, S,R,TW,T,Ω,O)。其中:

TimeLine 为规划调度起止时间[*ST*,*ET*],*ST*、*ET*分别为调度开始时间和结束时间;

S为在轨运行的卫星集合,其空间位置由卫星半长轴、偏心率、轨道倾角、近地点幅角、升交点赤径、真近点角等六个轨

收稿日期: 2012-04-16; 修回日期: 2012-05-26 基金项目: 湛江市科技攻关计划资助项目(2012C3106010)

作者简介:孙兵(1974-),男,重庆人,讲师,硕士,主要研究方向为智能算法、智能信息处理(sunzjou@yahoo.com.cn);陈祥国(1979-),男,工程师,博士,主要研究方向为智能算法、资源优化与任务调度.

道根数确定, $S = \{S_1, S_2, \dots, S_{|S|}\};$

R为正常工作的地面资源集合,其地理位置由经纬度、海拔高度等参数确定, $R = \{R_1, R_2, \dots, R_{|R|}\}$;

TW 为卫星与地面资源之间的可见时间窗口集合,*TW* = $\{TW_1, TW_2, \dots, TW_{|TW|}\}$,每个时间窗口的开始时间、结束时间和持续时间可由 STK 工具根据卫星运行轨道和地面资源的地理位置确定;

T为所有卫星的数传任务集, $T = \{T_1, T_2, \dots, T_{|T|}\}$,数传任务具有优先级、最短持续时间、最早开始时间、最迟结束时间以及数传频段等要求;

Ω为调度约束集合,是为数传任务分配地面资源和可见时间窗口必须满足的约束条件,主要包括地面资源频段符合任务数传频段、地面资源具有足够的设备状态切换时间,地面资源 在同一时刻只能为一个数传任务提供数传服务,数传任务在同一时刻只能利用一个地面资源的可见时间窗口;

0为调度优化目标。本文以最大任务调度收益率为调度 目标,如式(1)所示,任务调度收益与任务需求收益的比值即 为任务调度收益率。该目标取决于成功调度的任务数量以及 任务调度收益。

Maxmize
$$O = f(x) = (\sum_{i=1}^{|T|} p_i x_i) / (\sum_{i=1}^{|T|} p_i)$$
 (1)

其中: p_i 为任务 T_i 的调度收益,考虑任务优先级、任务最短持续时间等属性,采用基于信息熵的多属性决策方法确定^[18]。 $x_i \in \{0,1\}$ 为决策变量, $x_i = 0$ 表示任务 T_i 调度失败,即未满足 调度约束; $x_i = 1$ 表示任务 T_i 调度成功,即满足相关调度约束。

2 混合蚁群优化算法

2.1 蚁群优化的解构造图

卫星数传调度过程中,需要在数传任务与地面资源之间建 立数传链路并进行数传服务,这种数传服务关系的建立成为任 务数传操作。由于一个任务可由多个地面资源提供数传服务, 一个地面资源可为多个任务提供数传服务,所以通过确定任务 与资源之间的数传操作对应关系,即可获得一个满足约束条件 的调度方案。

为了构造人工蚁群的生存环境,将每个任务与每个地面资 源之间可能建立的数传操作抽象为节点,构造基于任务数传操 作的解构造图(solution construction graph based on tasks data transmission operation), $G = (N, A, E, \Gamma, \Psi, \Omega)$,简称任务数传 操作图。如图1所示,任意两列之间通过无向实线边连接,将 实线边称为任务边。每列的任意两个节点间通过无向虚线弧 连接,将虚线弧称为资源弧。

图 G中,m = |R|,n = |T|。其中:

N 表示图 *G* 中的节点集合,*N* = {*N_{ij}* | *i* = 1,2,…, |*R*|;*j* = 1, 2,…, |*T*|},每个节点 *N_{ij}* ∈ *N* 表示地面资源 *R_i* ∈ *R* 与任务 *T_j* ∈ *T* 可能建立的数传操作;

A 表示资源弧,为图 G 中每列各节点之间的无向弧集合, $A = \{ (N_{k}, N_{k}) | k, l \in \overline{|R|}; k \neq l; j \in \overline{|T|} \};$

E表示任务边,为图 G中每列之间的无向边集合, $E = {(N_{*p}, N_{*q}) | p, q \in \overline{|T|}; p \neq q}, N_{*p} 和 N_{*q}$ 分别代表第 p 列和 第 q 列;

 Γ 表示图 G中与节点集合 N 相关联的信息素分布, Γ =

 $\{\tau_{ij} | i \in |R|, j \in |T|\}$ 。信息素 τ_{ij} 分布在节点 N_{ij} 上,表示任务 T_i 在地面资源 R_i 上建立数传操作的期望;

 Ψ 表示 *G*中的列信息素分布, $\Psi = \{\tau_{*j} | j \in \overline{|T|}\}$, 信息素 τ_{*j} 分布在列 N_{*j} 上, 表示优先调度任务 T_j 的期望;

Ω表示人工蚂蚁在图 G 中游历的约束条件;蚂蚁从某个虚 拟节点 N₀ 出发,首先沿任务边选择某列,然后在被选择的列 中沿资源弧遍历该列的每个节点,任何一个节点只遍历一次, 直到遍历了图 G 中每一列的每一个节点。



2.2 解构造的概率决策模型

为了便于构造解,人工蚁群将在图1中首先构造列遍历序列,即确定数传任务的调度序列,然后构造每列中的节点序列, 即确定每个数传任务的资源分配序列。

2.2.1 任务调度序列概率决策模型

设 k = 1, 2, ..., |T|,人工蚂蚁从虚拟节点 N_0 出发, $\pi(N) =$ { N_0 }, $\pi_0 = N_0$,第k - 1步时,已构造列遍历序列 $\pi(N) =$ { N_0 , $\pi(N_{*p}), \pi(N_{*q}), ..., \pi(N_{*i})$ }, $\pi_1 = \pi(N_{*p}), \pi_2 = \pi(N_{*q}),$ $\pi_{k-1} = \pi(N_{*i}), p, q, i \in \overline{|T|}, p \neq q \neq i_0$,人工蚂蚁在第k步时所 选择列 π_k ,即从列 N_{*i} 经过任务边(N_{*i}, N_{*j})选择列 N_{*j} 由式 (2)确定的伪随机比例概率决策模型(pseudo random proportional probability decision-making model)决定。

$$\pi_{k} = \begin{cases} \underset{N_{*j} \in J(N_{*i})}{\arg \max} \left(\left[\tau_{*j}(t) \right]^{\alpha} \left[\eta_{*j}(t) \right]^{\beta} \right) & \text{if } q < q_{0} \\ \\ S & \text{otherwise} \end{cases}$$
(2)

其中: $\tau_{*j}(t)$ 表示在t时刻(每次算法迭代),人工蚂蚁第k步 选择列 N_{*j} 时, N_{*j} 上分布的信息素大小; $\eta_{*j}(t)$ 表示在t时刻 人工蚂蚁第k步选择列 N_{*j} 时所利用的启发式信息大小,由基 于随机加权的启发式信息利用策略确定;参数 α,β 分别表示 信息素浓度和启发式信息的权重;伪随机比例参数 $q_0 \in [0,1]$ 决定了人工蚂蚁在列选择过程中对贪婪搜索和随机探索的倾 向性选择;S表示根据式(3)的概率分布选择的列。当区间[0,1]内均匀分布的随机数 $q < q_0$ 时,人工蚂蚁以概率 q_0 贪婪选 择 $[\tau_{*j}(t)]^{\alpha}[\eta_{*j}(t)]^{\beta}$ 值最大的列;否则,以概率 $1-q_0$ 根据 式(3)所示的概率随机选择节点 S_o

$$p(\pi_{k} = N_{*j} | \pi_{k-1} = N_{*i}, t) =$$

$$\begin{cases} \frac{[\tau_{*j}(t)]^{\alpha} [\eta_{*j}(t)]^{\beta}}{\sum_{N_{*l} \in J(N_{*i})} [\tau_{*l}(t)]^{\alpha} [\eta_{*l}(t)]^{\beta}} & \text{if } N_{*j} \in J(N_{*i}) \\ 0 & \text{otherwise} \end{cases}$$

$$(3)$$

其中:*J*(*N*_{*i})表示人工蚂蚁在第*k*步列选择时,从列*N*_{*i}出发 可选择的列集合,并由式(4)确定。

$$J(N_{*i}) = \{N_{*i} | N_{*i} \subseteq N, N_{*i} \notin \pi(N)\}$$
(4)

2.2.2 资源分配序列概率决策模型

设l=1,2,...,|GR|,人工蚂蚁第l-1步时已构造列节点 遍历序列 $\pi(N_{*j}) = \{N_{jj}, N_{ij}, ..., N_{kj}\}, \pi_1(N_{*j}) = N_{jj}, \pi_2(N_{*j}) =$ $N_{ij}, \pi_{l-1}(N_{*j}) = N_{kj}, p, q, k \in \overline{|GR|}, p \neq q \neq k$ 。设 $\pi_l(N_{*j})$ 为人 工蚂蚁在列节点遍历中第l步所选择的节点,即从节点 N_{kj} 经 过资源弧 (N_{kj}, N_{ij}) 选择节点 N_{ij} ,将由式(5)确定的伪随机比例 概率决策模型决定。

$$\pi_{l}(N_{*j}) = \begin{cases} \underset{N_{ij} \in J(N_{kj})}{\arg \max} \left(\left[\tau_{ij}(t) \right]^{\alpha} \left[\eta_{ij}(t) \right]^{\beta} \right) & \text{if } q < q_{0} \\ S' & \text{otherwise} \end{cases}$$
(5)

其中: $\tau_{ij}(t)$ 表示在t时刻(每次算法迭代)人工蚂蚁在列 N_{*j} 中 遍历节点时,第l步选择的节点 N_{ij} 上分布的信息素大小: $\eta_{ij}(t)$ 表示在t时刻人工蚂蚁第l步选择节点 N_{ij} 时,所利用的启发式 信息大小由基于随机加权的启发式信息利用策略确定;参数 α 、 β 分别表示信息素浓度和启发式信息的权重;伪随机比例参 数 $q_0 \in [0,1]$ 决定了人工蚂蚁在列选择过程中,对进行贪婪搜 索和随机探索的倾向性选择;S'表示根据式(6)的概率分布选 择的某个节点; $J(N_{kj})$ 表示人工蚂蚁在第l步列节点遍历时, 节点 N_{ki} 的邻域节点集合, $J(N_{kj}) \subseteq N_{*io}$

当均匀分布的随机数 $q < q_0$ 时, $q \in [0,1]$, 人工蚂蚁以概 率 q_0 贪婪选择 $[\tau_{ij}(t)]^{\alpha} [\eta_{ij}(t)]^{\beta}$ 值最大的节点; 否则, 以概 率 $1 - q_0$ 根据式(6) 随机确定 S'。

$$p \left\{ \pi_{l}(N_{*j}) = N_{ij} | \pi_{l-1}(N_{*j}) = N_{kj}, t \right\} = \left\{ \frac{\left[\tau_{ij}(t) \right]^{\alpha} \left[\eta_{ij}(t) \right]^{\beta}}{\sum_{N_{hj} \in J(N_{hj})} \left[\tau_{hj}(t) \right]^{\alpha} \left[\eta_{hj}(t) \right]^{\beta}} & \text{if } N_{ij} \in J(N_{kj}) \right.$$

$$(6)$$

2.3 启发式信息混合利用策略

为了在任务调度概率决策模型和资源分配概率决策模型 中综合利用各种启发式信息,使算法不至于因为单一利用某种 启发式信息而影响寻优方向。由于不能准确判定各类启发式 信息的权重,以及各类启发式信息对算法性能的影响程度,通 过随机加权混合利用各类启发式信息,将避免单纯利用某类启 发式信息对算法性能的限制。

设 $TH = \{TH_1, TH_2, TH_3, TH_4\}$ 为任务调度启发式信息,本 文考虑基于任务开始时间、任务收益属性、任务调度灵活度和 调度冲突度四种任务调度启发式信息;设 $RH = \{RH_1, RH_2, RH_3, RH_4\}$ 为资源分配启发式信息,本文考虑基于资源可见窗 口时间点、资源优先级、资源可见窗口持续时间和资源可见窗 口冲突等四种资源分配启发式信息^[18]。

启发式信息的混合利用策略如下:

a)每只人工蚂蚁开始解构造前,分别为 $TH_{R}H$ 生成一组 (0,1)内的随机数 $r = \{r_1, r_2, r_3, r_4\}$ 和 $e = \{e_1, e_2, e_3, e_4\}$ 。

b)分别计算随机权重 $\lambda_i = \frac{r_i}{\sum_{j=1}^4} r_j, \gamma_i = \frac{e_i}{\sum_{j=1}^4} e_j$,并保存在 *TH*、*RH*的权重向量 $\lambda = \{\lambda_1, \lambda_2, \lambda_3, \lambda_4\}$ 和 $\gamma = \{\gamma_1, \gamma_2, \gamma_3, \gamma_4\}$ 中。

c)人工蚂蚁进行列选择时,利用随机权重和 $\sum_{i=1}^{4} \lambda_i \cdot TH_i$ 计算列选择的任务调度启发式信息;进行每列的节点遍历时, 利用 $\sum_{i=1}^{4} \gamma_i \cdot RH_i$ 计算列节点遍历的资源分配启发式信息。

2.4 信息素混合更新策略

2.4.1 基于混沌变异的列信息素向量更新

任务调度序列概率决策模型中,列信息素向量中存储的信

息素对计算决策概率十分重要。如果能够对列信息素向量作 足够的扰动,那么就能够使蚁群优化算法避免在构造任务调度 序列时过早陷入局部最优。由于混沌(chaos)是系统的一种整 体行为^[19,20],混沌现象具有随机性、遍历性、规律性等特点,因 此可利用混沌的随机性使算法具有跳出局部最优的能力,利用 混沌的遍历性使算法到达全局最优任务调度序列的附近。

人工蚁群完成一次算法迭代的解构造之后,应用常见的 Logistic 混沌映射对列信息素执行如式(7)的混沌操作,当 $a = 4 \sqrt{\tau_{*i}}(0) < 1$ 时,Logistic 系统处于完全混沌状态^[20]。

$$\tau_{*i}(t+1) = a\tau_{*i}(t) \{1 - \tau_{*i}(t)\}$$
(7)

通过式(7)对列信息素向量的混沌变异,将完全扰动其中 的每个信息素值,这样将使得蚁群中的蚂蚁可能构造完全不同 的任务调度序列,从而增强算法构造任务调度序列的多样性。 2.4.2 具有补偿机制的全局信息素更新策略

每次算法迭代后,根据式(8)进行图 G 中各节点的信息素 更新。

$$\tau_{ij}(t+1) = (1 - \rho_g)\tau_{ij}(t) + \rho_g \Delta \tau_{ij}(t)$$
(8)

式(8)中, $\Delta \tau_{ij}(t)$ 由式(9)确定。

$$\Delta \tau_{ij}(t) = \lambda_{is} \Delta \tau_{ij}^{is}(t) + \lambda_{bs} \Delta \tau_{ij}^{bs}(t)$$
(9)

式(9)中, λ_{is} , $\lambda_{bs} \in [0,1]$ 为权重系数, $\lambda_{is} + \lambda_{bs} = 1.0$; $\tau_{ij}(0) = \tau_0, \Delta \tau_{ij}^{is}(t)$ 是根据迭代最优解 s^{is} 确定的信息素遗留量,由式(10)确定。

$$\Delta \tau_{ij}^{is}(t) = \begin{cases} f(s^{is}) & \text{if } x_j = 1, y_{ij} = 1\\ 1 - f(s^{is}) & \text{if } x_j = 1, y_{ij} = 0\\ \frac{k}{|N_{*j}|} \cdot f(s^{is}) & \text{if } x_j = 0, y_{ij} = 1, \\ \pi_k(N_{*j}) = N_{ij} & (10)\\ \frac{|N_{*j}| - k}{|N_{*j}|} \cdot f(s^{is}) & \text{if } x_j = 0, y_{ij} = 0, \\ \pi_k(N_{*i}) = N_{ii} & (10) \end{cases}$$

其中: x_j 为决策变量, $x_j = 1$ 表示任务 T_j 调度成功, $x_j = 0$ 表示 任务 T_j 调度失败; y_i 也为决策变量, $y_{ij} = 1$ 表示任务 T_j 在地面 资源 R_i 上建立了数传操作, $y_{ij} = 0$ 表示任务 T_j 在地面资源 R_i 上没有建立数传操作。显然,式(10)对图 G中的每个节点都 遗留了一定信息素,通过信息素遗留的少量补偿,可避免某些 节点上的信息素挥发殆尽,而使算法过早陷入局部最优。

式(10)中, $\Delta \tau_{i}^{bs}(t)$ 是根据至今最优解 s^{bs} 确定的信息素遗 留量,由式(11)确定。

$$\Delta \tau_{ij}^{bs}(t) = \begin{cases} f(s^{bs}) \frac{DTW_{ij}^{dur}}{T_j^{dur}} & \text{if } x_j = 1, y_{ij} = 1\\ 0 & \text{otherwise} \end{cases}$$
(11)

其中: DTW_{ij}^{dur} 表示任务 T_j 在地面资源 R_i 上建立数传操作的持续时间, T_j^{dur} 为任务 T_j 的最短持续时间。式(11)是对信息素 遗留的强化,通过 $\frac{DTW_{ij}^{dur}}{T_j^{fur}}$ 来使算法在下一次迭代中能够对列节点遍历有倾向性选择。

2.5 算法基本流程描述

卫星数传调度混合蚁群优化算法描述如下:

a)设置算法参数,即蚁群规模 N_{ants} 、最大算法迭代次数 NC_{max} 、伪随机比例参数 q_0 、信息素浓度权重 α 、启发式信息权 重 β 、初始信息素浓度 τ_0 和全局信息素挥发系数 $\rho_{\mu o}$ 。

b)构造任务数传操作图 *G*。图中节点集合 *N*, |*N*| = |*R*|×|*T*|,初始化信息素矩阵 $\Gamma = \{\tau_{ij}(0) = r | i = 0, 1, 2, \cdots, |R| - 1; j = 0, 1, 2, \cdots, |T| - 1\}$,构造图 *G* 中各列的信息素向量 $\Psi = \{\tau_{*j} = r | j = 0, 1, 2, \cdots, |T| - 1\}$, r 为(0,1]均匀分布的随 机数。

c)计算启发式信息^[18],包括任务集*T*的任务调度启发式 信息*TH* = {*TH*₁,*TH*₂,*TH*₃,*TH*₄},以及任务*T_j* \in *T* 与每个具有 可见时间窗口的地面资源的资源分配启发式信息*RH* = {*RH*₁, *RH*₂,*RH*₃,*RH*₄}。

d)算法迭代,置至今最优解 $s^{b} \leftarrow \varphi$,算法迭代计数器 NC $\leftarrow 0$ 。

e) 置迭代最优解 $s^{ii} \leftarrow \varphi$ 。

f)人工蚁群解构造,置蚂蚁计数器 $k \leftarrow 0$,蚁群构造的可行 解集 $S_{aut} \leftarrow \varphi$:

(a) $k \leftarrow k + 1$,初始化蚂蚁 k,将其置于 N_0 , $\pi^k(N) \leftarrow \varphi$, $s_k \leftarrow \varphi$;

(b)为*TH*、*RH*生成一组(0,1)间的随机数,并计算其权重 向量 $\lambda = \{\lambda_1, \lambda_2, \lambda_3, \lambda_4\}$ 和 $\gamma = \{\gamma_1, \gamma_2, \gamma_3, \gamma_4\}$;

(c)根据由式(2)和(3)确定的列选择概率决策模型,确定 当前列 N_{*i} ,任务调度启发式信息由随机权重和 $\sum_{i=1}^{4} \lambda_i \cdot TH_i$ 确定;

(d)根据式(5)和(6)确定的列节点遍历概率决策模型,构 造列 N_{*j} 的节点遍历序列 $\pi^{k}(N_{*j})$,节点遍历的资源分配启发 式信息由随机权重和 $\sum_{i=1}^{4}\gamma_{i} \cdot RH_{i}$ 确定;

(e)将 $\pi^{k}(N_{*j})$ 顺序插入 $\pi^{k}(N)$ 。重复前两步,遍历所有 列中的节点;

(f) 调度解构造,根据 $\pi^{k}(N)$ 中 $\pi^{k}(N_{*j})$ 的顺序确定任务 调度序列 $\pi^{k}(T)$,根据 $\pi^{k}(N_{*j})$ 中的节点顺序,确定每个任务 的资源分配序列 $\pi^{k}(R_{*j})$,按照 $\pi^{k}(T)$ 中的顺序,为每个任务 进行解成分构造,即根据 $\pi^{k}(R_{*j})$ 的顺序为任务确定数传时间 窗口,直到满足任务最短持续时间要求,或者已无资源可用,从 而形成可行调度解 s_{k} ,并置 $S_{att} \leftarrow S_{att} \cup \{s_{k}\}$;

(g)如果 $k < N_{ants}$,则转(a);否则转步骤 g)。

g)根据调度目标指标评价函数评价 S_{ant}中的解,并确定迭 代最优解 s^{is}和至今最优解 s^{bs}。

h)对列信息素向量按式(7)执行混沌操作。

i)根据s^{is},由式(8)~(11)执行全局信息素更新。

j) NC←NC + 1, 如果 NC < NC_{max}, 则转步骤 e); 否则输出并 评价 s^{bs} 。

3 仿真实验分析

3.1 仿真实验设计

利用 STK 工具设计五个调度场景^[18],即 5 站 20 星、5 站 30 星、5 站 40 星、5 站 50 星、5 站 60 星场景各一个。场景中各 地面站有两根数传天线,各卫星有一根数传天线。所有数传天 线频段相同,地面站地理位置参数均相同。所有卫星采用太阳 同步轨道,同高度轨道面上均匀布置五颗卫星。最低轨道高度 370 km,每隔 30 km 一个轨道面。利用 STK 工具生成场景预报 流数据,获得各卫星与地面站之间的几何可见时间窗口。通过 计算机生成各调度场景的基准测试数传任务集^[18]。各调度场 景的可见时间窗口冲突及数传任务基本信息如表 1 所示。从 表1中可以看出,随着卫星数量的增多,数传任务规模不断增大,调度场景的可见时间窗口冲突也不断增大,这也意味着问题的求解难度更大。

表1 场景冲突及数传任务

场景	无冲突时间比/%	任务总数/个	任务总收益
5站60星	12.97	637	366.28
5站50星	24.52	526	300.95
5站40星	35.21	420	245.14
5站30星	53.63	316	182.87
5站20星	71.20	209	124.40

3.2 仿真结果分析

对 5 站 40 星调度场景进行仿真实验,分析并确定算法的 最佳参数组合。测试每个参数性能时,按照缺省设置固定其他 参数,算法终止条件为达到最大迭代次数。对于每个参数值, 算法独立运行 30 次,生成 30 个样本,然后利用 MATLAB 生成 的箱形图分析关键参数对算法性能的影响。最终确定算法参 数组合如下 Nants = 25, q_0 = 0.5, α = 5.0, β = 4.0, ρ_s = 0.95, τ_0 = 0.05。

将本文算法及文献[8]算法(遗传算法)每次运行在达到 最大迭代次数 NC_{max} = 100 时终止,针对每个场景实验中各独 立运行 10 次,取其平均结果,文献[9]算法(基于综合优先度 的启发式算法)只运行一次。仿真实验结果如表 2 所示。从 表 2 可以看出,本文在各调度场景中都能获得最好的结果,文 献[9]算法获得的结果最差,说明在较大规模卫星数传调度问 题中,智能算法优于启发式算法,本文提出的算法是可行的。 在仿真实验中还发现,本文算法的运算速度明显快于文献[8] 算法,说明本文提出的算法具有较高的效率。

表2 仿真实验结果

场景	本文算法(平均耗时/s)	文献[8]算法(平均耗时/s)	文献[9]算法
5站60星	0.8612(673.35)	0.799 5(1755.33)	0.811 6
5站50星	0.8992(427.05)	0.8557(1068.94)	0.841 6
5站40星	0.945 1(282.56)	0.911 3(619.05)	0.875 9
5站30星	0.990 3(168.18)	0.955 6(304.35)	0.929 1
5站20星	0.985 4(79.69)	0.979 5(122.51)	0.947 2

图 2 是本文算法针对 5 站 40 星场景某次运行的收敛曲 线,由图 2 可以看出,算法具有较好的收敛性能。图 3 是本文 算法的解多样性曲线,由图 3 可以看出,算法对解构造的多样 性维护较好。



4 结束语

本文提出了求解卫星数传调度的混合蚁群优化算法。为 便于人工蚁群进行解构造,基于任务与资源之间的数传操作对 应关系,提出了基于任务数传操作的解构造图。人工蚁群在图 中通过列遍历构造任务调度序列,通过列节点遍历构造每个任 务的数传资源分配序列,最后根据所构造的序列构造满足调度 约束的可行解。算法提出了基于随机加权的启发式信息混合 利用策略,以充分利用基于问题的启发式信息。为扩大解搜索 空间并使算法向最优解收敛,提出了基于混沌变异的列信息素 向量更新策略和具有补偿机制的全局信息素更新策略。仿真 实验结果表明,本文提出的算法收敛性和解多样性较好,能获 得优于遗传算法和启发式算法的可行结果,且算法速度快于遗 传算法。

参考文献:

- BARBULESCU L, WATSON J P, WHITLEY L D, et al. Scheduling space-ground communications for the air force satellite control network
 J. Journal of Scheduling,2004,7(1):7-34.
- [2] MARINELLI F, NOCELLA S, ROSSI F, et al. A Lagrangian heuristic for satellite range scheduling with resource constraints [J]. Computers & Operations Research, 2011, 38(11):1572-1583.
- [3] BARBULESCU L, HOWE A E, WHITLEY L D, et al. Understanding algorithm performance on an oversubscribed scheduling application [J]. Journal of Artificial Intelligence Research, 2006, 27 (12):577-615.
- [4] ZUFFEREY N, AMSTUTZ P, GIACCARI P. Graph colouring approaches for a satellite range scheduling problem [J]. Journal of Scheduling,2008,11(4):263-277.
- [5] WU Guo-hua, MA Man-hao, WANG Hui-lin, *et al.* Multi-satellite observation scheduling based on task clustering[J]. Acta Aeronautica et Astronautica Sinica, 2011, 32(7):1275-1282.
- [6] 王万玉,张志强. 多站多星任务调度模型及求解[J]. 电讯技术, 2011,51(4):1-6.
- [7] WANG Pei, REINELT G, GAO Peng, et al. A model, a heuristic and a decision support system to solve the scheduling problem of an earth observing satellite constellation [J]. Computers & Industrial Engineering, 2011, 61(2):322-335.
- [8] 李云峰,武小悦. 遗传算法在卫星数传调度问题中的应用[J].系 统工程理论与实践,2008,28(1):124-131.

(上接第4052页)

从图 2~4 中可以看出, BCC 算法收敛速度较慢, 往往在某 一局部最优点进化长期停滞不前, 经过大量的迭代才能跳出局 部最优, 但又会进入另一个局部最优, 算法的开发能力严重不 足。而 CBCC 因为加入了混沌搜索策略, 提高了细菌移动的遍 历性, 其跳出局部最优的能力大大超过 BCC 算法的跳出局部 最优的能力, 增强了算法的全局寻优能力, 加快了算法的收敛 速度。

5 结束语

本文提出在细菌群体中加入混沌搜索策略,提高细菌群体 中个体的多样性,增强算法的探索开发能力,从而提高了算法 跳出局部最优能力和全局收敛能力。实验结果表明,新算法加 快了算法的收敛速度,寻优速度更快,更准确。

参考文献:

 SIBYLLE D, JARNO M, STEFANO A, et al. Optimization based on bacterial chemotaxis [J]. IEEE Trans on Evolutionary Computa-

- [9] 李云峰,武小悦.基于综合优先度的卫星数传调度算法[J].系统 工程学报,2007,22(6):644-648.
- [10] DORIGO M, STÜTZLE T. Ant colony optimization[M]. Cambridge: MIT Press, 2004.
- [11] CHANDRA M B, BASKARAN R. A survey: ant colony optimization based recent research and implementation on several engineering domain[J]. Expert Systems with Applications, 2012, 39(4):4618-4627.
- [12] YANG Jin-hui, SHI Xiao-hu, MARCHESE M, et al. An ant colony optimization method for generalized TSP problem [J]. Progress in Natural Science, 2008, 18(11):1417-1422.
- [13] FUELLERER G, DOERNER K F, HARTL R F, et al. Ant colony optimization for the two-dimensional loading vehicle routing problem
 [J]. Computers & Operations Research, 2009, 36(3):655-673.
- [14] TAVARES N R F, GODINHO F M. An ant colony optimization approach to a permutational flowshop scheduling problem with outsourcing allowed [J]. Computers & Operations Research, 2011, 38 (9):1286-1293.
- [15] PANAHI H, TAVAKKOLI-MOGHADDAM R. Solving a multi-objective open shop scheduling problem by a novel hybrid ant colony optimization[J]. Expert Systems with Applications, 2011, 38 (3): 2817-2822.
- [16] FETANAT A, SHAFIPOUR G. Generation maintenance scheduling in power systems using ant colony optimization for continuous domains based 0-1 integer programming[J]. Expert Systems with Applications, 2011, 38(8):9729-9735.
- [17] DURAN T M, TOKSARI M. Ant colony optimization based on multisearching to estimate the natural gas demand: case of turkey [J].
 Journal Energy, Exploration & Exploitation, 2012, 30 (2): 223-238.
- [18] 陈祥国.卫星数传调度的蚁群优化模型及算法研究[D].长沙: 国防科学技术大学,2010.
- [19] PAINTER K J, HILLEN T. Spatio-temporal chaos in a chemotaxis model[J]. Physica D,2011,240(4):363-375.
- [20] GHARAJEDAGHI J. Systems thinking: managing chaos and complexity: a platform for designing business architecture [M]. 3rd ed. Oxford: Butterworth-Heinemann, 2011.

tion,2002,6(1):16-29.

- [2] 李威武,王慧,邹志君,等.基于细菌群体趋药性的函数优化方法
 [J].电路与系统学报,2005,10(1):58-63.
- [3] WANG Zhi-qiang, ZHANG Li-xin, FAN Y. A study on bacterial colony chemotaxis algorithm and simulation based on differential strategy [J]. International Journal of Modelling, Identification and Control, 2010, 9(1):136-143.
- [4] 陈雷,张立毅,郭艳菊,等.基于细菌群体趋药性的有序盲信号分 离算法[J].通信学报,2011,32(4):77-85.
- [5] LI Guo-qing, LIAO Hai-liang, CHEN Hou-he. Improved bacterial colony chemotaxis algorithm and its application in available transfer capability[C]//Proc of the 5th International Conference on Natural Computation. Washington DC: IEEE Computer Society, 2009: 286-291.
- [6] 刘文霞,刘晓茹,张建华,等.基于微分进化和混沌迁移的细菌群 体趋药性算法[J].控制理论与应用,2009,26(4):353-357.
- [7] 单梁,强浩,李军,等.基于 Tent 映射的混沌优化算法[J].控制与 决策,2005,20(2):179-183.
- [8] 陈如清,俞金寿.混沌粒子群混合优化算法研究与应用[J].系统 仿真学报,2008,20(3):685-689.

b)随机初始化 BCC 群和 C 群中各个细菌的位置,并根据 初始化的位置采用适应值函数计算 BCC 群和 C 群中各个细菌 的适应值。

c)对 BCC 群中的每个细菌按照 BCC 算法在感知区域内 趋药移动并根据条件进行迁徙移动(详见第1章细菌群体趋 药性算法中细菌的移动过程)。

d)对 C 群中的每个细菌按照 Tent 映射产生混沌序列的方法(详见第2章混沌优化算法中 Tent 映射产生混沌序列过程) 进行混沌移动。

e) 替代过程。根据适应值计算函数选择 C 群中最优的 W 个细菌替代 BCC 群中最差的 W 个细菌。其中 W 的值越大混 沌搜索算法对本算法的影响就越大,若 W 的值过大,细菌趋药 搜索收敛速度将受到较大影响,探索能力将降低,并且增加了 计算量。为了平衡算法的探索能力和开发能力,一般 W 的值 取 2 即可。

f)满足结束条件结束迭代,否则返回步骤 c)。

4 实验仿真

4.1 测试函数简介

采用文献[1,2,6]中的八个经典标准测试函数对本文提 出的 CBCC 算法和传统 BCC 算法的寻优能力进行对比分析, 函数 $F_1(x,y) \sim F_7(x,y)$ 的定义域都是 $(x,y) \in [-20,20]$,最 优点都是 $F_{optm}(0,0) = 0, F_8(x,y)$ 是一个高维的约束优化函 数,各个分量定义域都是[0,10],最优点为0.803553。

$$(1)F_{1}(x,y) = (x^{2} + y^{2})^{0.25} \times (\sin^{2}(50(x^{2} + y^{2})^{0.1}) + 1.0)$$

$$(2)F_{2}(x,y) = 20 + (x^{2} - 10\cos(2\pi x) + y^{2} - 10\cos(2\pi y))$$

$$(3)F_{3}(x,y) = \frac{(\sin\sqrt{x^{2} + y^{2}})^{2} - 0.5}{(1 + 0.001(x^{2} + y^{2}))^{2}} + 0.5$$

$$(4)F_{4}(x,y) = (x - 1)^{2} + (y - 1)^{2}$$

$$(5)F_{5}(x,y) = (x - 1)^{4} + (y - 1)^{4}$$

$$(6)F_{6}(x,y) = 100(x^{2} - y)^{2} + (1 - x)^{2}$$

$$(7)F_{7}(x,y) = (x - 1)^{6} + (y - 1)^{6}$$

$$(8)F_{8}(x) = \left| \frac{\sum_{i=1}^{n}\cos^{4}(x_{i}) - 2\prod_{i=1}^{n}\cos^{2}(x_{i})}{\sqrt{\sum_{i=1}^{n}ix_{i}^{2}}} \right|,$$

$$\ddot{B}E_{8}(x) = \left| \frac{\sum_{i=1}^{n}\cos^{4}(x_{i}) - 2\prod_{i=1}^{n}\cos^{2}(x_{i})}{\sqrt{\sum_{i=1}^{n}ix_{i}^{2}}} \right|$$

 $\prod_{i=1}^{n} x_i \ge 0.75, \sum_{i=1}^{n} x_i \ge 7.5n, 0 \le x_i \le 10 \text{ for } 1 \le i \le n, n = 40$

4.2 实验参数设置

算法中参数对整个算法的性能有很大影响,本文对文献 [1,2,6]中八个经典标准测试函数进行仿真实验,通过实验数据 分析得到参数设置为 $\varepsilon_{\text{begin}} = 2, \varepsilon_{\text{end}} = 10e - 6, \alpha = 6, e_{\varepsilon} = 10e - 3,$ $\nu = 1, \text{limit} = [-20, 20], N = 20, n_{\varepsilon} = 5, C = 6, W = 2, \text{num} = 300$ 较为合理,算法最大迭代步数为 500 步,迭代结束的精度要求 为 10e - 6。

下面以函数 $F_1(x,y)$ 为例说明混沌群替代 BCC 群的替代 宽度参数 W的设置对算法的影响。图 1 给出了函数 $F_1(x,y)$ 采用本文提出的 CBCC 算法迭代过程中 W=1、W=2、W=3 时 函数适应值的变化情况。从图 1 可以看出,W=1 时算法的收 敛速度明显不如 W=2 与 W=3 时的收敛速度,W=3 时收敛速 度和 W=2 时收敛速度基本相当,但 W=3 时计算量较大,所以 本算法设置参数 W取 2。

4.3 实验结果

用 MATALB 编程实现 BCC 算法和 CBCC 算法,重复实验

20 次,计算各步迭代的平均值。表 1 是 BCC 算法和 CBCC 算 法在求解 *F*₁(*x*,*y*) ~ *F*₈(*x*,*y*)时的迭代次数比较。



算法 $F_1(x,y)$ $F_2(x,y)$ $F_3(x,y)$ $F_4(x,y)$ $F_5(x,y)$ $F_6(x,y)$ $F_7(x,y)$ $F_8(x,y)$ BCC 398 435 342 118 142 216 201 968 CBCC 118 48 60 37 36 165 125 427

从表1可以看出,CBCC 算法由于加入混沌策略,用较少的迭代次数找到较高精度的目标最优值,CBCC 算法比 BCC 算 法收敛速度更快,全局收敛性能更优。

图 2 是求解 $F_1(x,y)$ 用 CBCC 算法和 BCC 算法的寻优过 程中迭代次数和函数目标值对应关系的对比图;图 3 是求解 $F_2(x,y)$ 用 CBCC 算法和 BCC 算法的寻优过程中迭代次数和 函数目标值对应关系的对比图;图 4 是求解 $F_3(x,y)$ 用 CBCC 算法和 BCC 算法的寻优过程中迭代次数和函数目标值对应关 系的对比图。

