

# 基于蚁群算法的月球软着陆轨迹优化

段佳佳<sup>1</sup>, 徐世杰<sup>1</sup>, 朱建丰<sup>2</sup>

(1. 北京航空航天大学宇航学院, 北京 100083; 2. 中国科学院空间科学与应用中心, 北京 100080)

**摘 要:** 针对基于最省燃料的月球软着陆轨迹优化问题进行了研究。首先通过改进的函数逼近法, 将月球软着陆的轨迹优化问题转化为参数优化问题, 并且使优化变量及状态变量均有明确的物理意义。然后利用增加了局部搜索策略的十进制蚁群算法对该优化问题进行研究。仿真算例证明十进制蚁群算法能快速地搜索到满足终端约束条件的最优月球软着陆轨迹, 而且燃料消耗也与采用极大值原理得到的最优燃料消耗相当; 同时与改进的遗传算法- 自适应模拟退火遗传算法相比, 在优化精度相差不多的情况下十进制蚁群算法收敛速度要快很多。仿真结果也说明增加局部搜索策略的十进制蚁群算法具有优良的全局和局部搜索能力。

**关键词:** 轨迹优化; 月球软着陆; 蚁群算法; 局部搜索策略

中图分类号: V448.233

文献标识码: A

文章编号: 1000-1328(2008)02-0476-06

## 0 引言

我国的探月工程已于 2004 年启动, 根据开展月球探测工程的基本原则, 月球探测工程将分为“绕”、“落”、“回”三个阶段实施。在“落”和“回”这两个阶段里, 在月球表面实施软着陆是一项关键技术, 也是进行月球勘探的重要前提。由于月球表面没有大气, 着陆器的速度必须完全由制动发动机抵消, 才能安全实现软着陆, 这一过程需要消耗大量燃料。所以有必要对这一过程进行优化设计。

从目前的文献来看, 求解月球软着陆轨迹优化问题的方法, 可分间接法和直接法两类。间接法的研究比较少, 比如有 WANG Da-yi<sup>[1]</sup> 等根据终端着陆条件和性能指标, 由 Pontryagin 极大值原理得出了从近月点到月球表面的最优着陆轨迹, 并给出一种基于模糊神经网络的非线性最优控制策略。直接法现在研究的比较多, 比如孙军伟<sup>[2]</sup> 等人通过将常推力月球软着陆轨道离散化, 利用离散点处状态连续作为约束条件, 把常推力月球软着陆轨道优化问题归结为一个非线性规划问题, 对于此问题的求解, 其初值均为有物理意义的状态和控制量, 从而避免了采用传统优化方法在解决此优化问题时对没有物理意义变量初值的猜测。王明光<sup>[3]</sup> 等使用伪光谱方法将软着陆轨道优化问题转为一个约束参数优化问题, 然后采用乘子法处理约束条件, 采用变尺度法求解

处理后的参数优化问题, 此方法具有收敛速度快、对初始控制量不敏感、鲁棒性强的优点。朱建丰<sup>[4]</sup> 等自适应遗传算法与模拟退火算法相结合, 得到一种全局搜索能力和局部搜索能力均较强的自适应模拟退火遗传算法, 对月球软着陆轨迹进行优化后, 能够搜索到比较精确的全局最优轨迹。

本文采用一种新的优化算法即蚁群算法 (Ant Colony Algorithm, ACA) 对月球软着陆轨迹优化问题进行研究。通过对基本蚁群算法进行改进得到适合连续参数优化问题的十进制蚁群算法 (Decimalization Ant Colony Algorithm, DACA), 并在算法中增加局部搜索策略来提高搜索精度。然后将其应用到月球软着陆的最优轨迹设计中。

## 1 动力学模型

典型的月球软着陆任务中, 探测器一般首先发射到 100km 的环月停泊轨道, 然后根据所选定的着陆位置, 在合适的时间给着陆器一个有限脉冲, 使得着陆器转入近月点 (在着落位置附近) 为 15km, 远月点为 100km 的月球椭圆轨道, 这一阶段称为霍曼转移段。当着陆器运行到近月点时, 制动发动机开始工作, 其主要任务是抵消着陆器的初始动能和势能, 使着陆器接触地面时, 相对月面速度为零, 即实现所谓的软着陆, 这一阶段称为动力下降段。着陆器的大部分燃料都是消耗在此阶段, 所以月球软着陆轨

迹优化主要是针对动力下降段这一阶段。

由于月球表面附近没有大气, 所以在飞行器的动力学模型中没有大气阻力项。而且从 15km 左右的轨道高度软着陆到月球表面的时间比较短, 一般在几百秒的范围内, 所以诸如月球引力非球项、日月引力摄动等影响因素均可忽略不计, 所以这一过程可以在二体模型下描述。其示意图如图 1 所示, 其中  $o$  为月球质心,  $x$  轴方向为由月心指向着陆器的初始位置,  $y$  轴方向为初始位置着陆器速度方向。

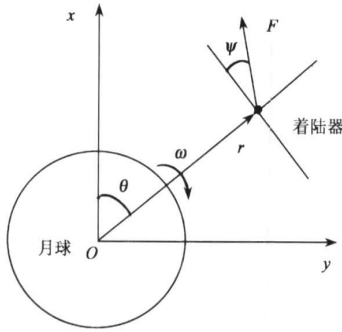


图 1 月球软着陆极坐标系

Fig. 1 The reference frame for lunar soft landing

其动力学方程如下:

$$\dot{r} = v \quad (1a)$$

$$\dot{\theta} = \omega \quad (1b)$$

$$\dot{v} = (F/m) \sin \Psi - \mu/r^2 + r\omega^2 \quad (1c)$$

$$\dot{\omega} = -((F/m) \cos \Psi + 2v\omega)/r \quad (1d)$$

$$\dot{m} = -F/I_{sp} \quad (1e)$$

在上式中  $r$  为着陆器与月心距离,  $v$  为着陆器径向速度,  $\theta$  为着陆器极角,  $\omega$  为着陆器极角角速度,  $\mu$  为月球引力常数,  $F$  为着陆器制动发动机推力,  $m$  为着陆器质量,  $\Psi$  为制动发动机推力方向角, 其定义为  $F$  与当地水平方向夹角,  $I_{sp}$  为制动发动机比冲。

根据动力下降段的起点位置可以确定动力学方程初始条件, 由于起点处于霍曼转移轨道的近地点, 故其初始条件为:

$$r_0 = r_p \quad (2a)$$

$$\theta_0 = 0 \quad (2b)$$

$$v_0 = 0 \quad (2c)$$

$$\omega_0 = \frac{1}{r_p} \sqrt{\frac{\mu}{r_p} \left( \frac{2r_a}{r_a + r_p} \right)} \quad (2d)$$

其中  $r_p$  和  $r_a$  分别为霍曼转移段的近地点半径和远地点半径。

终端条件为实现软着陆, 即

$$r_f = R \quad (3a)$$

$$v_f = 0 \quad (3b)$$

$$\omega_f = 0 \quad (3c)$$

其中  $R$  为月球半径, 终端条件中对终端极角  $\theta$  及终端时间  $t_f$  无约束。

优化变量为制动发动机推力方向角  $\Psi(t)$ 。

优化的性能指标为在满足上述初始条件和终端条件的前提下, 使着陆过程中燃料消耗最少, 即

$$J = \int_{t_0}^{t_f} \dot{m}(t) dt \quad (4)$$

## 2 参数化方法

本文也采用直接法进行轨迹优化, 由于优化变量  $\Psi(t)$  的搜索空间是一个泛函空间, 无法直接应用优化算法, 因此首先要将这个轨迹优化问题转化参数优化问题。现在参数化方法比较多, 有直接离散法, 多段参数插值法和函数逼近法等, 这些方法有各自的优缺点, 本文采用函数逼近法进行参数化。设制动发动机的推力方向角  $\Psi(t)$  可以表示成一个多项式的形式, 即:

$$\Psi(t) = \lambda_0 + \lambda_1 t + \lambda_2 t^2 + \lambda_3 t^3 \quad (5)$$

这样轨迹优化问题就转化为对多项式系数  $\lambda_0$ 、 $\lambda_1$ 、 $\lambda_2$  和  $\lambda_3$  四个参数的优化, 但这四个参数没有明确的物理意义, 确定初值及搜索空间比较困难, 为此本文对函数逼近法作进一步改进。首先将月球软着陆轨迹离散化, 分割成  $n$  个小段, 每段的节点设定一个推力方向角, 那么可以将  $n+1$  个节点的推力方向角和终端时刻  $t_f$  作为待优化的参数。每个节点的時刻可以由下式得到:

$$t_i = t_0 + i(t_f - t_0)/n, \quad (i = 0, 1, \dots, n) \quad (6)$$

这样, 就使得每个节点的推力方向角都有一个对应的节点时刻。那么利用这  $n+1$  个节点的推力方向角及对应时刻对式(5)进行拟合, 可以求得多项式的系数  $\lambda(i = 0, 1, 2, 3)$ , 进而得到整个着陆轨迹的推力方向角曲线  $\Psi(t)$ 。

## 3 蚁群算法

蚁群算法(ACA)是一种新兴的优化算法, 由意大利学者 Dorigo 等人<sup>[5,6]</sup>于 20 世纪 90 年代首次提出, 它是一种基于种群的启发式仿生进化系统。经典的蚁群算法比较适合处理离散的优化问题, 从而限制了其在连续优化问题中的应用。为此本文对经

典蚁群算法进行改进, 得到比较适合连续优化问题的十进制蚁群算法(DACA)<sup>[17]</sup>。

### 3.1 十进制蚁群算法原理

下面以一元连续函数优化为例, 来说明 DACA 算法原理。设一元连续函数优化的数学模型如下:

$$\begin{cases} \min f(x) \\ x \in [x_1, x_2] \end{cases} \quad (7)$$

#### (1) 优化参数区间转换

先通过变换将优化参数  $x$  的变化范围由  $[x_1, x_2]$  转化为  $[0, 1]$  上。设转换后的优化参数  $x$  要求精确到小数点后  $d$  位, 则该优化参数值可以用  $1+d$  个十进制数字来构成。

#### (2) 构造城市群

这里构造  $1+d \times 10$  个城市如图 2 所示, 其结构为  $1+d$  列, 第一列代表个位上的数, 这一列只有一个城市, 代表个位上只能选的数字 0。其余  $d$  列从左至右分别代表小数点后十分位、百分位、千分位等, 每列包括 10 个城市, 分别代表每个数位上可选的 0 到 9 这 10 个十进制数字。

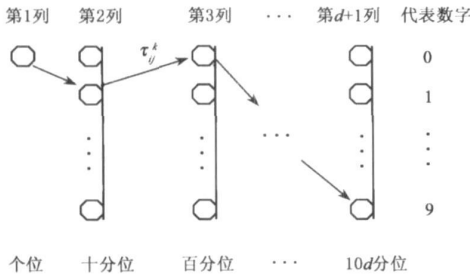


图 2 十进制蚁群算法城市群示意图

Fig. 2 The city group of DACA

#### (3) 搜索路径及可行解

蚂蚁在这样的城市群上选择路径时, 根据一定的路径转移规则从左向右走, 每列上必须且只能选择一个城市。这样蚂蚁从左到右完成一个循环以后, 走过的路径包含有  $1+d$  个城市, 它们所代表的数字可以构成一个在  $[0, 1]$  上有  $d$  位有效数字的小数。经过路径解码后, 即成为优化参数的一个可行解。

#### (4) 更新信息素

所有蚂蚁完成一次循环后, 要根据信息素更新规则更新相关路径上的信息素量。使蚂蚁经过多次循环之后, 逐渐向最优路径聚集, 也即使优化参数逐渐向最优解收敛。

### 3.2 路径转移规则和信息素更新规则

在上述的十进制蚁群算法原理中有两个规则,

即路径转移规则和信息素更新规则, 这是蚁群算法的核心所在<sup>[8]</sup>。下面就本文所使用的这两个规则的内容加以说明。

#### 3.2.1 路径转移规则

记第  $k$  列第  $i$  城市到第  $k+1$  列第  $j$  城市之间的信息素量为  $\tau_{ij}^k$  ( $k=1, 2, \dots, d; i=1, 2, \dots, 10; j=1, 2, \dots, 10$ )。则蚂蚁从在一次循环中从第  $k$  列的第  $i$  个城转移到第  $k+1$  列的第  $j$  个城市的概率为

$$P_{ij}^k = \frac{\tau_{ij}^k}{\sum_{r \in \{1, 2, \dots, 10\}} \tau_{ir}^k} \quad (8)$$

从式(8)中可以看出, 从第  $k$  列的第  $i$  个城市到第  $k+1$  列的 10 个城市中, 路径上信息素量越大, 这个城市被选择的概率就越大。这种基于概率选择的机制, 使蚂蚁能以较大的可能性选择较好的路径, 从而使较多的蚂蚁聚集在好的路径上。同时也没有绝对排除其它路径, 为跳出局部最优解提供了可能性。

#### 3.2.2 信息素更新规则

经过一个循环以后, 要更新路径上的信息素量, 以形成正反馈使蚂蚁向最优路径靠近。信息素更新公式如下:

$$\tau_{ij}^k(N+1) = (1-\rho) \tau_{ij}^k(N) + \Delta \tau_{ij}^k \quad (9)$$

上式中  $N$  表示循环次数,  $\rho$  为挥发系数, 其大小决定了经过一个周期所有路径上信息素的挥发强度, 其值越大, 挥发比例越大。信息素的挥发可以使相对不好的路径上的信息素不断减少, 加快向最优路径上收敛的速度。 $\Delta \tau_{ij}^k$  为信息素加强量, 在本文中采用精英策略进行信息素加强, 即当所有蚂蚁都完成了一次循环后, 找出前十只对应目标函数值最小的蚂蚁, 对这前十只蚂蚁经过的路径上的信息素按下式作全局更新:

$$\Delta \tau_{ij}^k = KQ^{(f(x_k) - best)}, \quad (k=1, 2, \dots, 10) \quad (10)$$

其中  $K$  为权重系数, 可以调整  $\Delta \tau_{ij}^k$  的权重。 $Q$  为小于 1 的正数,  $f(x_k)$  为前十只蚂蚁中第  $k$  只对应的目标函数值,  $best$  为已经历的所有循环中最优蚂蚁对应的目标函数值。这样每次循环后前十个最好的路径上的信息素会根据其对应函数值的大小得到不同程度的加强, 其中目标函数值越小加强量越多。根据路径转移规则, 在下次循环中, 会有更多的蚂蚁选择这些较好路径。

### 3.3 局部搜索策略

为了增强蚁群算法的局部搜索能力, 提高最终结果的精度, 本文在上述规则的基础上, 增加了局部

搜索策略如图 3 所示。设每次循环中的最优路径上根据 3.2.2 节的信息素更新规则需要增强的信息素量为  $\Delta\tau_{ij}$ , 那么规定对其相邻的路径也进行加强, 加强值为  $\Delta\tau_{ij}/4$ , 这样以后的循环中, 该次循环最优路径附近中更好的路径将有比较大的机会被搜索到, 从而可以提高蚁群算法的局部搜索能力。

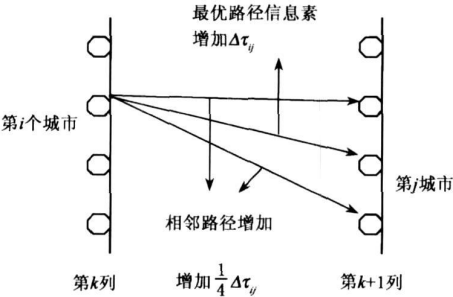


图 3 局部搜索策略示意图  
Fig. 3 Sketch map for local search

3.4 用于多元连续函数的十进制蚁群算法

用于多元连续函数的 DACA 算法与一元函数类似。设有  $m$  个优化参数, 并要求第  $k$  个优化参数精确到小数点后  $d_k$  位, 那么可按照一元函数的情况从左到右依次构建  $\sum_{k=1}^m (1 + d_k)$  列城市, 其余路径转移规则、信息素更新规则和局部搜索策略都与一元连续函数情况相同。

3.5 算法的实现

- 下面给出 DACA 算法应用过程的具体步骤。
- (1) 设置初始参数, 包括蚂蚁数  $num\_ant$ , 循环次数  $num\_clc$ , 挥发系数  $\rho$ , 调节系数  $K、Q$ , 所有路径信息素量初值  $\tau_0$ , 蚂蚁初始位置等。
  - (2) 根据式 (2) 计算每只蚂蚁的转移概率, 然后依据赌轮原则为每只蚂蚁选择下一个路径城市。重复上述操作直至所有蚂蚁均完成一次循环。
  - (3) 将每只蚂蚁的路径解码为优化参数值, 计算目标函数值, 找出最好的前 10 只蚂蚁的路径。然后根据式 (9)、(10) 和局部搜索策略规则更新相关路径上的信息素。
  - (4) 判断是否满足终止条件, 不满足则重复 (1) 至 (3) 步过程; 满足则结束计算输出结果。

4 仿真实例

月球软着陆的初始条件如表 1 所示:

表 1 月球软着陆的初始条件相关参数

Table 1 Initial parameters of lunar soft landing

$r_p / \text{km}$	$r_d / \text{km}$	$R / \text{km}$	$m_0 / \text{kg}$	$F / \text{N}$	$I_{sp} / (\text{m/s})$
1753	1838	1738	600	1350	2940

用函数逼近法进行参数化的相关参数设置为: 设将轨迹离散化化为 9 段, 那么待优化参数共 11 个参数, 即 10 个推力方向角和 1 个终端时刻  $t_f$ 。在用蚁群算法进行优化过程中需要确定这 11 个优化参数的搜索范围。对于 10 个方向角, 由经验可知, 推力方向与着陆器速度反方向夹角不会超过 90 度, 否则着陆器将会被加速, 燃料消耗将更多。所以确定 10 个推力方向角的变化范围为:

$$0^\circ < \psi_i < 90^\circ, i = 1, 2, \dots, 10 \quad (11)$$

对于终端时刻  $t_f$ , 根据齐奥尔科夫斯基公式和软着陆初始条件, 可由下式估计:

$$t_f = (1 - \exp((V_f - V_0)/I_{sp})) (I_{sp} m_0 / F) \quad (12)$$

式中  $V_f$  和  $V_0$  分别表示着陆器的终端速度和初始速度, 经计算确定  $t_f$  搜索范围为(单位秒):

$$500 < t_f < 700 \quad (13)$$

十进制蚁群算法中的相关参数设置如表 2 所示:

表 2 DACA 算法相关参数

Table 2 Initial parameters for DACA

$num\_ant$	$num\_clc$	$K$	$Q$	$\rho$	$\tau_0$
1000	50	5	0.5	0.2	5

按照前述方法用 DACA 对月球软着陆轨迹进行优化。由于 DACA 是基于概率转移的仿生算法, 优化结果具有一定的随机性, 所以这里进行了 10 次仿真, 结果全部收敛, 证明 DACA 具有良好的收敛性。对仿真结果取平均值并与由 Pontryagin 极大值原理得到的结果比较如表 3 所示:

表 3 DACA 与 Pontryagin 结果比较

Table 3 The results of DACA compared with Pontryagin's maximum principle

	DACA 结果	Pontryagin 结果	约束值
最优喷气时间长度/s	604.75	604.5	无约束
最优燃料消耗/kg	277.69	277.58	无约束
着陆径向速度/(m/s)	-0.0010135	0.00132226	0
着陆月心距/m	1738000.008	1737999.999	1738000
着陆极角角速度/(rad/s)	5.118e-010	4.382e-008	0

可以看出 DACA 得到的最优轨迹的着陆状态值非常接近终端约束值, 说明着陆器很好的实现了软

着陆。同时与利用 Pontryagin 极大值原理得到的最优结果相比,两者在着陆精度上相差不多,在着陆径向速度和着陆极角角速度上 DACA 还要优于 Pontryagin 极大值原理的结果。从燃料消耗来看, DACA 得到 10 次结果的均值仅比 Pontryagin 极大值原理结果多 0.11kg, 其中最优的一次仅多 0.0669kg, 可见 DACA 的优化精度很高的。但从算法的适应性考虑, DACA 与 Pontryagin 极大值原理相比要有很大的优势, 因为当软着陆轨迹的初始条件或终端条件改变后, 再应用极大值原理进行轨迹优化时, 必须重新对初值进行猜测, 而其猜测过程极其繁琐; 而 DACA 则只要对算法的参数进行适当的调整就能求得新的最优轨迹。

大值原理最优结果的推力方向角曲线(图 4a)及状态曲线(图 4b, 4c, 4d)比较图。可以看出 DACA 得到的结果与利用 Pontryagin 极大值原理得到的结果有所区别, 这是因为本文的推力方向角曲线是用 3 阶多项式拟和的, 它必然不会与 Pontryagin 极大值原理得到最优推力方向角曲线完全重合, 从而也导致了其它三个状态变化曲线的不一致。从理论上说提高多项式的阶数可以使推力曲线和状态曲线进一步逼近 Pontryagin 极大值原理的结果。但是从燃料消耗及着陆效果的角度来说, 用 3 阶多项式拟和时的燃料消耗已经非常接近最优值, 而且得到的推力方向角曲线也同样实现了高精度的软着陆, 所以用 3 阶多项式已经足够, 没有必要再提高多项式的阶数。

图 4 为 DACA 最好的一次结果和 Pontryagin 极

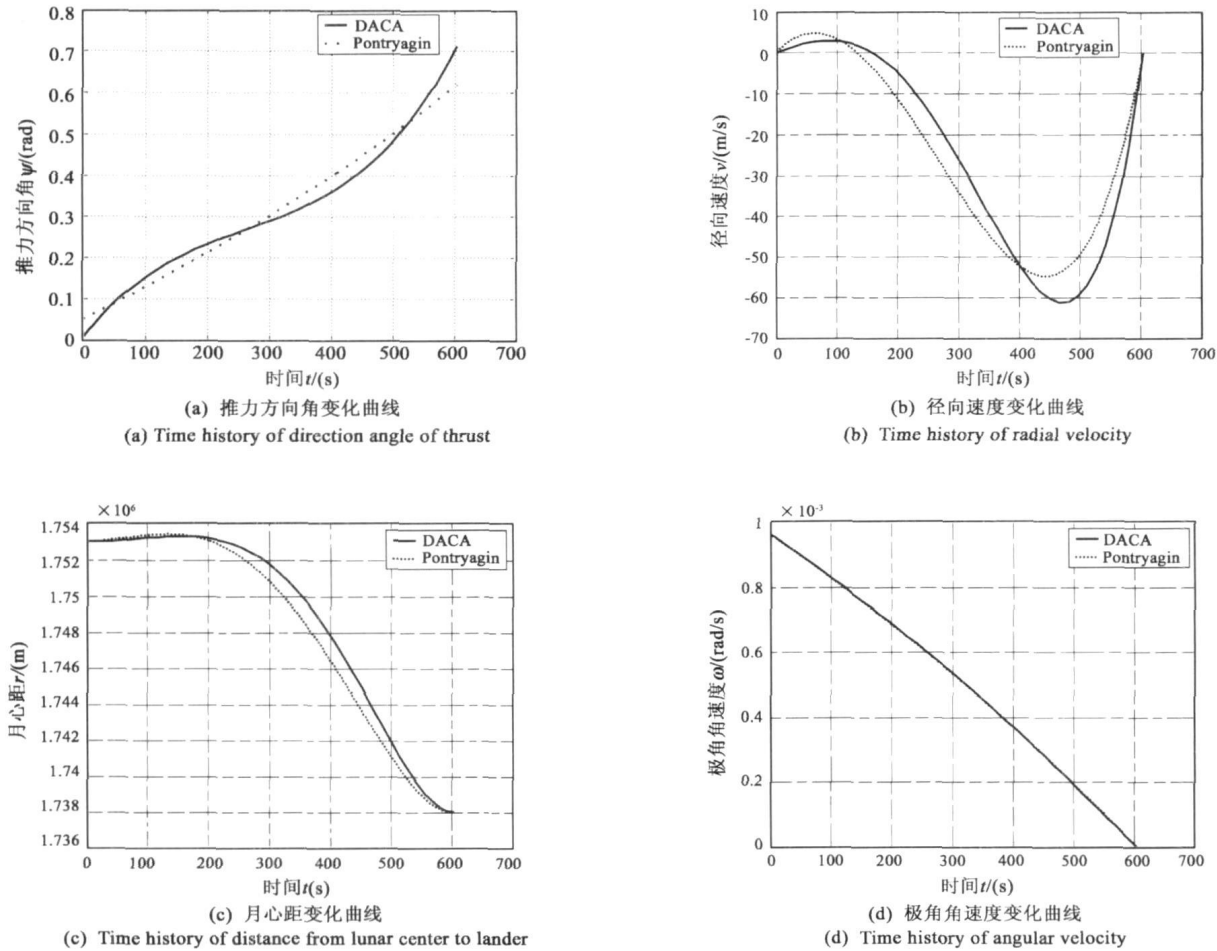


图 4 推力方向及状态变化曲线

Fig. 4 Time history of state parameters

表 4 为 DACA 与自适应模拟退火遗传算法 (Adaptive Simulated Annealing Genetic Algorithm, ASAGA) 的比较结果。ASAGA 是将自适应遗传算法与模

拟退火算法相结合而得到一种混合算法, 该算法具备了自适应遗传算法的强大全局搜索能力和模拟退火算法的强大局部搜索能力, 已被很多算例证明具

有优良的优化性能。表 4 中数据为两种算法都运算 10 次后得到的结果的平均值。

表 4 DACA 与 ASAGA 结果比较

Table 4 The results of DACA compared with ASAGA

	DACA 结果	ASAGA 结果	约束值
平均喷气时间长度/s	604.75	604.72	无约束
平均燃料消耗/kg	277.69	277.68	无约束
平均着陆径向速度/ (m/s)	- 0.001013	0.006599	0
平均着陆时月心距/m	1738000.008	1737999.919	1738000
平均着陆极角角速度/ (rad/s)	5.118e- 010	1.660e- 008	0
平均优化时间/min	43	216	无约束

可以看出, 两者优化的结果非常接近, 着陆精度 DACA 略好, 燃料消耗 ASAGA 略好。但是从优化时间上, 我们可以看出 DACA 要比 ASAGA 好得多。原因是 ASAGA 每次循环过程中有交叉, 变异及模拟退火局部搜索等多种算子操作, 需要多次计算种群适应度, 比较消耗时间。而在 DACA 中比较消耗时间的目标函数计算每次循环只有一次。故 DACA 在收敛速度上要比 ASAGA 快很多。

5 结论

本文针对基于最省燃料的月球软着陆轨迹优化问题进行了研究, 首先通过改进的函数逼近法, 将月球软着陆的轨迹优化问题转化为参数优化问题。然后对基本蚁群算法进行改进, 得到了适合连续参数优化十进制蚁群算法, 并在算法中增加了局部搜索策略来提高搜索精度, 然后将其应用到月球软着陆的最优轨迹设计中。仿真结果证明十进制蚁群算法能快速地搜索到满足终端约束条件的最优月球软着陆轨迹, 且燃料消耗与采用极大值原理得到的最优燃料消耗相当。同时与自适应模拟退火遗传算法相比, 在优化精度相差不多的情况下前者的收敛速度要快很多。结果同时说明, 在参数化时虽然推力方向角曲线只用三阶的多项式曲线来近似, 但这并不影响软着陆的精度, 也不会明显增加燃料消耗量。本文得到的增加局部搜索策略的十进制蚁群算法具有非常良好的全局和局部搜索能力, 也可以应用到其它的优化问题中。

参考文献:

[1] WANG Da-yi, QIAO Guo-dong, LI Tie-shou. FNN guidance control for lunar optimal trajectory tracking descent[J]. Journal of Astronautics, 2007, 28(5): 1149- 1155.

[2] 孙军伟, 乔栋, 崔平远. 基于 SQP 方法的定常推力月球软着陆轨道优化方法[J]. 宇航学报, 2006, 27(1): 99- 103. [SUN Jun-wei, QIAO Dong, CUI Ping-yuan. Study on the optimal trajectories of lunar soft-landing with fixed thrust using SQP method[J]. Journal of Astronautics, 2006, 27(1): 99- 103(in Chinese).]

[3] 王明光, 裴听国, 袁建平. 基于伪光谱方法月球软着陆轨道快速优化设计[J]. 中国空间科学技术, 2007, 27(5): 27- 32. [WANG Ming-guang, PEI Ting-guo, YUAN Jian-ping. Legendre pseudospectral method for rapid lunar soft-landing trajectory optimization[J]. Chinese Space Science and Technology, 2007, 27(5): 27- 32(in Chinese).]

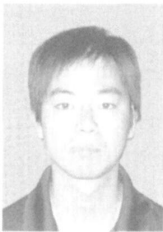
[4] 朱建丰, 徐世杰. 基于自适应模拟退火遗传算法的月球软着陆轨迹优化[J]. 航空学报, 2007, 28(4): 806- 812 [ZHU Jian-feng, XU Shi-jie. Optimization of lunar soft landing trajectory based on adaptive simulated annealing genetic algorithm[J]. Chinese Journal of Aeronautics, 2007, 28(4): 806- 812(in Chinese).]

[5] Colomi A, Dorigo M, Maniezzo V, et al. Distributed optimization by ant colonies[C]. Proceedings of the 1st European Conference on Artificial Life, 1991: 131- 142.

[6] Dorigo M, Maiezzo V, Colomia A. Ant system: optimization by a colony of cooperating agents[J]. IEEE Transaction on Systems, Man, and Cybernetics- Part B, 1996, 26(1): 29- 41.

[7] 陈烨. 用于连续函数优化的蚁群算法[J]. 四川大学学报(工程科学版), 2004, 36(6): 117- 120. [CHEN Ye. Ant colony system for continuous function optimization[J]. Journal of Sichuan University (Engineering Science Edition), 2004, 36(6): 117- 120(in Chinese).]

[8] Dorigo M, Gambardella L M. Ant colony system: A cooperative learning approach to the traveling salesman problem[J]. IEEE Transactions on Evolutionary Computation, 1997, 1(1): 53- 66.



作者简介: 段佳佳(1983- ), 男, 硕士生, 研究方向为航天器轨道设计、优化及控制。  
通信地址: 北京航空航天大学宇航学院 504 教研室(100083)  
电话: (010) 82339751  
E-mail: duanjiajia@sa.buaa.edu.cn

(下转第 488 页)

## Orbit Design on Cislunar Transfer Trajectory under Multi Constraints

LIU Lei, XI Xiao-ning, BAI Yi-zhu

(College of Aerospace and Material Engineering, National University of Defense Technology, Changsha 410073, China)

**Abstract:** Orbit design on cislunar transfer trajectory under multi constraints is discussed in this article. The constraints include conditions of TT&C on the ground, solar illumination, flight time and position of orbit entrance, etc. By the search arithmetic of cislunar transfer trajectory, the article puts forward a search strategy combining resolution and numerical methods and using a step-by-step technique. The strategy introduces the above constraints in search processing and works preferably to get a result meeting expectations on the precise and the constraint conditions. The case in the end and lots of numerical simulations show that the strategy is simple, effective and swift to receive a satisfying result.

**Key words:** Cislunar transfer trajectory; Multi constraints; Orbit design; Differential correction; Search strategy

---

(上接第 481 页)

## Optimization of Lunar Soft Landing Trajectory Based on Ant Colony Algorithm

DUAN Jia-jia<sup>1</sup>, XU Shi-jie<sup>1</sup>, ZHU Jian-feng<sup>2</sup>

(1. School of Astronautics, Beijing University of Aeronautics and Astronautics, Beijing 100083, China;

2. Center for Space Science and Applied Research, China Academy of Sciences, Beijing 100080, China)

**Abstract:** The optimization of lunar soft landing trajectory was studied in this paper. Firstly a new parameterized method was developed to convert the trajectory optimization problem into a parameter optimization problem, in which all optimization parameters and motion states had specific physical meanings. Then a new decimalization ant colony algorithm (DACA) with a local search strategy was proposed and applied to solve this optimization problem. Simulation results show that DACA is efficient for finding the global optimum trajectory, which has a fuel consumption quite close to the result of Pontryagin's maximum principle. It is also proved DACA has a strong ability in both global search and local search.

**Key words:** Trajectory optimization; Lunar soft landing; Ant colony algorithm; Local search strategy