

求解航天器最优交会问题的改进和声搜索算法

王 皓, 欧阳海滨, 高立群

(东北大学 信息科学与工程学院, 辽宁 沈阳 110819)

摘 要: 针对航天器最优交会问题, 基于 C-W 模型建立一种燃料时间混合指标, 并提出一种改进和声搜索 (AHS) 算法进行求解. 在 AHS 算法中, 提出一种全局均匀学习操作, 利用了当前全局最优和声的指导作用, 取代了原始和声搜索算法的基音调整操作, 增强全局搜索和局部搜索的平衡, 并对参数 PAR 进行了有效的动态调整, 以更好适应算法的搜索进程. 利用几个最优交会实例对 AHS 算法的有效性进行了测试, 数值结果表明 AHS 算法能够取得满意的结果, 并且优于其他算法.

关 键 词: 最优交会问题; 改进和声搜索算法; 全局均匀学习; 全局搜索; 局部搜索

中图分类号: TP 301.6 **文献标志码:** A **文章编号:** 1005-3026(2015)07-0913-05

Amended Harmony Search Algorithm for Solving Spacecraft Optimal Rendezvous Problem

WANG Hao, OUYANG Hai-bin, GAO Li-qun

(School of Information Science & Engineering, Northeastern University, Shenyang 110819, China. Corresponding author: OUYANG Hai-bin, E-mail: oyhb1987@163.com)

Abstract: A hybrid index of fuel-time on the basis of C-W equations was built for the spacecraft optimal rendezvous problem, and an amended harmony search (AHS) algorithm was proposed to solve this problem. In the AHS algorithm, a global uniform learning operation was presented that the guidance of the current global best harmony was utilized and the pitch adjusting operation was replaced, resulting in the enhancement of the balance between the global search and local search. The PAR was dynamically adjusted to adapt the search process of algorithm. Several optimal rendezvous cases were used to test the effectiveness of AHS algorithm, and it was verified by the numerical results that correct satisfied results could be obtained with the proposed AHS algorithm, which is better than that of the other algorithms.

Key words: optimal rendezvous problem; amended harmony search algorithm; global uniform learning; global search; local search

空间交会对接 (rendezvous and docking) 是进行空间组装、空间救援、深空探测的必要技术^[1]. 空间交会过程主要有远距离导引段、近距离导引段、最后接近段以及逼近对接段^[2]. 其中在近距离导引变轨, 实现空间最优交会对接对未来空间探索具有十分重要的作用.

在过去的几十年中, 关于航天器最优交会问题的研究, 已有很多学者做出了许多贡献. 在这些研究中, 许多常规的寻优方法应用于最优交会问题, 例如解析法、动态规划方法等. 但是传统的常

规寻优方法求解速度慢, 优化效果并不令人满意, 要求的数学条件苛刻, 而智能优化算法在最近几十年得到了迅速发展, 具有寻优效率高、操作简单、数学条件要求低等优势, 在许多工程领域都得到了广泛应用. 在航空航天领域, 许多智能优化算法也逐渐得到应用. 例如: 模拟生物进化理论“适者生存, 优胜劣汰”准则的遗传算法以及混合遗传算法^[3], 基于群体差异的差分演化算法^[4], 基于统计学理论的分布估计算法^[2]. 通过大量的数值仿真表明智能优化算法相对传统的常规方法搜

收稿日期: 2014-05-28

基金项目: 国家自然科学基金资助项目 (61273155); 辽宁省科技厅博士科研启动基金资助项目 (L201205118).

作者简介: 王 皓 (1981-), 女, 辽宁沈阳人, 东北大学博士研究生; 高立群 (1949-), 男, 辽宁沈阳人, 东北大学教授, 博士生导师.

索效率更高,并且能够较好地搜索到比较满意的结果.但是智能优化算法存在易陷入局部最优的不足,最终搜索的结果不稳定.

和声搜索(harmony search, HS)算法是由 Geem 等于 2001 年提出的一种新颖的智能算法^[5]. HS 算法已被成功应用到许多实际优化问题中.本文针对航天器最优交会问题,提出一种改进的和声搜索算法进行求解.通过几个实例表明了本文算法的有效性,并与 HS 算法、粒子群(PSO)^[6]算法、差分演化(DE)^[7]算法及人工蜂群(ABC)^[8]算法进行了比较,结果显示了本文算法的优越性.

1 最优交会模型

航天器交会实质上是追踪航天器按照一定的规律进入到指定的目标航天器轨道.文献[2]给出了简单的目标轨道坐标系如图 1 所示.

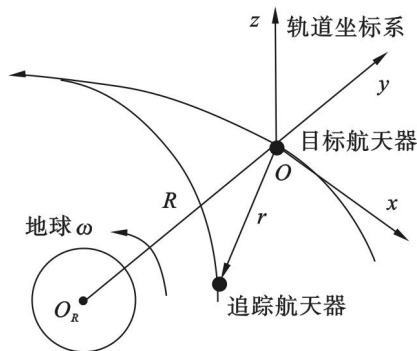


图 1 目标轨道坐标系
Fig. 1 Target orbit coordinate system

图 1 中,设目标航天器的质心为 O , Ox 轴为目标航天器所在轨道速度切线的反方向,背向地心的方向为 Oy 轴,与 Ox 轴, Oy 轴构成右手直角坐标系为 Oz 轴.为了方便相对轨道位置运动的研究,认为目标轨道坐标系随着目标航天器的轨道运行.依据此坐标系建立的 C-W 线性模型如下:

$$\left. \begin{aligned} \ddot{x} &= 2\omega\dot{z} + B_x, \\ \ddot{y} &= -\omega^2 y + B_y, \\ \ddot{z} &= -2\omega\dot{x} + 3\omega^2 z + B_z. \end{aligned} \right\} \quad (1)$$

其中: B_x, B_y, B_z 分别表示施加于追踪航天器上 3 个方向的连续加速度; ω 为目标航天器的轨道角速度.当采用脉冲推力假设时,只需设置飞行状态的初始值,并将 B_x, B_y, B_z 同时置为零.设状态变量 $\mathbf{x} = [x, y, z, \dot{x}, \dot{y}, \dot{z}]^T$, 控制变量 $\mathbf{u} = [B_x, B_y, B_z]^T$, 则式(1)可写成如下状态方程:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}. \quad (2)$$

其中:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 2\omega & 0 \\ 0 & 3\omega^2 & 0 & -2\omega & 0 & 0 \\ 0 & 0 & -\omega^2 & 0 & 0 & 0 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

可得到状态转移矩阵为

$$\boldsymbol{\varphi}(t-t_0) = \begin{bmatrix} \varphi_{11} & \varphi_{12} \\ \varphi_{21} & \varphi_{22} \end{bmatrix}. \quad (3)$$

其中:

$$\begin{aligned} \varphi_{11} &= \begin{bmatrix} 1 & 6(\tau - \sin\tau) & 0 \\ 0 & 4 - 3\cos\tau & 0 \\ 0 & 0 & \cos\tau \end{bmatrix}, \\ \varphi_{12} &= \begin{bmatrix} (4\sin\tau - 3\tau)/\omega & 2(1 - \cos\tau)/\omega & 0 \\ 2(1 - \cos\tau)/\omega & \sin\tau/\omega & 0 \\ 0 & 0 & \sin\tau/\omega \end{bmatrix}, \\ \varphi_{21} &= \begin{bmatrix} 1 & 6\omega(1 - \cos\tau) & 0 \\ 0 & 3\sin\tau & 0 \\ 0 & 0 & -3\sin\tau \end{bmatrix}, \\ \varphi_{22} &= \begin{bmatrix} 4\cos\tau - 3 & 2\sin\tau & 0 \\ -2\sin\tau & \cos\tau & 0 \\ 0 & 0 & \cos\tau \end{bmatrix}. \end{aligned}$$

其中, $\tau = \omega(t - t_0)$.

设在 t_0 时刻,航天器的初始状态设为 $(x_0, y_0, z_0, v_{x0}, v_{y0}, v_{z0})$. 运行一段时间后,加入第一次脉冲,假设此时为 t_1 时刻, t_1 时刻前航天器的状态为 $(x_{10}, y_{10}, z_{10}, v_{x10}, v_{y10}, v_{z10})$, 由状态转移矩阵式(3)可以得到 $(x_{10}, y_{10}, z_{10}, v_{x10}, v_{y10}, v_{z10}) = \boldsymbol{\varphi}(t - t_0) \cdot (x_0, y_0, z_0, v_{x0}, v_{y0}, v_{z0})$. 如果 t_2 时刻航天器加入第二次脉冲时实现交会,并且末端状态为 $(0, 0, 0, 0, 0, 0)$, 则利用状态方程和状态转移矩阵可推导出 t_1 时刻后的速度 $(v_{x12}, v_{y12}, v_{z12})$ 和 t_2 时刻之前航天器的速度 (v_{x2}, v_{y2}, v_{z2}) , 则有第一次脉冲速度为

$$\Delta v_1 = \sqrt{(v_{x12} - v_{x1})^2 + (v_{y12} - v_{y1})^2 + (v_{z12} - v_{z1})^2}. \quad (4)$$

第二次脉冲速度为

$$\Delta v_2 = \sqrt{v_{x2}^2 + v_{y2}^2 + v_{z2}^2}. \quad (5)$$

航天器交会优化目的是实现时间和燃料消耗的最小化,文献[6-11]采用加权求和的方法建立如下指标:

$$J = |\Delta v_1| + |\Delta v_2| + \lambda(T_1 + T_2). \quad (6)$$

其中, $T_1 = t_1 - t_0$, $T_2 = t_2 - t_1$; λ 为可调的权系数;

$|\cdot|$ 表示向量取模. 因为燃料消耗和时间是两个不同物理量, 不能单纯叠加, 式(6)有些不足, 最好是分别归一化之后再相加. 因此, 在本文中能量-时间混合优化指标设计如下:

$$J = \frac{|\Delta v_1| + |\Delta v_2|}{\max(|\Delta v_1| + |\Delta v_2|)} + \lambda \left(\frac{T_1 + T_2}{\max(T_1 + T_2)} \right). \quad (7)$$

其中, $\max(x)$ 表示取 x 的最大值. 如果 T_1 和 T_2 给定, 则 Δv_1 和 Δv_2 也可以根据式(2)~式(5)求出. 所以实质上只需要优化变量 T_1 和 T_2 .

2 AHS 算法设计

本文结合航天最优交会问题, 提出一种改进和声搜索算法. 下面具体介绍全局均匀调整方法和动态参数设计.

2.1 全局均匀调整

在和声搜索算法中, 即兴创作过程是和声搜索算法最关键的部分, 主要包括3个操作: 和声记忆考虑、基音调整和随机选择. 为了更好地增强和声搜索算法全局搜索和局部搜索的平衡, 在改进和声搜索算法中, 设计了一种全局均匀调整方法. 其具体的表达式如下:

$$x_j = \begin{cases} x_j + \text{rand}_1 \times (x_{\text{best},j} - c_1 \times \bar{x}_j) + \\ \text{rand}_2 \times (x_{r_1,j} - x_{r_2,j}), f(x_{r_1}) \leq f(x_{r_2}); \\ x_j + \text{rand}_1 \times (x_{\text{best},j} - c_1 \times \bar{x}_j) + \\ \text{rand}_2 \times (x_{r_2,j} - x_{r_1,j}), f(x_{r_1}) > f(x_{r_2}). \end{cases} \quad (8)$$

其中, $x_{\text{best},j}$ 表示当前最好和声的第 j 维变量; \bar{x}_j 表示所有和声第 j 维变量的平均值; 第 $x_{r_1,j}$ 和 $x_{r_2,j}$ 表示随机选的两个和声的第 j 维变量; rand_1 和 rand_2 表示 0 和 1 之间的随机数; c_1 为全局调整因子, c_1 为 1 或 2, $c_1 = \text{round}(1 + \text{rand})$. 通过当前全局最好的和声与整个和声记忆库中所有和声的平均值的比较, 促使新产生的和声向当前最优方向靠近, 增强算法收敛性能. 另一方面利用和声记忆库中随机的较好和声拓展搜索的广泛性, 防止算法陷入局部极值.

2.2 参数动态调整

在和声算法中, 基音调整概率 PAR 决定了全局均匀调整的执行概率. 为了适应算法的搜索进程, 文献[9-12]中提出了各种 PAR 动态调整的方法:

$$\text{PAR}_k = \text{PAR}_{\min} + \frac{(\text{PAR}_{\max} - \text{PAR}_{\min})}{K} \times k. \quad (9)$$

$$\text{PAR}_k = \text{PAR}_{\min} + \frac{(\text{PAR}_{\max} - \text{PAR}_{\min}) \times (k-1)}{K-1}. \quad (10)$$

$$\text{PAR}_k = \text{PAR}_{\max} - \frac{(\text{PAR}_{\max} - \text{PAR}_{\min})}{K} \times k. \quad (11)$$

$$\text{PAR}_k = \text{PAR}_{\min} + (\text{PAR}_{\max} - \text{PAR}_{\min}) \cdot \exp(-k/K). \quad (12)$$

$$\begin{aligned} \text{PAR}_k &= m \cdot \exp(n \times k^2), \\ n &= \frac{1}{K^2 - 1} \ln\left(\frac{\text{PAR}_{\max}}{\text{PAR}_{\min}}\right), \\ m &= \text{PAR}_{\min} \exp(-n). \end{aligned} \quad (13)$$

$$\text{PAR}_k = \text{PAR}_{\min} \exp\left(\frac{k}{K} \ln\left(\frac{\text{PAR}_{\max}}{\text{PAR}_{\min}}\right)\right). \quad (14)$$

其中: k 和 K 分别代表当前迭代次数和最大迭代次数; PAR_{\min} 和 PAR_{\max} 分别代表最小基音调整概率和最大基音调整概率. 在式(9)~式(11)中 PAR 随着迭代次数呈线性变化, PAR 在整个搜索过程中, 变化的速率是恒定的, 这种调整简单易行, 但比较粗糙. 式(12)~式(14)中 PAR 随着迭代次数呈指数变化. PAR 在整个搜索过程中, 变化的速率是变化的; 如果 PAR 在迭代早期变化比较缓慢, 更能挖掘局部信息, 而随着迭代次数的增加 PAR 的变化逐渐加快, 这样能加快搜索速度. 因此, 本文将参数 PAR 设计为

$$\text{PAR}_k = 1 - \sqrt{1 - k/K}. \quad (15)$$

假设 $K = 1000$, 参数 PAR 随着迭代次数 k 的变化曲线如图2所示.

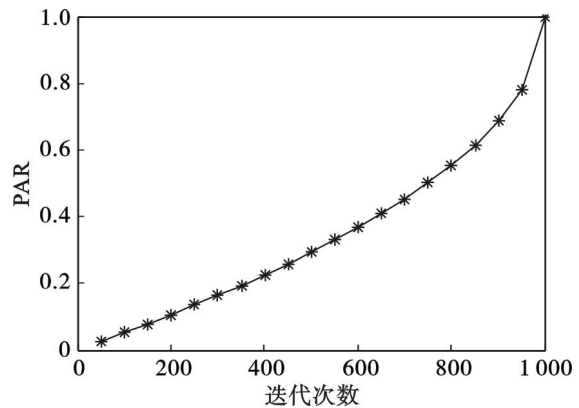


图2 PAR 变化曲线
Fig. 2 Changing curve of PAR

2.3 AHS 算法流程

步骤1 优化问题和算法参数初始化.

航天器最优交会问题的参数包括问题的维数 $D = 2$, 变量的取值范围 $x_{j,L} \leq x_j \leq x_{j,U}, j = 1, 2$; 和声搜索算法参数主要有最大迭代次数 K , 和声记忆库大小 HMS (harmony memory size), 和声记忆考虑概率 HMCR (harmony memory considering rate), 基音调整概率 PAR (pitch adjusting rate) 及调整步长 bw.

步骤2 和声记忆库进行初始化.

和声记忆库用来存储 HMS 个随机的和声解向量. 和声解向量的每一维和声分量由变量的上下界 $x_{j,U}$ 和 $x_{j,L}$ 产生.

步骤 3 进行即兴创作, 产生一个新的和声解向量. 其具体的即兴创作伪代码如下:

For $j = 1$ to D do

If $\text{rand}() \leq \text{HMCR}$ 和声记忆库考虑

$x_{\text{new},j} = x_{r,j}, r \in (1, 2, 3, \dots, \text{HMS})$

If $\text{rand}() \leq \text{PAR}$ then 全局均匀调整

If $f(x_{r_1}) < f(x_{r_2})$

$x_{\text{new},j} = x_{\text{new},j} + \text{rand}_1 \cdot (x_{\text{best},j} - c_1 \cdot x_{\text{new},j}) + \text{rand}_2 \cdot (x_{r_1,j} - x_{r_2,j})$

Else

$x_{\text{new},j} = x_{\text{new},j} + \text{rand}_1 \cdot (x_{\text{best},j} - c_1 \cdot x_{\text{new},j}) + \text{rand}_2 \cdot (x_{r_2,j} - x_{r_1,j})$

End If

End If

Else

$x_{\text{new},j} = x_{j,L} + \text{rand}() \times (x_{j,U} - x_{j,L})$ 随机选择

End If

End For

步骤 4 进行选择操作, 更新和声记忆库.

如果新的和声解向量对应的适应值优于最差和声解向量对应的适应值, 则新的和声解向量取代最差和声解向量, 否则最差和声解向量保持不变.

步骤 5 检查停止准则.

如果当前迭代次数等于最大迭代次数 K , 则

停止运行 HS 算法, 否则重复运行步骤 3 和步骤 4, 直到迭代次数达到最大迭代次数为止.

3 仿真实验与结果分析

假设目标航天器在 400 km 高的圆轨道上运行, 追踪航天器取两个初始相对状态计算, 分别为 (70 000, -30 000, 0, -40, 30, 0) 和 (100 000, -20 000, 0, -20, 0, 0), 交会终端相对状态为 (0, 0, 0, 0, 0, 0).

本文算法和 PSO 算法^[6]、DE 算法^[7]、HS 算法^[5]及 ABC 算法^[8]对此问题进行求解, 各个算法的参数设置如表 1 所示, 每个算法的目标函数评价次数都设为 50 000. 实验仿真环境为 Windows XP 系统, Intel(R) Pentium(R) 4 CPU 2.93 GHz, 2.94 GHz, 4 GB 内存; 仿真软件 MATLAB 7.04, 独立运行 30 次, 所求得的结果如表 2 所示.

表 1 算法参数

Table 1 Algorithm parameters

算法	参数
HS	HMCR = 0.9; HMS = 5; PAR = 0.33; bw = 0.000 1
PSO	$c_1 = 2$; $c_2 = 2$; population size = 40
AHS	HMCR = 0.9; HMS = 40
DE	$F = 0.5$, CR = 0.9; population size = 40
ABC	population size = 40; limit = 500

表 2 5 种算法的优化结果 ($\lambda = 0.1$)

Table 2 The optimization results of 5 algorithms ($\lambda = 0.1$)

初始位置/m	初始速度/($\text{m} \cdot \text{s}^{-1}$)	算法	最好值	最差值	中间值	平均值	方差
$x_0 = 70\ 000$	$v_{x0} = -40$	PSO	$4.375\ 9\text{e}-1$	$1.092\ 3\text{e}+1$	$8.067\ 6\text{e}-1$	1.974 7	3.207 1
$y_0 = -30\ 000$	$v_{y0} = 30$	DE	$8.723\ 8\text{e}-2$	$1.818\ 9\text{e}-1$	$1.370\ 1\text{e}-1$	$1.342\ 8\text{e}-1$	$3.002\ 3\text{e}-2$
$z_0 = 0$	$v_{z0} = 0$	ABC	$8.723\ 8\text{e}-2$	$1.039\ 6\text{e}-1$	$8.724\ 3\text{e}-2$	$9.017\ 7\text{e}-2$	$5.682\ 2\text{e}-3$
		HS	$9.278\ 5\text{e}-2$	$1.113\ 1\text{e}-1$	$9.719\ 1\text{e}-2$	$9.908\ 9\text{e}-2$	$6.281\ 6\text{e}-3$
		AHS	$8.723\ 8\text{e}-2$	$9.705\ 8\text{e}-2$	$8.724\ 3\text{e}-2$	$8.921\ 7\text{e}-2$	$3.310\ 4\text{e}-3$
$x_0 = 100\ 000$	$v_{x0} = -20$	PSO	$4.517\ 3\text{e}-1$	7.252 9	$5.192\ 7\text{e}-1$	1.830 0	2.569 5
$y_0 = -20\ 000$	$v_{y0} = 0$	DE	$4.834\ 5\text{e}-2$	$2.734\ 8\text{e}-1$	$7.863\ 5\text{e}-2$	$9.781\ 0\text{e}-2$	$6.633\ 1\text{e}-2$
$z_0 = 0$	$v_{z0} = 0$	ABC	$4.834\ 5\text{e}-2$	$6.279\ 4\text{e}-2$	$4.834\ 5\text{e}-2$	$4.978\ 9\text{e}-2$	$4.569\ 2\text{e}-3$
		HS	$5.302\ 1\text{e}-2$	$7.195\ 1\text{e}-2$	$5.853\ 9\text{e}-2$	$5.932\ 9\text{e}-2$	$5.270\ 0\text{e}-3$
		AHS	$4.834\ 5\text{e}-2$	$5.644\ 9\text{e}-2$	$4.834\ 5\text{e}-2$	$4.930\ 8\text{e}-2$	$2.554\ 6\text{e}-3$

基于燃料最优和时间最优的考虑, 将目标函数的权重 λ 分别设为 0, 0.01, 0.1, 0.25, 0.5, 0.75, 0.8, 0.95, 1, 利用本文算法进行进一步

求解, 所得结果见表 3, 并绘制了不同权重 λ 时算法的优化曲线, 分析燃料及时间的重要性和合理性.

表 3 不同权重时的优化结果
Table 3 The optimization results with different weights

初始位置/m	初始速度/(m·s ⁻¹)	权重 λ	最好值	最差值	中间值	平均值	方差
$x_0 = 70\ 000$ $y_0 = -30\ 000$ $z_0 = 0$	$v_{x0} = -40$ $v_{y0} = 30$ $v_{z0} = 0$	0.01	7.972 0e-2	8.475 9e-2	7.999 8e-2	8.049 4e-2	1.534 4e-3
		0.10	8.723 8e-2	9.705 8e-2	8.724 3e-2	8.921 7e-2	3.310 4e-3
		0.25	9.976 7e-2	1.097 2e-1	9.976 7e-2	1.019 7e-1	3.989 4e-3
		0.50	1.206 5e-1	1.609 0e-1	1.208 4e-1	1.367 9e-1	2.075 1e-2
		0.75	1.415 3e-1	1.817 7e-1	1.415 3e-1	1.536 0e-1	1.943 5e-2
$x_0 = 100\ 000$ $y_0 = -20\ 000$ $z_0 = 0$	$v_{x0} = -20$ $v_{y0} = 0$ $v_{z0} = 0$	0.90	1.540 6e-1	1.942 9e-1	1.581 5e-1	1.709 7e-1	2.022 2e-2
		0.01	4.083 4e-2	4.400 3e-2	4.166 9e-2	4.178 0e-2	9.939 4e-4
		0.10	4.834 5e-2	5.644 9e-2	4.834 5e-2	4.930 8e-2	2.554 6e-3
		0.25	6.086 1e-2	8.803 5e-2	6.086 3e-2	6.885 7e-2	1.234 7e-2
		0.50	8.172 3e-2	1.088 8e-1	8.172 3e-2	9.258 5e-2	1.402 3e-2
		0.75	1.025 8e-1	1.297 3e-1	1.075 4e-1	1.135 9e-1	1.250 7e-2
		0.90	1.151 0e-1	1.422 3e-1	1.286 7e-1	1.286 7e-1	1.430 0e-2

由表 2 可以看出,本文算法所得的结果是最好的.无论是最好值、最差值、中间值,还是平均值和方差,本文算法都比 PSO 算法、DE 算法、HS 算法、ABC 算法要好,表现出良好的优化性能,说明本文算法能够有效地求解航空器最优交会问题,并获得有竞争优势的结果.

由表 3 可知,随着时间权重的增加,最优指标也在逐渐增加,但增加的幅度相对较小,说明时间因素是不可忽略的.当权重 $\lambda < 0.5$ 时,最优指标的数量级保持在 10^{-2} ,当权重 $0.5 < \lambda < 1$ 时,最优指标的数量级保持在 10^{-1} .这表明在有限时间的要求下,最优的燃料消耗起到决定性的作用.

4 结 论

本文针对航天器最优交会问题,基于 C-W 模型建立一种有效合理的燃料时间混合指标,并提出一种改进和声搜索(AHS)算法进行求解.在 AHS 算法中,应用全局均匀学习操作取代 HS 算法中的基音调整,有效地平衡算法的全局搜索和局部搜索.通过最优交会实例,测试了 AHS 算法的有效性,数值统计结果表明 AHS 算法优于其他智能优化算法,具有良好的寻优潜力.

参考文献:

[1] Fehse W. Automated rendezvous and docking of spacecraft [M]. Oxford City: Cambridge University Press, 2003.
[2] 张琪新,王士星. 分布估计算法在航天器近距离最优交会中的应用[J]. 计算机工程与科学, 2012, 34(5): 89-94.
(Zhang Qi-xin, Wang Shi-xing. Application of the EDA algorithm in the near range optimal rendezvous for spacecrafts

[J]. Computer Engineering & Science, 2012, 34(5): 89-94.)
[3] 王华,唐国金. 用遗传算法求解双冲量最优交会问题[J]. 中国空间科学技术, 2003(1): 26-30.
(Wang Hua, Tang Guo-jin. Solving optimal rendezvous using two impulses based on genetic algorithms [J]. Chinese Space Science and Technology, 2003(1): 26-30.)
[4] 戴光明,李晖. DE 算法在空间交会中的应用[J]. 上海航天, 2007(3): 46-49.
(Dai Guang-ming, Li Hui. Study on application of differential evolution algorithm in space rendezvous [J]. Aerospace Shanghai, 2007(3): 46-49.)
[5] Geem Z W, Kim J H, Loganathan G V. A new heuristic optimization algorithm: harmony search [J]. Simulation, 2001, 76(2): 60-68.
[6] Kennedy J, Eberhart R. Particle swarm optimization [C]// Proceedings of IEEE International Conference on Neural Networks. Perth, 1995: 1942-1948.
[7] Storn R, Price K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces [J]. Journal of Global Optimization, 1997, 11(4): 341-359.
[8] Karaboga D, Basturk B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm [J]. Journal of Global Optimization, 2007, 39(3): 459-471.
[9] Wang L, Yang R X, Xu Y, et al. An improved adaptive binary harmony search algorithm [J]. Information Sciences, 2013, 232: 58-87.
[10] Mahdavi M, Fesanghary M, Damangir E. An improved harmony search algorithm for solving optimization problems [J]. Applied Mathematics and Computation, 2007, 188(2): 1567-1579.
[11] Chen J, Pan Q K, Li J Q. Harmony search algorithm with dynamic control parameters [J]. Applied Mathematics and Computation, 2012, 219(2): 592-604.
[12] Yadav P, Kumar R, Panda S K, et al. An intelligent tuned harmony search algorithm for optimization [J]. Information Sciences, 2012, 196: 47-72.