

## 论文

# 多阶段协同求解卫星舱耦合布局优化问题的双邻域人工蜂群算法

黄振东<sup>①②</sup>, 肖人彬<sup>①\*</sup>

① 华中科技大学自动化学院, 武汉 430074

② 湖北经济学院统计学院, 武汉 430205

\* 通信作者. E-mail: rbxiao@hust.edu.cn

收稿日期: 2015-05-20; 接受日期: 2015-06-23; 网络出版日期: 2015-11-30

国家自然科学基金项目 (批准号: 60974076)、高等学校博士学科点专项科研项目 (批准号: 200804870070) 和湖北省教育厅科学技术研究计划指导性项目 (批准号: B2015059) 资助

**摘要** 针对三维卫星舱布局优化问题 (three-dimensional satellite module layout optimization problem, 3DSMLOP), 本文提出了一种具有双邻域的改进人工蜂群算法, 并将其与多阶段求解策略结合, 形成多阶段双邻域人工蜂群算法 (multi-stage dual neighborhood artificial bee colony algorithm, MS-DABC). 3DSMLOP 是一个复杂的多约束耦合问题, 其解空间是非连续、非线性、多模态的. MS-DABC 将 3DSMLOP 分解为多个子系统, 并根据优化目标在各个子系统之间的耦合关系, 将优化过程分为两个阶段. 第一阶段, 针对无耦合的优化目标, 各个子系统利用具有双邻域结构的改进人工蜂群算法独立进行优化. 在第一阶段求得的最优解的基础上, 第二阶段采用一般人工蜂群算法来优化子系统的旋转角度, 利用各个子系统之间的协同旋转来解决耦合的优化目标. 仿真实验结果表明, 该算法求解复杂的卫星舱布局问题非常有效, 在 3DSMLOP 算例求解上性能突出.

**关键词** 卫星舱布局 人工蜂群算法 耦合问题 多阶段 双邻域

## 1 引言

卫星舱布局优化问题 (satellite module layout optimization problem, SMLOP) 研究给定的设备和仪器在卫星舱的有限空间中的合理布局, 使之满足卫星舱的设计要求和限制, 其属于带性能约束的布局优化问题.

对于 SMLOP, 大多数研究是以一类自旋的返回式卫星舱为背景的二维带平衡约束的布局优化问题<sup>[1~8]</sup>. 对于三维卫星舱布局优化问题 (3DSMLOP), 研究相对较少. 相对于二维带平衡约束的布局优化问题, 由于 3DSMLOP 布局物多, 并带有更为复杂的约束和优化目标, 其求解要困难得多. 本文主要以一简化国际商业通讯卫星为背景, 研究求解 3DSMLOP 的高效方法.

对于三维复杂布局问题, 国内外学者做了许多的工作, 主要求解方法有启发式算法、智能算法、启发式或局部搜索算法和智能算法相结合的混合算法等. 黄文奇等<sup>[9]</sup>提出了一种拟人启发式算法求解

引用格式: 黄振东, 肖人彬. 多阶段协同求解卫星舱耦合布局优化问题的双邻域人工蜂群算法. 中国科学: 信息科学, 2016, 46: 193-211, doi: 10.1360/N112014-00301

长方体布局问题. 张德富等<sup>[10]</sup>提出了基于块装载的基础启发式方法, 并将其与模拟退火算法结合形成混合算法求解三维装箱问题. 李未等<sup>[11]</sup>考虑了带有时间调度的三维装箱问题, 提出了一种基于贪心策略的启发式方法. Szykman 和 Cagan<sup>[12]</sup>利用模拟退火算法求解多目标和多约束的三维布局问题. Braun 等<sup>[13]</sup>采用协同优化方法求解单级轨道运载火箭的布局设计问题, 但耦合问题未得到较好的解决. Grignon 和 Fadel<sup>[14]</sup>提出一个 Pareto 遗传算法, 求解带约束的小卫星配置设计问题. 该算法类似于多群体并行遗传算法或分组遗传算法, 利用多个种群搜索, 在保证遗传算法收敛性的前提下, 尽量保证各种群之间的差异性, 避免陷入同一局部最优值陷阱, 从而找到布局问题的多个 Pareto 最优解, 但其研究的布局问题规模较小 (包含 7 个待布物). de Sousa 等<sup>[15]</sup>利用广义极值优化算法求解飞船内部电子设备的复杂布局问题, 该问题为多目标优化设计问题.

在 3DSMLOP 的研究方面, Sun 等<sup>[16]</sup>以一简化国际商业通讯卫星 (INTELSAT-III) 为背景, 首先设计向心平衡法, 将布局物分配到各个承压板表面, 然后利用蚁群优化算法对各承压板表面的布局物进行布局. Zhang 等<sup>[17]</sup>采用 Hopfield 神经网络算法分配组件到各个承压板表面, 将遗传算法和粒子群算法分别与准主成分分析法混合求解详细布局. Wang 等<sup>[18]</sup>在文献 [17] 的分配方案上, 将散射搜索与合作协同进化的体系结构结合, 形成协同进化散射搜索算法, 并将其与局部搜索算法 Nelder-Mead 复合形法结合, 进行各承载板表面的详细布局. 为了避免进化算法过早地陷入早熟问题, Teng 等<sup>[19]</sup>提出了双系统变粒度合作进化算法, 求解 3DSMLOP, 取得目前最好的结果.

3DSMLOP 非常难于求解, 其主要原因在于: (1) 这类问题系统规模大 (不少于 50 个布局物), 因此解空间维数巨大; (2) 具有较复杂的优化目标和约束条件, 特别是耦合的优化目标和约束, 使得解空间呈现出非线性、多模态、不连续的特征, 具有大量的局部最优解, 甚至是大量不可行的局部最优解, 直接在其解空间中搜索满足所有约束且目标函数较优的解非常困难. 文献 [17~19] 均直接考虑优化所有的目标, 尽管在启发式求解和智能算法的改进上做了大量的工作, 也取得了不错的效果, 但其求解质量还有改进的空间. 本文从两个方面进一步提高求解质量. (1) 针对 3DSMLOP, 提出了两阶段优化策略. 首先, 根据卫星舱布局特点, 对问题进行系统分解, 并按照耦合关系将整个优化过程分为两个阶段. 第一阶段针对无耦合的优化目标, 各个子系统独立进行优化; 第二阶段通过各个子系统协同旋转解决耦合约束限制. (2) 在每个阶段中利用具有较强全局搜索能力的人工蜂群算法进行详细优化. 其中第一阶段的求解质量直接影响第二阶段的优化结果, 为了提高第一阶段的解的质量, 提出双邻域人工蜂群算法 (dual neighborhood artificial bee colony algorithm, DABC). 通过对各种算法的数值实验比较显示, 对于具有复杂耦合约束的 3DSMLOP, 本文提出的多阶段优化策略和双邻域蜂群算法非常有效.

本篇文章的组织如下: 第 2 节描述了 3DSMLOP 的数学模型; 第 3 节给出了干涉量的定义以及分离算法; 第 4 节介绍了系统分解和多阶段优化策略; 第 5 节设计了双邻域蜂群算法, 并给出双邻域蜂群算法结合多阶段优化策略的 MS-DABC 算法描述; 第 6 节给出了数值实验的结果与分析; 最后, 第 7 节总结了本文的创新.

## 2 问题的数学模型

一个简化的国际商业通信卫星布局优化问题可以描述如下<sup>[18]</sup>.

对于给定的  $N$  个组件将被安置在如图 1(a) 所示的圆柱形卫星模块中, 连接模块中的一个立柱上的两个轴承板的 4 个表面 ( $S_1 \sim S_4$ ) 用于放置所有组件, 如图 1(b). 设计目标是整个卫星系统的转动惯量之和应尽可能小, 同时满足不干涉约束、静稳定性约束和惯性夹角约束.

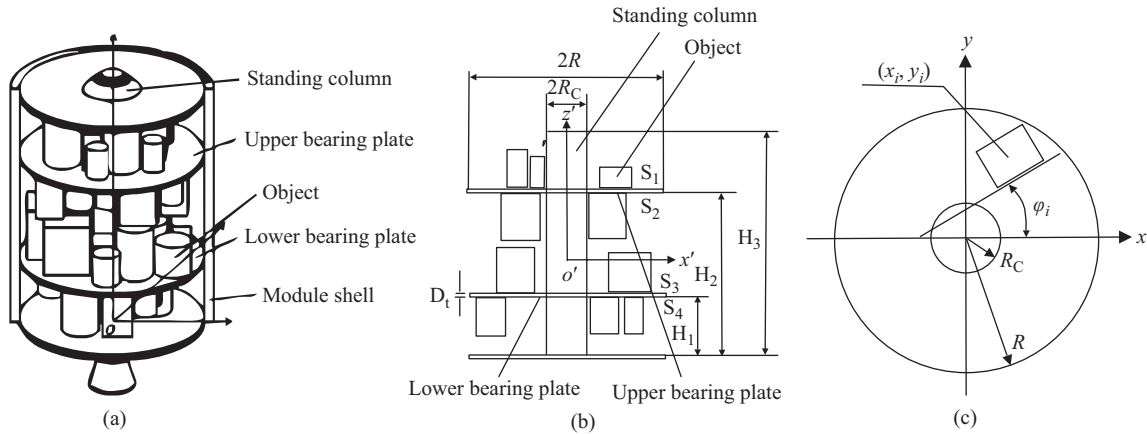


图 1 简化国际商业通信卫星舱布局示意图, 修改自文献 [18]

Figure 1 Layout diagram of simplified international commercial satellite module, modified from [18]

为了方便描述, 3 个 Decare 坐标系按文献 [16,17] 中的定义. 如图 1 所示, 参考坐标系  $oxyz$  用来确定各待布物的位置, 计算卫星的质心坐标等; 星体坐标系  $o'x'y'z'$  用来计算转动惯量矩阵及惯量夹角, 其坐标原点位于整个系统的质心; 局部坐标系  $o''x''y''z''$  用于计算各待布物对其自身轴的转动惯量矩阵.

设待布物集合  $A = \{A_1, A_2, \dots, A_N\}$ ,  $N$  为待布物总数. 各待布物均简化为质量分布均匀的长方体或圆柱体. 第  $i$  个待布物状态可表示为  $A_i(p_i, \varphi_i, d_i)$ , 其中  $p_i(x_i, y_i, z_i) \in R^{3N}$  为  $A_i$  质心坐标 (相对参考坐标系);  $\varphi_i$  为  $A_i$  在图 1(c) 所示的方向角 (逆时针方向为正).  $d_i \in R^3$  为待布物的几何尺寸向量, 对长方体,  $d_i = (a_i, b_i, h_i) \in R^3$  中的  $a_i, b_i, h_i$  分别为长、宽和高, 限定长方体之间正交放置, 即  $\varphi_i \in \{0, \pi/2\}$ ; 对圆柱体,  $d_i = (r_i, 0, h_i) \in R^3$ , 其中  $r_i, h_i$  分别为底面半径和高. 为叙述方便, 将所有舱体固定件记为  $A_0$ , 包括舱体壁、承载隔板和舱内支撑立柱等, 其质量记为  $m_0$ . 根据上述技术要求, 一个布局方案可由  $X = \{(x_i, y_i, z_i, \varphi_i) \mid i = 1, 2, \dots, N\} \in R^{4N}$  给出, 则 3DSMLOP 布局优化模型如下:

$$\min f(X) = I_{x'x'} + I_{y'y'} + I_{z'z'}, \tag{1a}$$

s.t.

(1) 不干涉约束

$$g_1(X) = \sum_{i=0}^{N-1} \sum_{j=i+1}^N \Delta A_{ij} = 0, \tag{1b}$$

(2) 静平衡约束

$$\begin{cases} g_2(X) = |x_c - x_e| \leq \delta x_e, \\ g_3(X) = |y_c - y_e| \leq \delta y_e, \\ g_4(X) = |z_c - z_e| \leq \delta z_e, \end{cases} \tag{1c}$$

(3) 惯性夹角约束

$$\begin{cases} g_5(X) = |\theta_{x'}(X)| \leq \delta \theta_{x'}, \\ g_6(X) = |\theta_{y'}(X)| \leq \delta \theta_{y'}, \\ g_7(X) = |\theta_{z'}(X)| \leq \delta \theta_{z'}, \end{cases} \tag{1d}$$

其中  $I(X) = (I_{x'x'}, I_{y'y'}, I_{z'z'})$  为卫星系统对星体坐标系  $o'x'y'z'$  在 3 个坐标轴方向上的转动惯量;  $\Delta A_{ij}$  表示待布局物  $A_i$  与  $A_j$  之间的干涉量, 当  $i = 0$  时,  $\Delta A_{0j}$  表示舱体固定件  $A_0$  (这里指的是舱体壁和舱内支撑立柱) 与待布局物  $A_j$  之间的干涉量;  $O_c(x_c, y_c, z_c)$  为卫星系统的实际质心坐标,  $O_e(x_e, y_e, z_e)$  为卫星系统质心的期望位置,  $(\delta x_e, \delta y_e, \delta z_e)$  为卫星系统质心的实际位置与期望位置在参考坐标系 3 个坐标轴方向上的误差允许值;  $\theta_{x'}, \theta_{y'}$  和  $\theta_{z'}$  为卫星系统的惯性主轴与星体坐标轴  $o'x'y'z'$  3 个坐标轴方向的夹角, 即惯性夹角;  $(\delta\theta_{x'}, \delta\theta_{y'}, \delta\theta_{z'})$  为卫星系统在 3 个坐标轴上的惯性夹角的误差允许值.  $I(X) = (I_{x'x'}, I_{y'y'}, I_{z'z'})$ ,  $O_c(x_c, y_c, z_c)$  与  $\theta_{x'}, \theta_{y'}$  和  $\theta_{z'}$  的计算见文献 [17,18].

本文采用文献 [17,18] 中各个布局物在承载板面上的分配方案, 即布局物的  $z$  坐标已经确定. 因此, 第  $i$  个布局物的状态由其坐标  $(x_i, y_i)$  和旋转角度  $\varphi_i$  确定, 即系统解可表示为  $X = \{(x_i, y_i, \varphi_i) | i = 1, 2, \dots, N\} \in R^{3N}$ . 采用线性加权和法将问题 (1) 转化为单目标优化问题. 设  $f_1(X) = f(X)$ ,  $f_2(X) = g_1(X)$ ,  $f_3(X) = |g_2(X)| + |g_3(X)| + |g_4(X)|$ ,  $f_4(X) = |g_5(X)| + |g_6(X)| + |g_7(X)|$ , 则整个布局方案的评价, 即全局目标函数的计算可表示如下:

$$\min F(X) = w_1 \lambda_1 f_1(X) + w_2 \lambda_2 f_2(X) + w_3 \lambda_3 f_3(X) + w_4 \lambda_4 f_4(X), \quad (2)$$

其中  $\lambda_i (i = 1, 2, 3, 4)$  和  $w_i (i = 1, 2, 3, 4)$  分别为各分目标的归一化因子和权重系数.

### 3 干涉量的计算处理

#### 3.1 干涉量的计算

本文利用两物体嵌入深度的平方来定义干涉量. 嵌入深度是机器人和计算视觉领域的一个重要概念, 如果两个物体相互嵌入, 则嵌入深度就是分离它们的最短距离; 如果两物体不干涉, 则嵌入深度为 0.

对于一个给定的集合  $A$ , 记  $\text{int}(A)$  表示集合  $A$  的内部,  $\partial A$  表示集合的边界, 为了简便, 记  $A = \text{int}(A) \cup \partial A$ .

**定义1** 给定布局物  $A$  和平移向量  $v = (v_x, v_y)$ , 定义平移函数  $\oplus$  为

$$A \oplus v = \{(p'_x + v_x, p'_y + v_y) | p' = (p'_x, p'_y) \in A\}.$$

**定义2** 两布局物  $A, B$  的嵌入深度定义为

$$\text{PD}(A, B) = \min\{\|v\| | \text{int}(A) \cap (B \oplus v) = \emptyset\},$$

舱体固定件  $A_0$  (这里指的是舱体壁和舱内支撑立柱) 与布局物  $A$  的嵌入深度定义为

$$\begin{aligned} \text{PD}(A_0, A) &= \min\{\|v\| | \text{int}(A_0) \cap (A \oplus v) = \emptyset\} \\ &= \begin{cases} \max\{\|p\| | p \in \text{int}(A)\} - R, & \text{if } \max\{\|p\| | p \in \text{int}(A)\} > R, \\ R_c - \min\{\|p\| | p \in \text{int}(A)\}, & \text{if } \min\{\|p\| | p \in \text{int}(A)\} < R_c, \\ 0, & \text{else,} \end{cases} \end{aligned}$$

其中  $\|\cdot\|$  表示欧氏距离,  $R_c$  为中心立柱半径,  $R$  为卫星舱承载板半径.

嵌入深度  $PD(A, B)$  可以通过用类似临界多边形的方法求解  $A, B$  的临界图形 (no-fit-pattern, 简记为 NFP)  $NFP_{AB}$  来得到.  $NFP_{AB}$  可以通过下述方式构造: 选定  $B$  上的一个点  $O_B$  作为参考点, 固定  $A$ , 将  $B$  围绕  $A$  保持相切做刚体运动一周, 参考点  $O_B$  的移动轨迹所围成的区域 (不含边界) 就是  $B$  相对于  $A$  的临界图形, 记为  $NFP_{AB}$ , 其边界集记为  $\partial NFP_{AB}$ . 如果参考点  $O_B \in NFP_{AB}$  时,  $PD(A, B)$  等于  $O_B$  到  $\partial NFP_{AB}$  的最短距离.

定义 2 中的向量  $v$  称为嵌入向量, 记为  $PV(A, B)$  (或  $PV(A_0, A)$ ). 嵌入向量可以在求解嵌入深度的过程中获得, 如果嵌入深度为 0, 则嵌入向量为  $(0, 0)$ . 嵌入向量有可能不唯一, 我们从最小的  $\|v\|$  中任取一个作为嵌入向量.

两布局物  $A_i, A_j$  的干涉量定义为  $\Delta A_{ij} = PD(A_i, A_j)^2 = \|v\|^2, (1 \leq i \neq j \leq N)$ .

舱体固定件  $A_0$  与布局物  $A_j$  的干涉量定义为  $\Delta A_{0j} = PD(A_0, A_j)^2, (1 \leq j \leq N)$ .

对于给定一个布局  $X$ , 布局物  $A_i$  对  $A_j$  的弹力以及舱体固定件  $A_0$  对  $A_j$  的弹力分别定义为

$$\tilde{f}_{ij}(X) = 2PV(A_i, A_j), (1 \leq i \neq j \leq N),$$

$$\tilde{f}_{0j}(X) = 2PV(A_0, A_j), (1 \leq j \leq N),$$

其中 2 为弹性系数, 很显然, 我们有  $\tilde{f}_{ij}(X) = -\tilde{f}_{ji}(X)$ , 则布局物  $A_j$  所受弹力的合力为

$$\tilde{f}_j(X) = \sum_{\substack{0 \leq i \leq n \\ i \neq j}} \tilde{f}_{ij}(X).$$

### 3.2 分离算法

由于不干涉约束是布局优化问题需要满足的必要条件, 对于给定的一个布局解  $X$ , 我们只考虑在平移情况下优化干涉量

$$\min g_1(X) = \sum_{0 \leq i < j \leq N} \Delta A_{ij}, \tag{3}$$

**定理1** 记  $\nabla_k = (\partial/\partial x_k, \partial/\partial y_k)$ , 则  $\nabla_k g_1(X) = -\tilde{f}_k(X)$ . 即  $g_1(X)$  对第  $k$  个布局物  $(x_k, y_k)$  坐标的负梯度为第  $k$  个布局物所受的合力.

**证明** 根据定义,  $\Delta A_{ij} (1 \leq i < j \leq N)$  的梯度为

$$\nabla_i(\Delta A_{ij}) = -\nabla_j(\Delta A_{ij}) = 2PV(A_i, A_j) = \tilde{f}_{ij}(X),$$

$$\nabla_k(\Delta A_{ij}) = 0, \quad k \in \{1, 2, \dots, N\} \setminus \{i, j\}.$$

$\Delta A_{0j} (1 \leq j \leq N)$  的梯度为

$$\nabla_j(\Delta A_{0j}) = -2PV(A_0, A_j) = -\tilde{f}_{0j}(X),$$

$$\nabla_k(\Delta A_{0j}) = 0, \quad k \in \{1, 2, \dots, N\} \setminus \{j\}.$$

因此

$$\nabla_k g_1(X) = \nabla_k \left( \sum_{0 \leq i < j \leq N} \Delta A_{ij} \right) = \sum_{0 \leq i < j \leq N} \nabla_k \Delta A_{ij}$$

$$\begin{aligned}
 &= \sum_{0 \leq i < k} \nabla_k \Delta A_{ik} + \sum_{k < j \leq N} \nabla_k \Delta A_{kj} \\
 &= \sum_{0 \leq i < k} (-\tilde{f}_{ik}(X)) + \sum_{k < j \leq N} \tilde{f}_{kj}(X) \\
 &= \sum_{0 \leq i < k} (-\tilde{f}_{ik}(X)) + \sum_{k < j \leq N} (-\tilde{f}_{jk}(X)) \\
 &= - \sum_{\substack{0 \leq i \leq n \\ i \neq k}} \tilde{f}_{ik}(X) \\
 &= -\tilde{f}_k(X) \quad k \in \{1, 2, \dots, N\}.
 \end{aligned}$$

根据定理 1, 我们可以利用干涉量的梯度信息快速分离布局物, 分离算法描述如下:

- (1) 对当前的布局解  $X$ , 根据干涉定义计算式 (3) 中的  $g_1(X)$ ;
- (2) 根据合力的定义计算各个布局物所受的合力  $\tilde{f}_j(X)$ , 得到  $g_1(X)$  的梯度

$$\nabla g_1(X) = \{-\tilde{f}_1(X), -\tilde{f}_2(X), \dots, -\tilde{f}_N(X)\};$$

(3) 根据 (1) 和 (2) 中得到的  $g_1(X)$  和它的梯度  $\nabla g_1(X)$ , 利用 L-BFGS (limited memory BFGS) 方法求解非线性优化问题 (3).

## 4 系统分解及耦合优化处理

### 4.1 系统分解

对于 3DSMLOP 这类具有多目标优化的复杂的耦合系统, 若直接考虑所有的布局物和所有的优化量, 求解难度较大. 因此, 我们采取复杂系统的分解 - 综合设计方法, 根据卫星舱系统的特点, 将其按 4 个承载面分成 4 个子系统, 相对于直接考虑全部布局物, 在每个子系统中考虑布局可有效降低布局规模. 以下给出 3DSMLOP 子系统状态变量的定义, 分析各子系统级布局问题之间的耦合关系, 为两阶段布局提供依据.

3DSMLOP 中第  $i$  个子系统的整体布局特征可由状态向量  $u^i$  描述, 定义如下:

$$u^i = (m^i, x^i, y^i, z^i, I_{xx}^i, I_{yy}^i, I_{zz}^i, I_{xy}^i, I_{yz}^i, I_{xz}^i) \in R^{10},$$

其中  $i(=1,2,3,4)$  为布局子系统序号,  $m^i, (x^i, y^i, z^i)$  和  $I_{xx}^i, I_{yy}^i, I_{zz}^i, I_{xy}^i, I_{yz}^i, I_{xz}^i$  分别为第  $i$  个子系统中所有待布物的总质量, 总质心在参考坐标系  $oxyz$  中的位置坐标和第  $i$  个子系统中待布物相对于参考坐标系的转动惯量和惯性积.

记  $A^i = \{A_1^i, A_2^i, \dots, A_{N^i}^i\}$  ( $i=1, 2, 3, 4$ ) 表示第  $i$  个子系统的布局物集合,  $X^i$  为子系统对应的解. 则有  $A = A^1 \cup A^2 \cup A^3 \cup A^4, X = X^1 \cup X^2 \cup X^3 \cup X^4$ . 记  $u^0$  为航天器固定构件的状态特征向量, 即

$$u^0 = (m^0, x^0, y^0, z^0, I_{xx}^0, I_{yy}^0, I_{zz}^0, I_{xy}^0, I_{yz}^0, I_{xz}^0) \in R^{10},$$

在布局优化过程中,  $u^0$  为常向量.

## 4.2 基于耦合关系的优化目标分解

对于多约束优化的 3DSMLOP, 根据第 2 节的线性加权和法, 其全局适应度函数可表示如下:

$$F(X) = w_1\lambda_1 f_1(X) + w_2\lambda_2 f_2(X) + w_3\lambda_3 f_3(X) + w_4\lambda_4 f_4(X),$$

其中, 相交量  $f_2(X) = g_1(X)$  各个系统之间没有耦合, 而静平衡量和惯性夹角  $f_3(X) = |g_2(X)| + |g_3(X)| + |g_4(X)|$ ,  $f_4(X) = |g_5(X)| + |g_6(X)| + |g_7(X)|$ , 在各个子系统之间具有耦合.

对于目标函数 (1a), 根据转动惯量、惯性积平行移轴原理可知

$$f_1(X) = f(X) = I_{x'x'} + I_{y'y'} + I_{z'z'} = I_{xx} + I_{yy} + I_{zz} - 2(x_c^2 + y_c^2 + z_c^2) \sum_{i=0}^N m_i,$$

其中相对于参考坐标系  $oxyz$  的转动惯量之和  $I_{xx} + I_{yy} + I_{zz}$ , 在各子系统之间没有耦合, 而  $2(x_c^2 + y_c^2 + z_c^2) \sum_{i=0}^N m_i$ , 这一部分与系统质心有关, 各个系统之间存在耦合.

因此, 全局目标函数可分解为

$$F(X) = w_1\lambda_1(I_{xx} + I_{yy} + I_{zz}) + w_2\lambda_2 f_2(X) - 2w_1\lambda_1(x_c^2 + y_c^2 + z_c^2) \sum_{i=0}^N m_i + w_3\lambda_3 f_3(X) + w_4\lambda_4 f_4(X),$$

其中,  $w_1\lambda_1(I_{xx} + I_{yy} + I_{zz}) + w_2\lambda_2 f_2(X)$  这一部分各个子系统之间没有耦合, 而  $-2w_1\lambda_1(x_c^2 + y_c^2 + z_c^2) \sum_{i=0}^N m_i + w_3\lambda_3 f_3(X) + w_4\lambda_4 f_4(X)$  各个子系统之间存在耦合.

## 4.3 两阶段优化策略

由于 3DSMLOP 是一个具有多目标多约束优化的复杂的优化问题, 若同时考虑目标函数和全部约束条件直接以  $F(X)$  为评价函数进行优化, 不仅会使得搜索空间为非连续、非线性、多模态的空间, 导致搜索易于陷入局部最优, 而且由于约束过多使得满足全部约束的合法解在解空间的比例过小, 从而增加搜索合法解的难度. 基于分解协同求解的思路, 我们根据布局子系统约束耦合关系, 将整个优化求解过程分为两个阶段.

### 4.3.1 第一阶段 无耦合目标的优化

在第一阶段, 不考虑各个子系统具有耦合的量, 只优化无耦合的目标, 即

$$\min F_1(X) = w_1\lambda_1(I_{xx} + I_{yy} + I_{zz}) + w_2\lambda_2 f_2(X),$$

由于  $F_1(X)$  各个子系统之间没有耦合, 故

$$F_1(X) = \sum_{i=1}^4 (w_1\lambda_1(I_{xx}^i + I_{yy}^i + I_{zz}^i) + w_2\lambda_2 f_2^i(X^i)),$$

其中,  $f_2(X^i) = \sum_{k=1}^{N^i-1} \sum_{j=k+1}^{N^i} \Delta A_{kj}^i = 0$  表示第  $i$  个子系统内布局的相交总量,  $X^i$  为子系统对应的布局解,  $N^i$  为子系统中布局物的个数. 故可以对各个子系统独立的优化

$$\min F_1(X^i) = w_1\lambda_1(I_{xx}^i + I_{yy}^i + I_{zz}^i) + w_2\lambda_2 f_2^i(X^i), \quad i = 1, 2, 3, 4. \quad (4)$$

第一阶段各个子系统独立优化, 其目的非常明显: 一方面, 各个子系统布局物个数大大减少, 从而子系统解空间的维数相应的变小, 另一方面, 由于约束的减少, 使得搜索空间非连续、非线性、多模态的性质减弱, 也使得满足当前约束的合法解比例增大, 从而能快速的搜索到较高质量的解. 当然, 这样得到的解不一定为合法解, 即不一定满足耦合约束, 我们在第二阶段子系统协同旋转中解决耦合约束.

在第一阶段的优化中, 各个子系统利用 DABC 算法独立的优化式 (4), 第  $i$  个子系统解  $X^i$  进行评价所得的适应度值为

$$\text{fit}(X^i) = 1/F_1(X^i). \quad (5)$$

尽管在各个子系统中布局物个数减少了, 同时约束条件也作了简化, 但其优化变量也仍然较多, 而第一阶段的解的质量会直接影响第二阶段的优化效果. 为了提高第一阶段的求解质量, 本文在第一阶段中提出双邻域人工蜂群算法 DABC. 具体的 DABC 描述见 5.2 小节.

#### 4.3.2 第二阶段 耦合约束求解

在第一阶段中, 系统对不干涉约束量和相对于参考坐标系的  $oxyz$  的转动惯量进行了优化, 但子系统具有耦合的静平衡量和惯性夹角没能优化, 其不一定满足约束条件. 第二阶段通过各个子系统绕卫星舱中心立柱 (参考坐标系的  $oxyz$  的  $z$  轴) 协同旋转, 来优化系统静不平衡量和惯性夹角, 各个子系统在第一阶段优化的干涉量和相对于参考坐标系的  $oxyz$  的转动惯量在这一阶段不会发生改变. 因此, 第二阶段优化目标函数如下:

$$\min F_2(X) = -2w_1\lambda_1(x_c^2 + y_c^2 + z_c^2) \sum_{i=0}^N m_i + w_3\lambda_3f_3(X) + w_4\lambda_4f_4(X). \quad (6)$$

各个子系统绕卫星舱中心立柱 (参考坐标系的  $oxyz$  的  $z$  轴) 协同旋转优化式 (6) 的过程中, 第一阶段已优化的量是不会发生改变的. 因为在子系统旋转过程中, 各个子系统内的待布物的相对位置不变, 故系统干涉量不会发生改变. 又因为

$$\begin{aligned} I_{xx} + I_{yy} + I_{zz} &= \sum_{j=0}^N (J_{x''j} \cdot \cos^2 \varphi_j + J_{y''j} \cdot \sin^2 \varphi_j) + \sum_{j=0}^N m_j((y_j)^2 + (z_j)^2) \\ &\quad + \sum_{j=0}^N (J_{y''j} \cdot \cos^2 \varphi_j + J_{x''j} \cdot \sin^2 \varphi_j) + \sum_{j=0}^N m_j((x_j)^2 + (z_j)^2) \\ &\quad + \sum_{j=0}^N J_{z''j} + \sum_{j=0}^N m_j((x_j)^2 + (y_j)^2) \\ &= \sum_{j=0}^N (J_{x''i} + J_{y''i} + J_{z''i}) + 2 \sum_{j=0}^N m_j((x_j)^2 + (y_j)^2 + (z_j)^2) \\ &= C + \sum_{j=0}^N 2m_j((x_j)^2 + (y_j)^2) = C + \sum_{j=0}^N 2m_j\rho_j, \end{aligned}$$

其中  $C = \sum_{j=0}^N (J_{x''i} + J_{y''i} + J_{z''i}) + 2 \sum_{j=0}^N m_j(z_j)^2$  在布局过程中为常数, 而  $\rho_j = (x_j)^2 + (y_j)^2$  表示第  $j$  个布局物质心到卫星舱中心立柱的距离, 在子系统绕卫星舱中心立柱的旋转下,  $\rho_j$  也保持不变, 从而系统相对于参考坐标系的  $oxyz$  的转动惯量之和  $I_{xx} + I_{yy} + I_{zz}$  在第二阶段的旋转中也保持不变.



设第一阶段求得的最优布局解为  $X_{\text{best}} = (X_{\text{best}}^1, X_{\text{best}}^2, X_{\text{best}}^3, X_{\text{best}}^4)$ , 记第  $i$  个子系统的旋转角为  $\alpha^i$ , 则这一阶段的系统解可表示为

$$\alpha = (\alpha^1, \alpha^2, \alpha^3, \alpha^4), \quad (7)$$

其中  $\alpha^i \in [0, 2\pi), i = 1, 2, 3, 4$ .

由于第二阶段优化参数较少, 我们利用基本 ABC 算法对  $\alpha$  进行优化, 其适应度评价函数为

$$\text{fit}(\alpha) = 1/F_2(X_{\text{best},\alpha}), \quad (8)$$

其中  $X_{\text{best},\alpha} = (X_{\text{best},\alpha^1}^1, X_{\text{best},\alpha^2}^2, X_{\text{best},\alpha^3}^3, X_{\text{best},\alpha^4}^4)$ ,  $X_{\text{best},\alpha^i}^i$  表示  $X_{\text{best}}^i$  旋转角度  $\alpha^i$  后得到的第  $i$  个子系统的解,  $X_{\text{best},0}^i = X_{\text{best}}^i$ .

通过对系统分解以及耦合性分析, 将整个优化过程分为两个阶段进行优化. 第一阶段各个子系统独立的优化无耦合目标, 大大减少了解空间维数, 并且由于没有考虑耦合目标, 从而有效减弱了解空间的非连续、非线性、多模态的性质, 故算法优化效果较好. 而耦合约束能通过第二阶段子系统的协同旋转得到解决, 并且其没有改变第一阶段优化的量. 在 6.2 小节的实验也验证, 对于具有多耦合约束的 3DSMLOP, 多阶段优化策略相对于单阶段优化策略能得到更好的优化结果.

## 5 双邻域人工蜂群算法

人工蜂群算法 (artificial bee colony, ABC) 是近年来发展出来的一种模拟蜜蜂觅食的仿生算法 [20], 由于其能较好平衡局部“挖掘”搜索与全局“勘探”搜索, 具有良好的搜索性能, 在一些 NP 难问题的求解上显示出 ABC 具有良好的全局优化能力 [21,22]. 已有研究证实在无约束的数值优化上, ABC 优于其他一些智能算法, 例如 PSO, GA 等等 [23]. 因此, 我们采用人工蜂群算法优化具体布局物坐标和子系统的旋转角度.

### 5.1 基本人工蜂群算法

基本 ABC 算法中, 人工蜂群包含 3 种个体: 雇佣蜂、观察蜂和侦查蜂. 每个雇佣蜂对应一个确定的蜜源并在迭代中对蜜源的邻域进行搜索.

初始化时, 按式 (9) 随机生成 SN 个蜜源 (等于雇佣蜂的数量) 并计算适应度函数值,

$$x_{ij} = x_{\min,j} + \text{rand}(0, 1)(x_{\max,j} - x_{\min,j}), \quad (9)$$

其中,  $x_i (i = 1, 2, \dots, \text{SN})$  为  $D$  维向量,  $D$  为优化参数的个数,  $j \in \{1, 2, \dots, D\}$ .

蜜蜂在其当前蜜源邻域内搜索新的蜜源, 基本 ABC 在蜜源附近搜索新蜜源的公式为

$$v_{ij} = x_{ij} + \varphi_{ij}(x_{ij} - x_{kj}), \quad (10)$$

式中,  $j \in \{1, 2, \dots, D\}$  是随机选择的某一个更新的维度,  $k \in \{1, 2, \dots, \text{SN}\}$ ,  $k$  为随机生成且  $k \neq i$ ,  $\varphi_{ij}$  为  $[-1, 1]$  之间的随机数.

观察蜂选择雇佣蜂的概率为

$$P_i = \text{fit}(x_i) / \sum_{n=1}^{\text{SN}} \text{fit}(x_n), \quad (11)$$

其中,  $\text{fit}(x_i)$  为第  $i$  个蜜源的适应值, 其对应蜜源的丰富程度. 解的适应值越好, 表示蜜源越丰富, 被观察蜂选择的概率越大.

为防止算法陷入局部最优, 当某蜜源迭代  $\text{limit}$  次没有改进时, 便放弃该蜜源, 并且将该蜜源记录在禁忌表中, 同时该蜜源对应的雇佣蜂转变为侦察蜂按式 (9) 随机产生一个新的位置代替原蜜源.

基本人工蜂群算法的控制参数主要有 3 个: 蜜源的个数 (与雇佣蜂或观察蜂相等) SN, 算法终止的最大进化数 (maximum evaluation number) MEN 以及  $\text{limit}$ .

## 5.2 双邻域蜂群算法 DABC

在基本的 ABC 算法中, 观察蜂和观察蜂均采用相同的邻域搜索新蜜源, 即根据式 (10) 在蜜源附近随机改变单个优化变量来搜索新解, 然后根据贪心策略更新蜜源. 当寻优问题在这种单一邻域下具有很多局部极值点, 基本 ABC 算法的贪心策略会过早陷入搜索空间中的局部最优, 而无法跳出, 导致搜索效率不高.

由于本文研究是多约束的复杂耦合布局问题, 其解空间很大, 并且具有大量的局部最优解, 甚至是不可行的局部最优解, 尽管我们采用了系统分解策略和优化目标的多阶段处理策略, 很大程度上简化了求解难度, 但在第一阶段的解空间仍然有许多的局部最优解, 采用单一更新邻域的基本 ABC 算法还是存在陷入局部最优的问题. 为此, 我们针对研究的布局问题, 设计了双邻域更新策略的 ABC 算法——DABC 算法, 相对于基本 ABC 算法中雇佣蜂和观察蜂采用相同的邻域搜索新蜜源, DABC 中雇佣蜂和观察蜂采用不同的邻域搜索新蜜源, 有效避免单一邻域容易陷入当前邻域的局部最优的问题, 使得算法具有一定的跳出局部最优的能力, 我们在 6.2 小节实验中给出了 DABC 算法与基本 ABC 算法的对比, DABC 优化能力改进明显.

### 5.2.1 分离算法的嵌入

由于不干涉约束是布局问题需要满足的必要条件, 为了避免蜜蜂对干涉的盲目搜索, 加快搜索速度, 每次更新解后, 均执行分离算法, 使得布局物快速分离, 再计算解的适应度值.

### 5.2.2 双邻域更新策略

第一阶段子系统独立优化过程中, 考虑第  $i$  个子系统, 其第  $j$  个蜜源对应的解为

$$X_j^i = \{(x_{jl}^i, y_{jl}^i, \varphi_{jl}^i) | l=1, 2, \dots, N^i\} \in \mathbb{R}^{3N^i},$$

其中  $N^i$  表示第  $i$  个子系统中的布局物个数.

#### (1) 原 ABC 平移邻域

根据基本 ABC 算法, 在蜜源附近搜索新蜜源的公式为

$$\tilde{x}_{jl}^i = x_{jl}^i + \varphi_{jl}[x_{jl}^i - x_{kl}^i], \quad \tilde{y}_{jl}^i = y_{jl}^i + \psi_{jl}[y_{jl}^i - y_{kl}^i]. \quad (12)$$

其中  $j \in \{1, 2, \dots, \text{SN}\}$  表示蜜源序号;  $k$  为  $[1, \text{SN}]$  之间的随机整数且  $k \neq j$ ,  $\varphi_{jl}, \psi_{jl}$  为  $[-1, 1]$  之间的随机数.

其实质是根据第  $k$  个蜜源的第  $l$  个布局物的坐标, 对第  $j$  个蜜源的第  $l$  个布局物的坐标做了一个平移来搜索新蜜源. 然而, 在布局问题中, 这种盲目的平移搜索在算法中后期大部分是无用的搜索, 其

很难打破布局的拓扑结构, 从而容易陷入当前邻域的局部最优, 影响搜索效果. 为了弥补这一缺点, 提出一个新的搜索邻域——交换邻域.

## (2) 交换邻域

随机交换蜜源的第  $k, l$  两个布局物的质心位置坐标来搜索新蜜源, 即

$$(\tilde{x}_{jl}^i, \tilde{y}_{jl}^i) = (x_{jk}^i, y_{jk}^i), \quad (\tilde{x}_{jk}^i, \tilde{y}_{jk}^i) = (x_{jl}^i, y_{jl}^i), \quad (13)$$

其中  $k, l$  为  $[1, N^i]$  中的随机整数且  $k \neq l$ .

原 ABC 平移邻域和交换邻域具有各自的优势和不足. 原 ABC 平移邻域能有效搜索布局物附近位置, 但当迭代一定程度后, 其大部分平移搜索是无用的, 很难打破当前布局物的拓扑结构. 交换邻域能有效改变当前布局的拓扑结构, 但其却无法在布局物附近位置进行搜索. 为了弥补单一邻域结构的不足, 我们提出双邻域蜂群算法 DABC. 在 DABC 中, 雇佣蜂按照原 ABC 平移邻域搜索新蜜源, 而观察蜂则根据交换邻域来搜索新蜜源. 这样对新蜜源的搜索, 既有能搜索布局物附近位置的平移搜索, 也有能快速改变当前布局拓扑结构的交换搜索, 有效地弥补了单一搜索邻域的不足. 第 6 节的实验也验证了相对于 ABC 算法, DABC 搜索能力更强.

### 5.3 多阶段双邻域蜂群 MS-DABC 算法描述

根据第 4 节的描述, 我们采用两阶段优化策略处理具有耦合约束的 3DSMLOP. 在第一阶段中, 由于优化变量较多, 并且其优化解的质量将直接影响第二阶段的求解效果. 因此, 用 DABC 算法进行求解. 在第二阶段中, 优化变量只有 4 个, 用基本的 ABC 算法进行求解.

MS-DABC 算法描述如下:

- 1 第一阶段各个子系统利用 DABC 对无耦合优化目标进行求解.
  - 1.1  $i = 1$ ;
  - 1.2 repeat;
  - 1.3 对第  $i$  个子系统, 根据式 (9) 随机初始化种群解  $X_j^i$ , 并执行分离算法,  $j \in 1, 2, \dots, SN_1$ ;
  - 1.4 根据式 (5) 计算种群中各个蜜蜂的适应值;
  - 1.5 cycle = 1;
  - 1.6 repeat;
  - 1.7 雇佣蜂根据原 ABC 平移邻域式 (12) 产生新的解, 执行分离算法后根据式 (5) 计算新解适应值;
  - 1.8 雇佣蜂根据贪心策略选择蜜源;
  - 1.9 根据式 (11) 计算选择蜜源  $X_j^i$  的概率  $P_j$ ;
  - 1.10 观察蜂根据概率  $P_j$  选择蜜源  $X_j^i$ , 再根据交换邻域式 (13) 更新蜜源后执行分离算法, 并根据式 (5) 计算新蜜源适应值;
  - 1.11 观察蜂根据贪心策略选择蜜源;
  - 1.12 决定是否是否存在需要放弃的蜜源, 如果存在, 根据式 (9) 随机产生一个蜜源替代它;
  - 1.13 记录最优解;
  - 1.14 cycle = cycle + 1;
  - 1.15 until cycle = MCN<sub>1</sub>;
  - 1.16 记录子系统最优解  $X_{best}^i$ ;

1.17  $i = i + 1$ ;

1.18 until  $i > 4$ .

2 第二阶段在第一阶段得到的系统最优解  $X_{\text{best}} = (X_{\text{best}}^1, X_{\text{best}}^2, X_{\text{best}}^3, X_{\text{best}}^4)$  基础上, 利用 ABC 算法优化各个子系统旋转角度.

2.1 初始化种群的解  $\alpha_j = (\alpha_j^1, \alpha_j^2, \alpha_j^3, \alpha_j^4)$ ,  $j \in 1, 2, \dots, \text{SN}_2$ ,  $\alpha_j^i \in [0, 2\pi)$ ,  $i = 1, 2, 3, 4$ ;

2.2 根据式 (8) 计算种群中各个蜜蜂的适应值;

2.3 cycle = 1;

2.4 repeat;

2.5 雇佣蜂根据式 (10) 产生新的解并计算适应值;

2.6 雇佣蜂根据贪心策略选择蜜源;

2.7 根据式 (11) 计算选择蜜源  $\alpha_j$  的概率  $P_j$ ;

2.8 观察蜂根据概率  $P_j$  选择蜜源  $\alpha_j$ , 再根据式 (10) 在该蜜源附近产生新的蜜源, 并根据式 (8) 计算新蜜源的适应值;

2.9 观察蜂根据贪心策略选择蜜源;

2.10 决定是否是否存在需要放弃的蜜源, 如果存在, 根据式 (9) 随机产生一个蜜源替代它;

2.11 记录最优解;

2.12 cycle = cycle + 1;

2.13 until cycle =  $\text{MCN}_2$ ;

2.14 输出最优结果.

## 6 数值实验及分析

### 6.1 数值实验

本文以简化国际商业通信卫星模块 (INTELSAT-III) 布局为例<sup>[15]</sup>, 测试本文算法的优化性能. 在图 1(a) 所示的卫星舱体内的 4 个承载隔板面上, 需要布置给定的 60 个待布物 (24 个长方体, 36 个圆柱体), 要求待布物之间间隔 10 mm, 各个待布物的几何尺寸、质量和在不同承载面上的分配见文献 [17,18]. 已知  $R = 500.0$  mm,  $R_C = 100.0$  mm,  $H_1 = 300.0$  mm,  $H_2 = 830.0$  mm,  $H_t = 1150.0$  mm,  $D_t = 20.0$  mm, 舱体壁及舱内固定件总质量为  $m^0 = 576.534$  kg, 其质心坐标为  $X_C = (0, 0, 553.56)$  mm, 在参考坐标系  $oxyz$  中, 空舱转动惯量阵  $I_0$  为

$$I_0 = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} = \begin{bmatrix} 352.2 & 0 & 0 \\ 0 & 352.2 & 0 \\ 0 & 0 & 106.82 \end{bmatrix} \text{ kg} \cdot \text{m}^2.$$

全舱的技术要求由式 (1b)~(1d) 给出, 其中干涉量  $V = 0$ ; 全舱质心的期望值  $x_e = 0$ ,  $y_e = 0$ , 允许偏差值  $\delta x_e = \delta y_e = 3$  mm; 舱体不平衡度允许值  $\delta \theta_{x'} = \delta \theta_{y'} = \delta \theta_{z'} = 0.03$  rad; 求满足不干涉约束 (1b), 静稳定性约束 (1c), 平衡度约束 (1d) 的条件下, 全舱转动惯量之和 (1a) 最优的布局方案.

为了对比各个算法的性能, 定义:

- 合法解. 当一个布局解满足不干涉、静稳定性和惯性夹角的要求, 即式 (1b)~(1d), 我们称其为一个合法解.

表 1 本文 ABC, DABC, MS-ABC 和 MS-DABC 的算法性能对比  
 Table 1 The performance comparison of ABC, DABC, MS-ABC and MS-DABC algorithm

	Design objective $f$ (kg·m <sup>2</sup> )				Computational time (s)				SR(%)
	Best	Average	Worst	SD	Best	Average	Worst	SD	
ABC	724.66	730.23	735.67	2.38	54.42	56.76	60.93	1.63	42
DABC	715.87	720.22	724.82	1.92	47.87	50.27	53.09	1.09	72
MS-ABC	716.89	720.10	724.78	1.59	63.57	66.31	70.07	1.69	88
MS-DABC	707.22	710.75	712.66	1.02	56.55	59.73	62.68	1.16	100

• 成功率 ( $\eta^s$ ). 实验中获得合法解的数目  $K^s$  占总的实验数目  $K^a$  (50 次运行) 的百分比, 即  $\eta^s = K^s / K^a \times 100\%$

从两个方面评价算法性能: (1) 求解质量, 即设计的目标函数值 (即转动惯量之和  $f(X) = I_{x'x'} + I_{y'y'} + I_{z'z'}$ ); (2) 计算稳定性 (目标函数值的标准差和成功率  $\eta^s$ ).

## 6.2 多种 ABC 算法的对比

### 6.2.1 实验结果

为了验证本文提出的多阶段优化策略和双邻域 ABC 算法的有效性, 将基本人工蜂群算法 ABC、双邻域人工蜂群算法 DABC、多阶段人工蜂群算法 MS-ABC 以及采用多阶段优化策略的双邻域人工蜂群算法 MS-DABC 进行对比. 以上算法均嵌入分离算法, 运算环境为 i5 3.1 GHz 4 GB RAM, 均采用 C++ 编程. 整个系统全局适应度函数见式 (2), 其中  $\lambda_1 = 1$ ,  $\lambda_2 = 0.001$ ,  $\lambda_3 = 1$ ,  $\lambda_4 = 1$ ,  $w_1 = 1$ ,  $w_2 = 20000$ ,  $w_3 = 5$ ,  $w_4 = 500$ .

在 ABC 和 DABC 中, 算法终止的最大进化数 MEN 以及 limit 均分别为 2000 和 500, 全局评价函数为式 (2) 中的  $F(X)$ . 在 MS-ABC 和 MS-DABC 中第一阶段终止的最大进化数 MEN<sub>1</sub>, 第二阶段终止的最大进化数 MEN<sub>2</sub> 以及 limit 分别为 2000, 1500 和 500. 每个算法对算例独立运行 50 次, 并将获得的整个系统的设计目标 (design objective)、运行时间 (computational time) 的最优值 (best)、平均值 (mean)、最差值 (worst) 和标准差 (standard deviation, SD) 以及成功率 (success rate, SR) 均列于表 1.

为检验实验结果之间差异是否具有显著性, 我们对实验结果进行了 t 检验. 首先利用 spss 软件中单样本 Kolmogorov-Smirnov 检验对各个算法的实验结果作正态分布假设检验, 检验结果显示各个算法实验结果服从正态分布. 接着采用两独立样本的 t 检验来检验任意两个算法得到的目标函数均值是否具有显著性差别, 采用默认的显著性水平  $\alpha = 0.05$ , 具体 t 检验结果见表 2.

### 6.2.2 结果分析

以上算法中, 我们均结合分离算法做局部搜索. 从表 1 可以看出:

(1) 算法的优化性能方面. 相对于 ABC 算法, 加入双邻域策略的 DABC 在转动惯量之和的最优值, 平均值, 最差值上分别提高了  $(724.66 - 715.87) / 724.66 \times 100\% = 1.21\%$ ,  $(730.23 - 720.22) / 730.23 \times 100\% = 1.37\%$ ,  $(735.67 - 724.82) / 735.67 \times 100\% = 1.47\%$ ; 加入多阶段优化策略的 MS-ABC 算法在转动惯量之和的最优值、平均值、最差值上分别提高了  $(724.66 - 716.89) / 724.66 \times 100\% = 1.07\%$ ,  $(730.23 - 720.10) / 730.23 \times 100\% = 1.39\%$ ,  $(735.67 - 724.78) / 735.67 \times 100\% = 1.48\%$ , 结合多阶段优化策略和双邻域策略的 MS-DABC 取得了最好的求解结果, 相对于 ABC 算法, 在转动惯量之和的最优值、平均值、最差值上

**表 2** 算法 ABC, DABC, MS-ABC 和 MS-DABC50 次实验的 t 检验结果  
**Table 2** Independent-samples T test of ABC, DABC, MS-ABC and MS-DABC algorithm

		Levene's test for equality of variances		t-test for equality of means		
		F	Sig.	t	Sig. (2-tailed)	Mean difference
ABC vs. DABC	Equal variances assumed	2.174	0.144	23.156	0.000	10.004362040
	Equal variances not assumed			23.156	0.000	10.004362040
ABC vs. MS-ABC	Equal variances assumed	6.014	0.016	25.042	0.000	10.127931400
	Equal variances not assumed			25.042	0.000	10.127931400
ABC vs. MS-DABC	Equal variances assumed	22.438	0.000	53.064	0.000	19.474699840
	Equal variances not assumed			53.064	0.000	19.474699840
DABC vs. MS-DABC	Equal variances assumed	11.875	0.001	30.760	0.000	9.470337800
	Equal variances not assumed			30.760	0.000	9.470337800
MS-ABC vs. MS-DABC	Equal variances assumed	8.005	0.006	34.907	0.000	9.346768440
	Equal variances not assumed			34.907	0.000	9.346768440

分别提高了  $(724.66-707.22)/724.66 \times 100\% = 2.41\%$ ,  $(730.23-710.75)/730.23 \times 100\% = 2.67\%$ ,  $(735.67-712.66)/735.67 \times 100\% = 3.13\%$ .

(2) 算法稳定性方面. 在目标函数值的标准差上, ABC 为 2.38, DABC 减少到 1.92, MS-ABC 减少到 1.59, MS-DABC 为最小的 1.02, 说明 DABC 与 MS-ABC 的优化结果相对 ABC 更为集中, 而 MS-DABC 的优化结果则具有最好的集中度. 在成功率方面, DABC 与 MS-ABC 分别将成功率从 ABC 的 42% 提高到 72% 和 88%, 而 MS-DABC 则取得了 100% 的成功率. 综上, 不管是在函数值的集中度还是在求解成功率方面, 本文提出的双邻域人工蜂群算法 DABC 和多阶段优化策略 MS-ABC 都较原有的人工蜂群算法有了较大地提高, 而结合这两种策略的 MS-DABC 具有最优的算法稳定性和成功率.

(3) 运算时间方面. 相较于 ABC 算法, DABC 算法运算时间更短, 说明交换邻域搜索相对于原 ABC 邻域搜索用时更少; 而 MS-DABC 用时比 DABC 长, 其原因是 MS-DABC 算法比 DABC 算法多了第二阶段的旋转寻优; MS-ABC 用时最长, 其原因是 ABC 较 DABC 算法用时更长.

表 2 给出了各个算法均值的 t 检验结果, 第 2 列进行的是两方法结果的方差齐性检验, 当 F 检验  $p$  值 (即图中的 sig. 值) 小于给定的显著性水平 (如本文中的 0.05) 时, 认为方差不相同, 从而根据 Equal variances not assumed 时 t 检验的  $p$  值, 来判断两方法结果的均值是否有显著性差异; 否则, 根据 Equal variances assumed 时 t 检验的  $p$  值, 来判断均值差异是否具有显著性; ABC 与 DABC 检验结果中, F 检验 Sig. = 0.144 > 0.05, 表示方差不具有显著性差异, 根据 Equal variances assumed 这一栏中 t 检验 Sig. (2-tailed) = 0.000 < 0.05, 说明 50 次实验结果中, ABC 与 DABC 求解的目标函数均值差异显著; ABC 与 MS-ABC 检验结果中, F 检验 Sig. = 0.016 < 0.05, 表示方差具有显著性差异, 根据 Equal variances not assumed 这一栏中 t 检验 Sig. (2-tailed) = 0.000 < 0.05, 说明 ABC 与 MS-ABC 求解的目标函数均值差异显著. ABC 与 MS-DABC 检验结果中, F 检验 Sig = 0.000 < 0.05, 表示方差具有显著差异, 根据 Equal variances not assumed 这一栏中 t 检验 Sig. (2-tailed) = 0.000 < 0.05, 说明 ABC 与 MS-DABC 的目标函数均值差异显著. DABC 与 MS-DABC 检验结果中, F 检验 Sig. = 0.001 < 0.05, 表示方差具有显著差异, 根据 Equal variances not assumed 这一栏中 t 检验

表 3 QPGP, CCSS, DVGCCGA 和 MS-DABC 的算法性能对比

Table 3 The performance comparison of QPGP, CCSS, DVGCCGA and MS-DABC algorithm

	Design objective $f$ (kg·m <sup>2</sup> )				Computational time (s)				SR(%)
	Best	Average	Worst	SD	Best	Average	Worst	SD	
QPGP	758.33	773.58	791.43	10.58	112.52	900.26	1152.23	105.33	80
CCSS	735.24	735.24	735.24	0.00	245.39	250.55	256.65	3.25	100
DVGCCGA	712.99	718.93	726.59	2.64	110.48	179.86	216.97	13.89	60
MS-DABC	707.22	710.75	712.66	1.02	56.55	59.73	62.68	1.16	100

Sig. (2-tailed) = 0.000 < 0.05, 说明 DABC 与 MS-DABC 求解的目标函数均值差异显著. ABC 与 MS-DABC 检验结果中, F 检验 Sig. = 0.006 < 0.05, 表示方差具有显著差异, 根据 Equal variances not assumed 这一栏中 t 检验 Sig. (2-tailed) = 0.000 < 0.05, 说明 MS-ABC 与 MS-DABC 求解的目标函数均值差异显著.

综上, 由 ABC 与 DABC, 以及 MS-ABC 与 MS-DABC 求解的目标函数均值有显著差别, 说明了双邻域蜂群算法 (DABC) 的有效性; 而 ABC 与 MS-ABC, DABC 与 MS-DABC 求解的目标函数均值有显著差别说明了两阶段优化策略的有效性.

本文提出的多阶段优化策略和双邻域更新策略对于改进 ABC 算法的优化能力均非常有效. 其中, 双邻域更新策略能有效避免 ABC 算法单邻域搜索容易陷入局部最优的问题, 提高了算法的搜索能力. 而多阶段的优化策略将优化目标进行分解处理, 使得原本搜索空间大, 具有不连续、多模态的解空间的问题分解成两个相对好解的问题, 有效地简化了计算. 尽管直接利用 ABC 算法在原解空间搜索, 其有机会搜索到更好的解, 但是由于同时考虑所有的变量和优化目标, 解空间不仅非常大, 而且具有不连续、多模态的性质, 存在大量局部最优解, 甚至是不可行的局部最优解, 可行解在解空间的比例过低, 这些都导致其很难搜索到高质量的可行解. 而多阶段优化策略尽管缩小了搜索空间, 但其在每个阶段解空间性质相对较好, 故其反而能求得较好的解.

### 6.3 MS-DABC 与已有算法的对比

#### 6.3.1 实验结果

将本文提出的 MS-DABC 算法与已有的 QPGP<sup>[17]</sup>, CCSS<sup>[18]</sup> 和 DVGCCGA<sup>[19]</sup> 3 种算法作对比. 其中 QPGP 算法在 CPU 为 Intel Celeron 1.2 GHz, 内存为 512 MB 的台式电脑上运行, 其编程语言没有给出; CCSS 算法在 CPU 为 1.2 GHz, 内存为 512 MB 的台式电脑上运行, 采用 C++ 语言编程; DVGCCGA 算法在 CPU 为 AMD 3600+, 内存为 1 GB 的台式电脑上运行, 编程语言也没有给出.

表 3 给出了 MS-DABC 与 QPGP, CCSS, DVGCCGA 的实验对比数据, 每一个算法均运行 50 次. 表 4 给出了 MS-DABC 与 QPGP, CCSS, DVGCCGA 的 t 检验结果. 表 5 给出了 4 个算法最优解的性能指标, 其中,  $I$  和  $\theta$  分别表示系统对星体坐标系  $o'x'y'z'$  在 3 个坐标轴方向上的转动惯量和惯性夹角,  $CP$  表示系统质心坐标,  $V$  表示系统干涉量,  $T$  表示算法时间. 图 2 给出了 MS-DABC 获得的卫星舱在 4 个承载板上的最好布局示意图, 其中阴影部分为中心立柱.

#### 6.3.2 结果对比与机理分析

从表 3 的实验对比结果来看, 有如下分析.

表 4 算法 ABC, DABC, MS-ABC 和 MS-DABC 50 次实验的 t 检验结果  
Table 4 Independent-samples T test of ABC, DABC, MS-ABC and MS-DABC algorithm

		Levene's test for equality of variances		t-test for equality of means		
		F	Sig.	t	Sig.(2-tailed)	Mean difference
QPGP vs.	Equal variances assumed	107.590	0.000	41.800	0.000	62.83
MS-DABC	Equal variances not assumed			41.800	0.000	62.83
CCSS vs.	Equal variances assumed	0	0.000	169.775	0.000	24.49
MS-DABC	Equal variances not assumed			169.775	0.000	24.49
DVGCCGA vs.	Equal variances assumed	6.699	0.000	20.437	0.000	8.18
MS-DABC	Equal variances not assumed			20.437	0.000	8.18

表 5 QPGP, CCSS, DVGCCGA 和 MS-DABC 的算法性能对比  
Table 5 The performance comparison of QPGP, CCSS, DVGCCGA and MS-DABC algorithm

Merits	QPGP			CCSS			DVGCCGA			MS-DABC		
	x axis	y axis	z axis	x axis	y axis	z axis	x axis	y axis	z axis	x axis	y axis	z axis
I (kgm <sup>2</sup> )	280.91	285.01	181.32	271.44	282.44	193.35	307.959	312.334	210.275	268.372	271.509	167.347
$\theta$ (rad)	0.0004	0.0000	-0.006	-0.002	-0.002	0.000	0.001	0.000	0.001	0.00004	0.000428	0.000001
CP (mm)	-0.002	-0.002	553.24	0.9498	2.7795	553.24	-0.027	-0.021	—	-0.1581	0.009	553.24
V (mm <sup>3</sup> )		0.0			0.0			0.0			0.0	
T (s)		492.56			244.39			345.5			57.86	

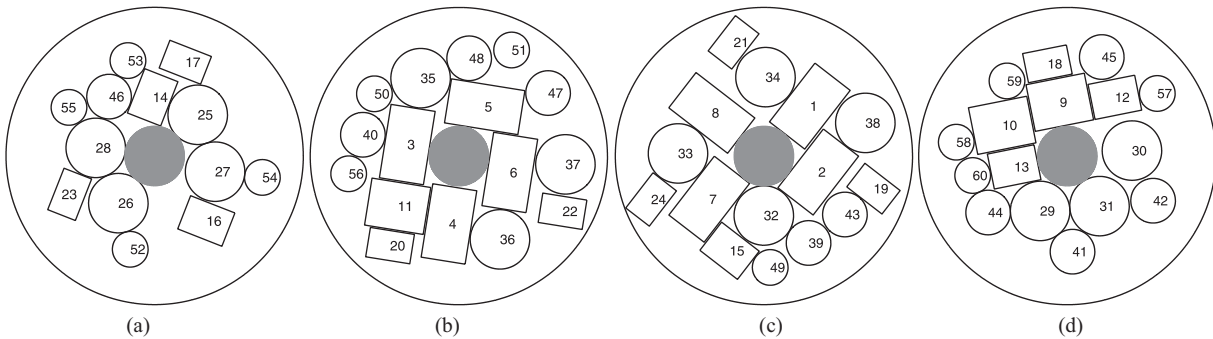


图 2 MS-DABC 算法求解卫星模块的最佳布局示意图

Figure 2 Best layout schematic diagram of satellite module obtained by MS-DABC. (a) Components layout on S<sub>1</sub>; (b) components layout on S<sub>2</sub>; (c) components layout on S<sub>3</sub>; (d) components layout on S<sub>4</sub>

(1) 在算法的优化性能方面. 本文的 MS-DABC 在各个承载板上布局物的分配与 QPGP 和 CCSS 一样, 求得的转动惯量的最优值、平均值、最差值均优于 QPGP 和 CCSS. DVGCCGA 对此卫星舱布局问题的计算结果是当今国际上的最好结果, 相较于 DVGCCGA, MS-DABC 在转动惯量之和的最优值、平均值、最差值上分别提高了  $(712.99-707.22)/712.99 \times 100\% = 0.81\%$ ,  $(718.93-710.75)/718.93 \times 100\% = 1.14\%$ ,  $(726.59-712.66)/726.59 \times 100\% = 1.92\%$ , 刷新了该算例的求解记录.

(2) 在算法稳定性方面. 在目标函数值的标准差上, MS-DABC 为 1.02, 均优于 QPGP 和 DVGC-



CGA, 但不如 CCSS; 在求解成功率上, MS-DABC 与 CCSS 一样, 均为 100%, 远好于 QPGP 的 80% 和 DVGCCGA 的 60%. 故在算法鲁棒性上, MS-DABC 均好于 QPGP 和 DVGCCGA, 稍差于 CCSS.

(3) 在运算时间方面. 由于各个算法的实验环境和编程语言不尽相同, 因此不便作详细地比较.

表 4 的  $t$  检验结果也表明, 本文 MS-DABC 算法在求解目标函数的均值上较 QPGP, CCSS 和 DVGCCGA 有显著性的差别.

总的说来, 相对于已有的算法, 本文提出的 MS-DABC 具有更好的优化能力和计算稳定性.

3DSMLOP 是一个多约束、非线性的耦合问题. 多约束耦合问题可以转换为 (或者说本质上) 一个多目标优化问题, 其没有全局最优解, 只有 pareto 最优解集. 我们求解的工程满意解就在 pareto 解集当中, 其是可行域与 pareto 前沿的接合点, 故有效的优化算法必须在 Pareto 前沿附近具备良好的优化性能.

QPGP, CCSS 和 DVGCCGA 均是由智能算法与启发式搜索方法混合得到的混合算法, 其评价函数是利用罚函数法将各个子目标和约束聚合成单目标评价函数  $F(X)$ , 该函数包含了问题的所有约束条件和目标函数. 其优点是继承了单目标优化的一些成熟算法机理, 但是其也有着致命缺点: (1) 对 pareto 前沿形状非常敏感, 很难在非凸均衡面上找到所有的 pareto 最优解. (2) 单一的全局评价函数  $F(X)$  包含了问题的所有约束条件和目标函数, 而我们研究的问题其解空间非常大, 可行解的比例占整个解空间的比例过低, 并且含有大量局部最优解, 甚至大量不可行局部最优解, 同时考虑所有的约束条件, 很容易陷入不可行的局部最优解, 而无法跳出, 从而使得算法在有限的时间内求解效率和解的质量都不高.

为了解决上述问题, 本文提出多阶段双邻域蜂群算法 MS-DABC. 多阶段优化策略通过对约束进行分步处理, 先优化无耦合的优化目标, 使系统快速搜索到系统解的 pareto 前沿, 当然, 这里的解也许为不可行解, 其不一定满足耦合约束. 然后通过各个子系统的协同旋转来消除耦合约束, 这样使得系统集中沿着 pareto 前沿附近搜索, 直到搜索到可行域与 pareto 前沿的接合点. 尽管多阶段优化策略一定程度缩小了搜索空间, 但其却能较快速地搜索到较优解. 实验证明, 分阶段优化是处理复杂耦合问题的一个有效方法. 此外, 我们提出了具有较强全局搜索能力的 DABC. DABC 在原 ABC 基础上加入一个新的邻域搜索——交换邻域. 原 ABC 邻域能对布局物附近进行平移搜索, 但是当进化到一定程度, 这种随机平移搜索大多数是无用的搜索, 当陷入局部最优时, 在当前的布局拓扑结构下, 改变单个布局物坐标无法得到更好的解, 因此其搜索效率不高. 交换邻域搜索能快速的改变当前布局拓扑结构, 从而使得算法能跳出局部最优, 搜索到更好的拓扑结构, 但是其也有不足——无法在布局物附近进行搜索. 为了有效利用两种邻域的优势, DABC 中雇佣蜂采用原 ABC 的邻域搜索, 而观察蜂则采用交换邻域搜索. 故 DABC 既能在布局物附近进行邻域搜索, 也能利用交换邻域跳出局部最优, 搜索到更好的拓扑结构. 实验也验证 DABC 的优化性能优于 ABC.

## 7 结论

3DSMLOP 是一个带多约束的复杂布局问题, 难于求解. 传统的做法利用罚函数将 3DSMLOP 转化为单目标优化问题, 然后开发具有较强的全局搜索能力的智能算法或者改进智能算法直接进行优化, 但鉴于这类复杂布局问题的解空间往往很大, 且具有非连续、多模态的性质, 直接优化很难得到较好的解. 本文首先针对直接优化困难的问题, 提出了多阶段优化策略. 相对于单阶段的直接优化策略, 多阶段优化策略虽然缩小了搜索的解空间, 但算法在各个阶段解空间的性质较好, 更容易求解较好的解. 此外, 针对基本 ABC 算法单一搜索邻域的不足, 设计了双邻域搜索人工蜂群算法 DABC. DABC 算法

中雇佣蜂和观察蜂采用不同的邻域搜索新蜜源,能有效避免陷入单一邻域的局部最优,提高了搜索效率.实验验证,对于具有耦合的复杂布局优化问题,利用针对耦合约束的多阶段处理策略和设计具有较强全局搜索能力的改进智能算法是两个非常有效的求解途径.

**致谢** 感谢吕志鹏教授对本文提出的修改建议.

## 参考文献

- 1 Teng H F, Sun S L, Ge W H, et al. Layout optimization for dishes installed on a rotating circular table—the packing problem with equilibrium behavioral constraints. *Sci Sin Math*, 1994, 24: 754–760 [滕弘飞, 孙守林, 葛文海, 等. 转动圆桌平衡摆盘——带平衡性能约束的 Packing 问题. *中国科学: 数学*, 1994, 24: 754–760]
- 2 Xiao R B, Xu Y C, Amos M. Two hybrid compaction algorithms for the layout optimization problem. *BioSystems*, 2007, 90: 560–567
- 3 Wang Y S, Shi Y J, Teng H F. An improved scatter search for circles packing problem with the equilibrium constraint. *Chin J Comput*, 2009, 32: 1214–1221 [王奕首, 史彦军, 滕弘飞. 用改进的散射搜索法求解带平衡约束的圆形 Packing 问题. *计算机学报*, 2009, 32: 1214–1221]
- 4 Liu J F, Li G. Basin filling algorithm for the circular packing problem with equilibrium behavioral constraints. *Sci Sin Inform*, 2010, 53: 885–895 [刘景发, 李刚. 求解带平衡性能约束的圆形装填问题的吸引盘填充算法. *中国科学: 信息科学*, 2010, 40: 423–432]
- 5 Li G, Liu J F. Heuristic algorithm based on tabu search for the circular packing problem with equilibrium constraints. *Sci Sin Inform*, 2011, 41: 1076–1088 [李刚, 刘景发. 基于禁忌搜索的启发式算法求解带平衡约束的圆形装填问题. *中国科学: 信息科学*, 2011, 41: 1076–1088]
- 6 Li Z Q, Tian Z J, Wang Y S, et al. A fast heuristic parallel ant colony algorithm for circles packing problem with the equilibrium constraints. *J Comput Res Dev*, 2012, 49: 1899–1909 [黎自强, 田茁君, 王奕首, 等. 求解平衡约束圆形 packing 问题的快速启发式并行蚁群算法. *计算机研究与发展*, 2012, 49: 1899–1909]
- 7 Huang Z D, Xiao R B. An emergent computation approach to the problem of polygon layout with performance constraints. *Phys A*, 2013, 392: 5074–5088
- 8 He K, Mo D Z, Xu R C, et al. A quasi-physical algorithm based on coarse and fine adjustment for solving circles packing problem with constraints of equilibrium. *Chin J Comput*, 2013, 36: 1224–1234 [何琨, 莫旦增, 许如初, 等. 基于粗精调技术的求解带平衡约束圆形 packing 问题的拟物算法. *计算机学报*, 2013, 36: 1224–1234]
- 9 Huang W Q, He K. A pure quasi-human algorithm for solving cuboid packing problem. *Sci China Ser F-Inf Sci*, 2009, 39: 617–622 [黄文奇, 何琨. 求解长方体 packing 问题的纯粹拟人算法. *中国科学 F 辑: 信息科学*, 2009, 39: 617–622]
- 10 Zhang D F, Peng Y, Zhu W X, et al. A hybrid simulated annealing algorithm for the three-dimensional packing problem. *Chin J Comput*, 2009, 32: 2147–2156 [张德富, 彭煜, 朱文兴, 等. 求解三维装箱问题的混合模拟退火算法. *计算机学报*, 2009, 32: 2147–2156]
- 11 Li W, Huang W Q, Jiang D C, et al. A heuristic algorithm for cube packing with time schedule. *Sci Sin Inform*, 2010, 53: 18–29 [李未, 黄文奇, 蒋东辰, 等. 一种求解带有时间调度的四维长方体装填问题的启发式算法. *中国科学: 信息科学*, 2010, 40: 1–12]
- 12 Szykman S, Cagan J. Constrained three-dimensional component layout using simulated annealing. *J Mech Design*, 1997, 119: 28–35
- 13 Braun R D, Moore A A, Kroo I M. Collaborative approach to launch vehicle design. *J Spacecraft Rockets*, 1997, 34: 478–486
- 14 Grignon P M, Fadel G M. A GA based configuration design optimization method. *J Mech Design*, 2004, 126: 6–14
- 15 de Sousa F L, Muraoka I, Galski R L. On the optimal positioning of electronic equipment in space platforms. In: *Proceedings of the 19th International Congress of Mechanical Engineering*, Brasilia, 2007. 5–9
- 16 Sun Z G, Teng H F. Optimal layout design of a satellite module. *Eng Optimiz*, 2003, 35: 513–529
- 17 Zhang B, Teng H F, Shi Y J. Layout optimization of satellite module using soft computing techniques. *Appl Soft Comput*, 2008, 8: 507–521
- 18 Wang Y S, Teng H F, Shi Y J. Cooperative coevolutionary scatter search for satellite module layout design. *Eng*

- Computations, 2009, 26: 761–785
- 19 Teng H, Chen Y, Zeng W, et al. A dual-system variable-grain cooperative coevolutionary algorithm: satellite-module layout design. *IEEE T Evolut Comput*, 2010, 14: 438–455
  - 20 Karaboga D. An Idea Based on Honey Bee Swarm for Numerical Optimization. Kayseri: Erciyes University. Technical Report TR06. 2005
  - 21 Karaboga D, Basturk B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *J Global optim*, 2007, 39: 459–471
  - 22 Li J Q, Pan Q K, Gao K Z. Pareto-based discrete artificial bee colony algorithm for multi-objective flexible job shop scheduling problems. *Int J Adv Manuf Tech*, 2011, 55: 1159–1169
  - 23 Karaboga D, Akay B. A comparative study of artificial bee colony algorithm. *Appl Math Comput*, 2009, 214: 108–132

## Multi-stage dual neighborhood artificial bee colony algorithm for satellite module layout optimization problem

Zhendong HUANG<sup>1,2</sup> & Renbin XIAO<sup>1\*</sup>

1 *School of Automation, Huazhong University of Science and Technology, Wuhan 430074, China;*

2 *School of Statistics, Hubei University of Economics, Wuhan 430205, China*

\*E-mail: rbxiao@hust.edu.cn

**Abstract** This paper presents an improved artificial bee colony algorithm with dual neighborhood, and combines it with a multi-stage solving strategy to form a multi-stage dual neighborhood artificial bee colony algorithm (MS-DABC) for the three-dimensional satellite module layout optimization problem (3DSMLOP). 3DSMLOP is a complex multi-constrained coupling problem whose solution space is non-continuous, non-linear, multi-modal. MS-DABC decomposes 3DSMLOP into several subsystems, and divides the optimization process into two stages according to the coupling relationship on optimization goals between subsystems. In the first stage, each subsystem uses the dual neighborhood artificial bee colony algorithm (DABC) to optimize no-coupling optimization goals independently. Based on the optimal solution obtained in the first stage, the coupling optimization goals are solved by the collaborative rotation between the subsystems in the second stage. The rotation angles are optimized by the artificial bee colony algorithm. Numerical experiment results show that MS-DABC is very effective and has outstanding performance for solving the 3DSMLOP example.

**Keywords** satellite module layout problem, artificial bee colony algorithm, coupling problem, multi-stage, dual neighborhood



**Zhendong HUANG** was born in 1980. He received his Ph.D. degree in Control Science and Engineering from Huazhong University of Science and Technology in 2014. He is now a lecturer at the School of Statistics, Hubei University of Economics, Wuhan, China. His research interests focus on layout optimization, computational intelligence, and emergent computation.



**Renbin XIAO** was born in 1965. He received his B.S. degree in Ship Engineering, M.S. degree in Ship Hydrodynamics, and Ph.D. degree in Systems Engineering from Huazhong University of Science and Technology in 1986, 1989, and 1993, respectively. Currently, he is a professor at the School of Automation, Huazhong University of Science and Technology, Wuhan, China. His research interests include swarm intelligence and emergent computation, management decision theory and decision support system, and creative design of complex products.