

# 基于果蝇优化算法的多工位装配序列规划

袁文兵 常亮 徐周波 古天龙

(桂林电子科技大学广西可信软件重点实验室 桂林 541004)

**摘要** 为同时解决产品装配序列规划和多工位分配问题,提出一种面向复杂产品的基于果蝇优化算法的多工位装配序列规划方法。首先,基于果蝇优化算法设计了针对求解序列的编码体系;其次,采用多子种群并行搜索模式,重新设计了果蝇优化算法的搜索过程;然后,为了综合考虑多工位上相关装配操作成本的影响,提出了新的适应度函数表达式,并将适应度函数与优先序列矩阵结合起来对进化过程进行引导,实现了对产品装配序列和工位分配顺序的优化;最后,以飞机起落架为例,验证了所提方法在解决多目标优化问题方面的有效性。

**关键词** 果蝇优化算法,多工位,装配序列规划,多目标优化

**中图分类号** TP391 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.04.052

## Multi-plant Assembly Sequence Planning Based on Fruit Fly Optimization Algorithm

YUAN Wen-bing CHANG Liang XU Zhou-bo GU Tian-long

(Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, China)

**Abstract** An improved fruit fly optimization algorithm(FOA) was proposed to solve the distribution problem of assembly sequence planning and plant assignment. Firstly, in the multi-plant assembly sequence planning model, the coding system of FOA is given. Secondly, it uses multiple fruit fly groups to enhance the parallel search ability of the FOA. According to the characteristics of the improved FOA, a smell-based search and a vision-based search are well designed for the groups to stress its exploitation. And a cooperative search process is developed to stress its exploration. Thirdly, the fitness function is proposed with comprehensive consideration of assembly operation cost, assembly tool change cost, the clamping change cost and general transportation cost. Moreover, the feasible assembly sequences and station allocation are guided by assembly precedence matrix (APM) and fitness function. At same time, they are optimized based on the improved FOA. Finally, an example of the product is tested and illustrated. The test results show that the presented method is feasible and efficient for solving the multi-plant assembly sequence planning problem.

**Keywords** Fruit fly optimization algorithm, Multi-plant optimization, Assembly sequence planning, Multi-objective optimization

## 1 前言

装配序列规划(Assembly Sequence Planning, ASP)是产品制造过程中的重要环节,装配质量直接影响着产品的性能。随着市场竞争的压力不断加剧,人们希望产品不仅要满足性能需求,而且其周期和成本在同行中能占据绝对的优势。相关研究表明,良好的装配序列可以减少20%~40%的制造费用,同时能将生产效率提高100%~200%<sup>[1]</sup>。因此,装配序列规划在产品制造过程中占有非常重要的地位。

针对装配序列规划问题,中外学者进行了大量的研究。这一问题的解决大致可以分为3个步骤:1)建立装配模型;

2)构建装配序列产生的算法;3)优化产生的装配序列,从而得到最优或次优的装配序列。从方法思想来看,算法主要分为间接法和直接法。间接法基于“可拆即可装”的假设,先求出产品的拆卸序列,再逆序即可得到产品的装配序列;直接法是先任意挑选一个零件作为初始装配件或选出基部件,再对剩余零部件进行路径规划,选出所有可行的装配路径实行装配,直到所有零件全部装完为止。

传统的获得产品装配序列的方法主要有割集法<sup>[2]</sup>、知识推理法<sup>[3]</sup>、几何推理法<sup>[4]</sup>等。这些方法虽然很直接,但随着产品零件数目的增加,常常会出现组合爆炸的问题<sup>[5]</sup>;并且需要的人为干预也较多,这对非专业人员存在着较大的难度。因

到稿日期:2016-03-24 返修日期:2016-06-19 本文受国家自然科学基金(61572146, 61363030, U1501252),广西自然科学基金(2015GXNSFAA139285, 2014GXNSFAA118354),广西可信软件重点实验室,广西高等学校高水平创新团队及卓越学者计划,桂林电子科技大学研究生教育创新计划(YJCXS201509)资助。

袁文兵(1989—),男,硕士,主要研究领域为装配序列规划、智能算法;常亮(1980—),男,博士,教授,主要研究领域为知识表示与推理、智能规划、形式化方法, E-mail: changl@guet.edu.cn;徐周波(1976—),女,博士,副教授,主要研究领域为符号计算、智能规划、约束满足问题求解;古天龙(1964—),男,博士,教授,博士生导师,主要研究领域为形式化方法、符号计算。

此,学者们认识到解决装配序列问题存在着两大难题:1)如何解决产品零件规模较大时的组合爆炸问题;2)如何解决序列产生的自动化问题。沿着这一思路,近 20 年来,大量学者围绕智能算法展开了深入的研究,非常巧妙地将它们应用到装配序列规划问题的求解上,并取得了比较好的效果。其中较有代表性的算法有遗传算法<sup>[6]</sup>、萤火虫算法<sup>[7]</sup>、免疫算法<sup>[8]</sup>、蚁群算法<sup>[9]</sup>、粒子群算法<sup>[10]</sup>等。大量实例证明,智能算法在一定程度上较好地解决了零件数目较多的装配序列规划问题。但现有的研究主要是应用智能算法解决单工位上的产品装配规划问题,而对现实装配中常见的多工位装配成本优化问题研究得较少;此外,这些智能算法中需要考虑的参数较多,参数之间的影响也较大,从而影响了问题求解的质量。

果蝇优化算法(Fruit Fly Optimization Algorithm,FOA)作为一种相对较新的智能算法,最早由台湾的潘文超<sup>[11]</sup>博士于 2011 年首次提出,该算法的产生是受到果蝇觅食行为的启发。在经典的果蝇优化算法中,种群中的果蝇基于气味搜索(smell-based search)和视觉搜索(vision-based search)<sup>[11-16]</sup>都可以在有限的时间内到达优化的搜索空间。基于果蝇特有的敏锐视觉和味觉,以及果蝇算法参数少(只有 3 个)且易完成的特点,许多学者成功地将其应用于财务危机预测<sup>[11]</sup>、比例积分微分控制器的调谐<sup>[12]</sup>、多维背包问题<sup>[13]</sup>、网络拍卖物流服务<sup>[14]</sup>、电子节气门的分数阶模糊 PID 控制器<sup>[15]</sup>、钢铁铸造任务的安排<sup>[16]</sup>、对电子商务零售商供应网络的库存问题的优化<sup>[17]</sup>、对 FOA 算法进行改进以用于连续空间的函数优化问题<sup>[18-19]</sup>以及动态蛋白复合物的鉴定<sup>[20]</sup>,均取得了优于其他智能算法的结果。此外,果蝇算法还可以植入许多启发或操作算子,实现并行搜索。

鉴于果蝇算法的优点,本文第一次将果蝇优化算法和多工位相结合,给出基于果蝇优化算法的多工位装配序列规划方法。在算法中,首先采用双向量的方式来编码每一只果蝇,分别用  $v_1, v_2$  表示产品的装配序列和工位分配序列。然后,为找到更好的序列,在全面考虑评价指标的基础上构造适应度函数,通过适应度函数来筛选出最优序列。针对果蝇算法同其他智能算法一样存在着早熟而陷入局部最优的这一缺陷,我们采用了多子种群并行搜索模式和种群间的学习机制。最后,利用果蝇敏锐的味觉、视觉和群体合作的特点,很好地设计了算法的搜索三部曲:基于气味搜索、基于视觉搜索和协同搜索。其中,针对协同搜索特意引入改进优先交叉算子(the Improved Precedence Operation crossover, IPOX)和多点保留交叉(Multi-point Preservative Crossover, MPX)算子,以促进种群进化。本文的方法兼顾了果蝇的局部和全局搜索能力,确保果蝇能够高效地找到全局最优解,从而得到最优的产品装配序列和工位分配序列。

## 2 果蝇优化算法的多工位装配规划模型

本节首先对经典的果蝇算法进行改进,得到解决多工位装配序列规划问题的算法框架。对果蝇优化算法的改进主要体现在以下几个方面:算法的编码方案,评价体系的构造,种群初始化和搜索过程的设计。其中搜索过程又包括基于气味搜索、基于视觉搜索以及协同搜索。

### 2.1 改进的 FOA 框架

FOA<sup>[11-16]</sup>是一种基于果蝇觅食行为的进化算法,它有两个独特的特征:基于气味搜索,通过气味对实物源进行定位并向该方向靠近;基于视觉搜索,通过敏锐视觉锁定食物源。传统果蝇算法的基本流程包括:

- 1)对果蝇的位置进行随机初始化;
- 2)对每只果蝇执行步骤 3)和步骤 4);
- 3)果蝇以任意方向和步长改变位置;
- 4)计算果蝇与可能食物源的距离,用距离的倒数作为气味浓度;
- 5)评估所有果蝇的适应度值,选出最优个体作为当前最优个体并记录其位置;
- 6)引导所有果蝇个体飞向最优位置;
- 7)若满足结束条件,算法停止;否则执行步骤 2)。

本文对传统的果蝇算法进行改进,采用多子种群并行搜索方式,重新设计算法的搜索过程。将改进的果蝇算法与多工位装配序列规划问题进行融合,其算法框架如图 1 所示。

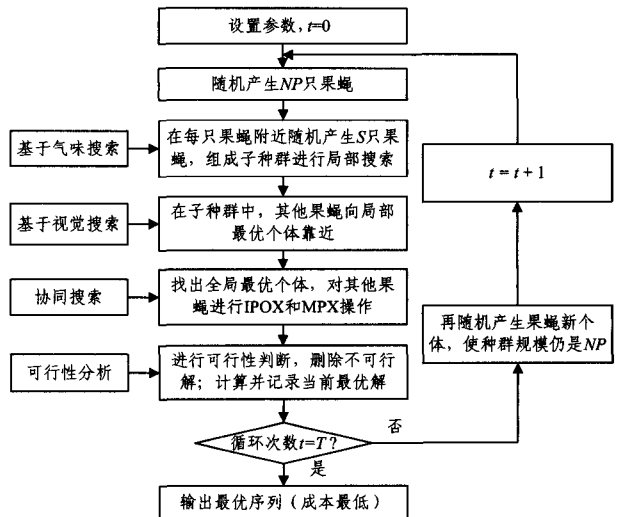


图 1 果蝇优化算法流程图

### 2.2 算法的编码方案

让每一只果蝇代表一个解决方案,即为装配序列  $v_1$  和  $v_2$  工位分配序列,这两个序列采用双向量编码方式分别用向量  $v_1$  和  $v_2$  表示。装配序列向量  $v_1$  表示零件的装配顺序,它的长度和零件的数目在数值上相等,并且零件号所在向量上的位置序号代表着对应零件的装配次序。工位分配向量  $v_2$  表示零件装配过程中所对应的装配工位,它的长度在数值上也与零件数目相等。

解码过程就是求解装配顺序和工位分配的过程,解的优劣还需要通过适应度函数、几何可行性以及工位表来综合评判。为了更好地说明算法的编码和解码过程(见图 2),用飞机起落架<sup>[10]</sup>(见图 3)作为案例进行说明。图 3 中,5 号零件最先装配并且在 1 号工位上装配,2 号零件在第 4 个进行装配并且在 2 号工位上装配,其他零件依次类推。

装配序列向量 $v_1$	5	3	6	2	7	4	1	9	0	10	8
装配序列向量 $v_2$	1	3	1	2	4	3	2	4	1	2	5

图 2 FOA 的编码方式

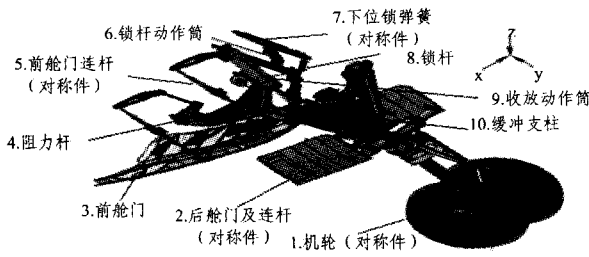


图3 飞机起落架的主要零部件

2.3 算法适应度函数的构造

在多工位的情况下,本文将产品最小装配成本作为适应度函数,主要包括两个方面:零件的装配操作成本和将零件分配到工位上的工位操作成本。其中,装配操作成本又包括更换工具成本、装配操作成本以及装夹变更成本;工位操作成本为零件分配到工位的运输成本。

(1)更换工具成本(ATC):装配零件时,如果连续的两个装配操作不同,就需要更换合适的装配工具。其成本函数如下:

$$ATC = W_{TC} \times \sum_{i=1}^n T_{cn}(T_{i-1}, T_i) \quad (1)$$

$$T_{cn}(T_{i-1}, T_i) = \begin{cases} 1, & \text{if } T_{i-1} \neq T_i \\ 0, & \text{if } T_{i-1} = T_i \end{cases} \quad (2)$$

其中,  $T_i$  为零件  $i$  的装配操作工具,  $W_{TC}$  为工具更换一次的单位成本,  $T_{cn}$  为零件  $i-1$  与  $i$  之间的工具更换,  $ATC$  为工具更换总成本。

(2)装配操作成本(AOC):为保证效益,零件在装配过程中必须要有合适的装配操作成本。其成本函数如下:

$$AOC = \sum_{i=1}^n \sum_{j=1}^m a(i, j) \times ACC(i, j) \quad (3)$$

其中,  $n$  为零件的数目,  $m$  为工位的数量,  $i(i=1, \dots, n)$  为零件,  $j(j=1, \dots, m)$  为工位,  $a(i, j)$  为零件  $i$  是否分配到工位  $j$  上,  $ACC(i, j)$  为零件  $i$  在工位  $j$  上的装配成本,  $ACO$  为装配操作总成本。

(3)装配装夹变更成本(ASC):在进行装配操作时,零件需要专用的夹具来进行固定。如果连续装配的两个部件在相同工位装配方向上发生了变化或需要在不同工位上装配,就必须更换夹具,这就增加了装夹成本。其成本函数如下:

$$ASC = W_{SC} \times \sum_{i=1}^n D_{cn}(D_{i-1}, D_i) \quad (4)$$

$$D_{cn}(D_{i-1}, D_i) = \begin{cases} 1, & \text{if } D_{i-1} \neq D_i \\ 0, & \text{if } D_{i-1} = D_i \end{cases} \quad (5)$$

其中,  $D_i$  为零件  $i$  的装配方向,  $W_{SC}$  为装配夹具变换一次的成本,  $D_{cn}$  为部件  $i-1$  与部件  $i$  之间装配方向的改变,  $ASC$  为装配装夹的总成本。

(4)运输成本(GTC):为完成整个产品的装配,把零件或子装配体运输到对应的工位上产生的成本。其成本函数如下:

$$GTC = \sum_{i=1}^n F_{cn}(F_{i-1}, F_i) \times M_o(F_{i-1}, F_i) \quad (6)$$

$$F_{cn}(F_{i-1}, F_i) = \begin{cases} 1, & \text{if } F_{i-1} \neq F_i \\ 0, & \text{if } F_{i-1} = F_i \end{cases} \quad (7)$$

其中,  $F_{cn}$  用于表示零件  $i-1$  与零件  $i$  是否在同一个工位上装配,  $M_o$  为从工位  $F_{i-1}$  转移到  $F_i$  的单位成本,  $GTC$  为运输总

成本。

装配规划的总成本(TC)为所有操作成本的总和,其等式形式化如下:

$$TC = AOC + ATC + ASC + GTC \quad (8)$$

适应度函数是用来引导果蝇飞向好的食物源(即对应的解),一个好的求解方案也是由适应度函数来评价的。文中选用  $TC$  作为果蝇优化算法的适应度函数,目的是通过求解出最小的装配总成本来得到最优的装配序列和工位分配。因此,果蝇的适应度值越小,认为其越优。目标函数如下:

$$\min \text{Fitness} = TC \quad (9)$$

2.4 种群初始化

2.4.1 初始序列的生成

为保证算法中果蝇个体的多样性,初始装配序列采取随机生成的方式;而与每个装配序列相对应的工位序列,则需要根据工位能力表的引导来产生。现有的装配规划方法基本都是考虑在单工位上进行装配。为接近工程实际,本文采用多工位完成装配规划操作,这就必须综合考虑每个工位的装配能力。表 1<sup>[10]</sup> 为飞机起落架的工位能力表,其中  $c_i(i=1, 2, \dots, n)$  表示零件,  $F_j(j=1, 2, \dots, m)$  表示工位。当  $t_{ij}=1$  时,表示零件  $c_i$  可以在工位  $F_j$  上装配;否则  $t_{ij}=0$ ,这说明零件  $c_i$  不能在工位  $F_j$  上进行装配。这样,每一个装配序列都有一个工位序列与之相对应。如此,种群中的每只果蝇个体就完成了自身的初始化。

表1 工位能力表(PCT)<sup>[10]</sup>

工位	零件									
	$c_1$	$c_2$	$c_3$	$c_4$	$c_5$	$c_6$	$c_7$	$c_8$	$c_9$	$c_{10}$
$F_1$	1	1	1	1	1	1	1	1	1	1
$F_2$	1	0	0	1	1	1	1	1	1	1
$F_3$	0	1	1	0	0	0	0	0	0	0
$F_4$	0	0	0	1	1	1	0	0	1	0
$F_5$	0	1	1	0	0	0	1	1	0	0

2.4.2 初始子种群的划分

有别于经典的果蝇优化算法(FOA),本文采用多子种群来加强算法的并行搜索能力。在算法中把初始种群(含  $NP$  只果蝇)中的每一只果蝇作为子种群的初始个体,因此初始种群被划分成了  $NP$  个子种群。

2.5 设计气味搜索和视觉搜索过程

子种群中的果蝇利用装配优先关系矩阵的引导,产生  $S$  只果蝇组成新的子种群,完成气味搜索过程;再利用适应度函数对子种群中的每只果蝇进行评估,选出局部最优个体(适应度值最小),完成视觉搜索过程。

2.5.1 优先关系矩阵的产生

装配优先序列图(Assembly Precedence Graph, APG)是一个有向图,它包括节点和有向弧。APG 的初衷是用几何数据和空间关系来表示零部件之间的关系,本文用它来表示零件之间的优先装配操作顺序。

用有向图  $G=(C, A)$  来表示 APG,其中  $C=\{c_1, c_2, \dots, c_n\}$  为产品集合,  $c_i$  表示零件节点,  $i=1, 2, \dots, n, n$  为实际产品包含的零件数目;  $A=\{A_1, A_2, \dots, A_m\}$  ( $m$  为操作弧的个数)为操作弧集合,用来表示零件节点之间的优先操作关系。在 APG 中,有向弧的起点零件的优先级比终点零件的优先级高,即节点  $c_i$  作为优先关系图的起点必须在作为优先关系图

的终点  $c_j$  之前进行装配。因此,一个可行序列可以从一个基部件进行有向弧操作得到。飞机起落架的连接关系和操作关系如图 4 所示,APG 如图 5 所示。

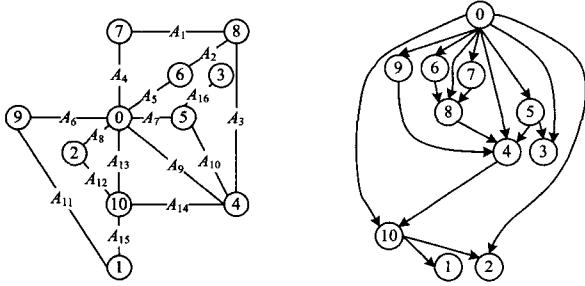


图 4 飞机起落架零件的连接图 图 5 飞机起落架的装配优先图

为方便果蝇算法的执行,将装配优先序列图转化为优先关系矩阵(Assembly Precedence Matrix, APM)。用  $b_{ij} (i \neq j)$  表示  $c_i$  和  $c_j$  之间的优先关系; $b_{ij} = 1$ ,表示  $c_i$  必须在  $c_j$  之前装配;否则  $b_{ij} = 0$ ,表示  $c_i$  和  $c_j$  之间没有优先关系。另,规定当  $i = j$  时,  $b_{ij} = 0$ 。这样,就可以得到飞机起落架装配优先序列图对应的优先关系矩阵。例如零件沿有向弧和零件  $c_{10}$ ,  $c_1, c_2$  相连,因此,将  $j = 10, j = 1, j = 2$  对应的  $b_{ij}$  的值设定为 1,其他没有有向弧连接的都设定为 0。以此类推,可以得到飞机起落架的优先关系矩阵如下:

	0	1	2	3	4	5	6	7	8	9	10
0	0	1	1	1	1	1	1	1	1	1	1
1	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0
4	0	1	1	0	0	0	0	0	0	0	1
5	0	1	1	1	1	0	0	0	0	0	1
6	0	1	1	0	1	0	0	0	1	0	1
7	0	1	1	0	1	0	0	0	1	0	1
8	0	1	1	0	1	0	0	0	0	0	1
9	0	1	1	0	1	0	0	0	0	0	1
10	0	1	1	0	0	0	0	0	0	0	0

### 2.5.2 气味搜索、视觉搜索过程的设计

气味搜索过程中,子种群果蝇个体在优先关系矩阵的指引下进行进化,又产生  $S$  只新果蝇个体,从而组成了新的子种群。每只新产生的子代果蝇相对于父代果蝇每次迭代只发生两个位置的交换,没有交换的保留即可。这样,子代果蝇既继承了父代果蝇的优点,又实现了更新,促进了种群的进化。其实现过程如图 6 所示。

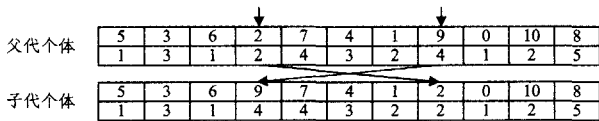


图 6 基于气味搜索

在经典的果蝇算法中,视觉搜索过程是用来引导整个种群向当前较优的搜索空间移动的,这样很容易使算法陷入局部最优,从而导致早熟现象。本文利用果蝇的视觉搜索对子种群中的个体进行评估,选出各子种群中的最优个体作为局部最优解;再采用局部最优解向全局最优解以概率  $P$  学习的机制,从而避免种群早熟的缺陷。

### 2.6 协同搜索过程的设计

在自然界,群体通过信息交流来提高搜索效率是一种常见的行为,如果蝇、蚂蚁、蜜蜂、鸟群等。果蝇通过交换位置信息来进行方位调整,从而有效地使群体飞向食物源。受这一行为的启示,将这一协同搜索过程引入改进的果蝇优化算法中,以增强算法的全局搜索能力。将文献中的改进优先交叉操作(the Improved Precedence Operation Crossover, IPOX)和多点保留交叉(Multi-point Preservative Crossover, MPX)引入到协同搜索机制中,并独立完成装配序列规划和工位分配。

不同于 IPOX 和 MPX<sup>[21]</sup>,引入学习机制,其他个体以概率  $P$  向最优的个体学习。

IPOX 的实现过程如下。

- 1)对于种群中每只果蝇的装配序列向量  $v_1$ ,选出种群中最优个体  $bFf_g$  的装配序列向量  $v_{b1}$ 。
- 2)将所有零件分成两个非空集合  $C_1$  和  $C_2$ 。对每一个零件随机生成一个随机数  $r \in (0, 1)$ ,当  $r < P$  时,将零件  $i$  加到  $C_1$ ;否则将其加到  $C_2$ 。如果  $C_1$  或  $C_2$  为空,将随机选择一个零件加入到空集中。
- 3)对于新个体  $Ff_{son}$  的装配序列规划向量  $v_{off1}$ ,在相同的位置上复制  $C_1$ ,剩余的空位按顺序从装配序列向量  $v_1$  中复制。

IPOX 的变换过程如图 7 所示。

随机向量 $r$	0.7	0.5	0.8	0.2	0.4	0.1	0.9	0.6	0.8	0.3	0.8
$v_{b1}$	5	3	6	2	7	4	1	9	0	10	8
$v_{off1}$	6	3	1	2	7	4	5	9	0	10	8
$v_1$	7	2	6	10	1	5	3	9	0	4	8

图 7 IPOX 对装配序列的改良

$C_1: \{2, 3, 4, 7, 9, 10\}, C_2: \{0, 1, 5, 6, 8\}, P = 0.7$ 。

MPX 的实现过程如下。

- 1)对种群中每只果蝇的工位分配向量  $v_2$ ,选出种群中最优个体  $bFf_g$  的装配序列向量  $v_{b2}$ 。
- 2)对每一个工位随机产生  $r \in (0, 1)$ ,当  $r < P$  时,工位分配按  $v_{b2}$  进行继承;否则按  $v_2$  有顺序地继承。这样就得到了新个体  $Ff_{son}$  的工位分配序列  $v_{off2}$ 。

MPX 的变换过程如图 8 所示。

随机向量 $r$	0.7	0.5	0.8	0.2	0.4	0.1	0.9	0.6	0.8	0.3	0.8
$bFf_g$	5	3	6	2	7	4	1	9	0	10	8
$v_{b2}$	1	3	1	2	4	3	2	4	1	2	5
$Ff_{son}$	6	3	1	2	7	4	5	9	0	10	8
$v_{off2}$	3	3	4	2	4	3	1	4	1	2	5
$v_1$	7	2	6	4	1	5	3	9	0	10	8
$v_2$	3	1	3	3	4	1	2	4	1	2	5

图 8 MPX 对工位序列进行的改进

### 2.7 序列几何可行性判断

经过 IPOX 和 MPX 操作后,得到新的果蝇个体,它代表的序列是否可行,还需要进行检测。除了对 APM, PCT 和 AIT 进行检查外,还需要对零件装配过程中的几何可行性进行检测。装配序列不仅要满足零件间的优先关系,还要满足零件之间的几何约束。本文通过集成干涉矩阵  $H_k$  来描述零件在装配方向(+X, +Y 和 +Z)上与其他零件的干涉关系。因此,在检测过程中,只要装配序列不违反干涉关系,就说明

装配序列满足几何可行性的要求。针对飞机起落架的集成干涉矩阵如下:

		0	1	2	3	4	5	6	7	8	9	10
	0	000	001	111	111	111	111	111	111	111	111	111
	1	000	000	000	000	000	000	000	000	000	000	000
	2	110	001	000	001	100	000	000	000	000	000	000
	3	110	000	000	000	110	000	000	000	000	000	000
	4	110	001	110	001	000	110	000	000	000	110	110
$H_k=5$	5	110	000	000	110	111	000	000	000	000	000	110
	6	110	000	000	001	001	000	000	000	111	000	000
	7	110	000	000	001	000	000	100	000	111	000	000
	8	110	000	000	001	111	000	111	000	000	000	000
	9	110	000	000	000	000	000	000	000	000	000	000
	10	110	011	001	000	110	000	000	000	000	000	000

表3 工位运输成本分布表<sup>[10]</sup>

零件	工位				
	F <sub>1</sub>	F <sub>2</sub>	F <sub>3</sub>	F <sub>4</sub>	F <sub>5</sub>
c <sub>0</sub>	0	9	8	8	9
c <sub>1</sub>	8	0	9	6	7
c <sub>2</sub>	5	9	0	6	4
c <sub>3</sub>	5	9	0	10	4
c <sub>4</sub>	7	3	4	0	5
c <sub>5</sub>	8	2	5	0	6
c <sub>6</sub>	5	3	4	0	3
c <sub>7</sub>	4	3	0	2	2
c <sub>8</sub>	3	4	0	5	1
c <sub>9</sub>	7	4	5	0	2
c <sub>10</sub>	9	8	6	0	6

### 3 实验结果与分析

#### 3.1 参数设置

改进的果蝇优化算法的主要参数有3个:种群的规模NP、在气味搜索阶段子种群产生的果蝇数量S以及学习的接受概率P。研究发现,子种群的规模是平衡局部搜索和全局搜索能力的决定因素。S取值大,表明追求的是局部搜索能力,这样会加大对许多劣质解的评估工作。对于概率P,取值太小,每一次循环对解的改进程度太小;取值太大,又易使种群出现早熟现象。所以,为确保改进的果蝇算法的效率,要设置合适的S值和P值。种群规模NP由于对算法的影响比较微弱,一般取大的值。综合考虑后,本文实验取:NP=80,S=3,P=0.7。

#### 3.2 实例仿真与分析

在算法确定、参数设定后,输入零件的定义、成本信息和装配操作等数据,并对数据进行格式转化和处理,最后用提出的改进果蝇优化算法产生与优化产品的装配序列和工位序列。

本文所用实例主要有11个零件,装配工位有5个;它的装配优先序图(APG)如图5所示,装配优先矩阵(APM)在2.5节给出,相应的装配信息表(AIT)如表2所列,工位能力表(PCT)已给出。在算法具体实现过程中,工具更换成本为1,夹具更换成本为2,工位运输成本如表3所列。

表2 起落架装配信息表(AIT)<sup>[10]</sup>

操作编码	装配工具	装配零件	各工位装配操作成本				
			F <sub>1</sub>	F <sub>2</sub>	F <sub>3</sub>	F <sub>4</sub>	F <sub>5</sub>
A <sub>1</sub>	T <sub>1</sub> T <sub>2</sub>	c <sub>8</sub> c <sub>7</sub>	3	4	—	—	5
A <sub>2</sub>	T <sub>1</sub> T <sub>2</sub> T <sub>3</sub>	c <sub>8</sub> c <sub>6</sub>	3	2	—	—	2
A <sub>3</sub>	T <sub>1</sub> T <sub>2</sub> T <sub>3</sub>	c <sub>8</sub> c <sub>4</sub>	2	2	—	2	—
A <sub>4</sub>	T <sub>1</sub> T <sub>4</sub>	c <sub>0</sub> c <sub>7</sub>	5	—	4	—	4
A <sub>5</sub>	T <sub>1</sub> T <sub>2</sub> T <sub>3</sub>	c <sub>0</sub> c <sub>6</sub>	2	2	—	—	—
A <sub>6</sub>	T <sub>1</sub> T <sub>3</sub>	c <sub>0</sub> c <sub>9</sub>	3	2	—	—	2
A <sub>7</sub>	T <sub>1</sub> T <sub>2</sub> T <sub>3</sub>	c <sub>0</sub> c <sub>5</sub>	4	4	—	5	—
A <sub>8</sub>	T <sub>1</sub> T <sub>3</sub>	c <sub>0</sub> c <sub>2</sub>	1	—	1	1	—
A <sub>9</sub>	T <sub>1</sub> T <sub>2</sub> T <sub>3</sub>	c <sub>0</sub> c <sub>4</sub>	1	1	—	2	—
A <sub>10</sub>	T <sub>1</sub> T <sub>2</sub> T <sub>3</sub>	c <sub>4</sub> c <sub>5</sub>	6	6	—	5	—
A <sub>11</sub>	T <sub>1</sub> T <sub>3</sub> T <sub>4</sub>	c <sub>9</sub> c <sub>10</sub>	1	1	—	—	—
A <sub>12</sub>	T <sub>1</sub> T <sub>2</sub> T <sub>3</sub>	c <sub>2</sub> c <sub>10</sub>	7	—	6	—	6
A <sub>13</sub>	T <sub>1</sub> T <sub>2</sub> T <sub>3</sub>	c <sub>0</sub> c <sub>10</sub>	1	1	—	—	—
A <sub>14</sub>	T <sub>2</sub> T <sub>3</sub> T <sub>5</sub>	c <sub>4</sub> c <sub>10</sub>	8	8	—	—	—
A <sub>15</sub>	T <sub>1</sub> T <sub>2</sub> T <sub>3</sub> T <sub>6</sub>	c <sub>1</sub> c <sub>10</sub>	9	9	—	—	—
A <sub>16</sub>	T <sub>1</sub> T <sub>2</sub>	c <sub>3</sub> c <sub>5</sub>	3	—	3	—	—

表4 果蝇算法运行结果

装配顺序	1	2	3	4	5	6	7	8	9	10	11
零件	c <sub>0</sub>	c <sub>9</sub>	c <sub>7</sub>	c <sub>6</sub>	c <sub>5</sub>	c <sub>4</sub>	c <sub>8</sub>	c <sub>10</sub>	c <sub>1</sub>	c <sub>2</sub>	c <sub>3</sub>
工位	F <sub>1</sub>	F <sub>2</sub>	F <sub>2</sub>	F <sub>2</sub>	F <sub>4</sub>	F <sub>4</sub>	F <sub>5</sub>	F <sub>2</sub>	F <sub>2</sub>	F <sub>3</sub>	F <sub>3</sub>
总成本	92										
运行时间	0.202s										

针对相同的实例,文献[10]用粒子群算法求出的装配成本是102,而本文用果蝇优化算法求出的最小成本是92,这表明果蝇算法在获取全局最优解的能力上优于粒子群算法。由于文献[10]没有给出算法的运行时间,所以这里没有进行比较。但是,从上述实验可知,在零件规模为11时,本文方法最多只需要0.202s,已经是一个很短的运行时间了,这得益于本文采用多子种群并行搜索并引入了协同搜索的学习机制。

结束语 为了更加接近实际装配的场景,打破在单一工位上做装配序列的传统,将改进的果蝇算法引入到多工位装

配序列规划问题上,从而建立了基于果蝇优化算法的多工位装配序列模型;给出了果蝇优化算法的双向量编码体系和更接近工程实际的适应度函数表达式;采用多子种群并行搜索方式,并引入 IPOX 与 MPX 操作算子和协同搜索的学习机制,兼顾了算法的局部和全局搜索能力,很好地提高了算法的效率和质量。

文中在进行实例验证时,通过许多学者总结的结果直接对  $NP, S, P$  进行设定,并未对它们之间的相互关系进行深入研究,如果利用田口方法<sup>[22]</sup>(Taguchi method)对它们三者之间的关系进行研究之后再行设定,有望使算法参数的设置更合理;算法在生成初始种群后,直接采用每只果蝇代表初始子种群进行局部搜索,这样可能会使搜索空间比较分散,导致算法搜索时间太长,如果采用正交化理论或均匀化理论对初始种群进行分组处理,使种群均匀地分布在搜索空间,将使算法搜索效率更佳。在下一步工作中,将从算法参数的相互影响关系和初始种群均匀化方面进行研究,以进一步提高算法效率。

### 参考文献

- [1] WANG J F, LI S Q, LIU J H, et al. Computer aided assembly planning a survey[J]. Journal of Engineering Graphics, 2005, 26(2): 1-6. (in Chinese)  
王俊峰,李世其,刘继红,等.计算机辅助装配规划研究综述[J].工程图学学报,2005,26(2):1-6.
- [2] DE MELLO H. A Correct and Complete Algorithm for the Generation of Mechanical Assembly Sequences[J]. IEEE Transaction Robotics and Automation, 1991, 2(7): 228-240.
- [3] GOTTIPOL U R B, GHOSH K. A simplified and efficient representation for evaluation and selection of assembly sequences[J]. Computers in Industry, 2003, 50(3): 251-264.
- [4] LI R, FU Y L, FENG H B. Assembly sequence planning based on connector-structure knowledge [J]. Computer Integrated Manufacturing Systems, 2008, 14(6): 1130-1135. (in Chinese)  
李荣,付宜利,封海波.基于连接结构知识的装配序列规划[J].计算机集成制造系统,2008,14(6):1130-1135.
- [5] WANG Y, LIU J H, LI L S. Assembly sequence merging based on assembly unit partitioning [J]. International Journal of Advanced Manufacturing Technology, 2009, 45(7): 808-820.
- [6] SMITH G C, SMITH S. An enhanced genetic algorithm for automated assembly planning[J]. Robotics and Computer Integrated Manufacturing, 2000, 18(5): 355-364.
- [7] ZENG B, LI M F, ZHANG Y. Assembly sequence planning based on improved firefly algorithm [J]. Computer Integrated Manufacturing Systems, 2014, 20(4): 799-806. (in Chinese)  
曾冰,李明富,张翼.基于改进萤火虫算法的装配序列规划方法[J].计算机集成制造系统,2014,20(4):799-806.
- [8] NING L H, GU T L. Immune algorithm for assembly sequence planning problem[J]. Computer Integrated Manufacturing Systems, 2007, 13(4): 762-769. (in Chinese)  
宁黎华,古天龙.基于免疫算法的装配序列规划问题求解[J].计算机集成制造,2007,13(4):762-769.
- [9] SHI S C, LI R, FU Y L, et al. Assembly sequence planning based on improved ant colony algorithm [J]. Machinery Design & Manufacture, 2010, 16(6): 1189-1194. (in Chinese)  
史士财,李荣,付宜利,等.基于改进蚁群算法的装配序列规划[J].计算机集成制造系统,2010,16(6):1189-1194.
- [10] WANG F C, SUN Y C, LI N. Multi Station Assembly Sequence Planning Based on Particle Swarm Optimization Algorithm[J]. Journal of Mechanical Engineering, 2012, 48(9): 1-8. (in Chinese)  
王丰产,孙有朝,李娜.多工位装配序列粒子群优化算法[J].机械工程学报,2012,48(9):1-8.
- [11] PAN W T. A new fruit fly optimisation algorithm; taking the financial distress model as an example[J]. Knowledge-Based Systems, 2012(2): 69-74.
- [12] HAN J, WANG P, YANG X. Tuning of PID controller based on fruit fly optimisation algorithm[C]// International Conference on Mechatronics & Automation. 2012: 409-413.
- [13] WANG L, ZHENG X L, WANG S Y. A novel binary fruit fly optimisation algorithm for solving the multidimensional knapsack problem[J]. Knowledge-Based Systems, 2013, 48(2): 217-23.
- [14] LIN S M. Analysis of service satisfaction in web auction logistics service using a combination of Fruit fly optimisation algorithm and general regression neural network [J]. Neural Computing and Applications, 2013, 22(3): 783-791.
- [15] SHENG W, BAO Y. Fruit fly optimisation algorithm based fractional order fuzzyPID controller for electronic throttle[J]. Non-linear Dynamics, 2013, 73(1/2): 611-619.
- [16] LI J Q, PAN Q K, KUN M, et al. Solving the steelmaking casting problem using an effective fruit fly optimization algorithm [J]. Knowledge-Based Systems, 2014, 72(5): 28-36.
- [17] MOUSAVI S M, ALIKAR N, NIAKI S T A. Optimization a location allocation-inventory problem in a two-echelon supply chain network; A modified fruit fly optimization algorithm[J]. Computer & Industrial Engineering, 2015, 87(c): 543-560.
- [18] NIU J W, ZHONG W M, LIANG Y. Fruit fly optimization algorithm based on differential evolution and its application on gasification process operation optimization[J]. Knowledge-Based Systems, 2015, 88(c): 253-263.
- [19] WANG L, LIU R, LIU S. An effective and efficient fruit fly optimization algorithm with levelprobability policy and its applications[J]. Knowledge-Based Systems, 2016, 97(c): 158-174.
- [20] LEI X, DING Y, FUJITA H. Identification of dynamic protein complexes based on fruit fly optimization algorithm[J]. Knowledge-Based Systems, 2016, 105(c): 270-277.
- [21] WANG X J, GAO L, ZHANG C Y, et al. A multi-objective genetic algorithm based on immune and entropy principle for flexible job-shop scheduling problem[J]. The International Journal of Advanced Manufacturing Technology, 2010, 51(5): 757-767.
- [22] MONTGOMERY D C. Design and Analysis of Experiments[J]. Journal of the American Statistical Association, 2005, 81(16): 308.