

DOI:10.3969/j.issn.1001-4551.2022.01.012

电子液压制动控制单元的混流 装配线平衡优化研究*

蔡启明, 李伟伟

(南京航空航天大学 经济与管理学院, 江苏 南京 211106)

摘要:针对国内某汽车电子液压制动控制单元的混流装配线存在工作站数目过多、生产节拍不合理、作业负荷不均衡的问题,对该混流装配线进行了数学建模、自适应遗传算法求解和分析研究。首先,介绍了混流装配线平衡的基本概念,以及汽车电子液压制动控制单元的混流装配线;然后,在传统混流装配线平衡的基础上,综合考虑了工作站的数目最少、生产节拍最小、装配线平滑指数最小等 3 个目标函数,建立了混流装配线平衡的数学模型;之后,按照遗传算法的流程图,描述了改进后自适应遗传算法的步骤;最后,利用 MATLAB 编写了自适应遗传算法,求解了混流装配线平衡问题,并对装配线的运行结果进行了分析。研究结果表明:采用自适应遗传算法得到的优化结果明显优于采用传统遗传算法的结果,可减少 1 个工作站,降低工人与设备的成本;生产节拍增加 2.66 秒/件,可完成目标产量并减少库存成本;装配线平滑指数降低 59.36%,作业负荷更加均衡。

关键词:混流装配线平衡;电子液压制动控制单元;多目标优化;自适应遗传算法

中图分类号:TH165.2 文献标识码:A

文章编号:1001-4551(2022)01-0077-10

Optimization of mixed-model assembly line balancing for hydraulic electronic control unit

CAI Qi-ming, LI Wei-wei

(College of Economics and Management, Nanjing University of
Aeronautics and Astronautics, Nanjing 211106, China)

Abstract: Aiming at the problems of excessive number of workstations, unreasonable cycle time, and unbalanced workload variance in the domestic automobile mixed-model assembly line of a Hydraulic Electronic Control Unit (HECU), mathematical modeling, adaptive genetic algorithm and result analysis were carried out. Firstly, the basic concept of the mixed-model assembly line balancing and the HECU mixed-model assembly line were introduced. Then, based on the traditional mixed-model assembly line balancing problem, the three objective functions of the minimum number of workstations, the smallest cycle time, and the minimum workload variance within workstations were considered to establish a mathematical model of mixed-model assembly line balancing. Then, following the flow chart of genetic algorithm, the steps of the improved adaptive genetic algorithm were described. Finally, MATLAB was used to write an adaptive genetic algorithm to solve the mixed-model assembly line balancing problem, and the running results were analyzed. The research results indicate that the optimization result of adaptive genetic algorithm is significantly better than traditional genetic algorithm. One workstation is reduced, which can decrease the cost of workers and equipment. The cycle time is increased by 2.66 seconds/piece, which achieves the target output and can reduce inventory costs. The minimum workload variance within workstations is reduced by 59.36%, which can make the workload more balanced.

Key words: mixed-model assembly line balancing; hydraulic electronic control unit; multi-objective optimization; adaptive genetic algorithm

收稿日期:2021-07-20

作者简介:蔡启明(1964-),男,上海人,博士,教授,主要从事物流与供应链管理方面的研究。E-mail:qiming@nuaa.edu.cn

0 引 言

国内某汽车零部件制造商加大对零部件的研发与投入,新开设了汽车电子液压制动控制单元的混流装配线。

电子液压制动系统是一种比机械制动更为高级的汽车安全技术,能够大大缩短制动距离。而电子液压制动控制单元是电子液压制动控制系统的核心元器件,对电子液压制动系统发出指令,直接影响汽车的制动性能。

混流装配线要能够满足多品种、小批量的需求,是一种能将多种具有相似工业流程的产品进行混合生产的装配线。但是,企业运用混流装配线时涉及到混流装配线的平衡问题,例如:工作站的数目过多,占用过大的面积,造成空间的浪费;各个工作站的装配时间分布差距过大会导致生产瓶颈的出现,工人负荷不均,生产停滞;分配工人在不同的工作站,不同的分配方式带来的成本也不一样;装配线的生产节拍,影响到企业能否完成企业的生产目标,等等。

目前,有很多学者关注混流装配线的研究。姚午厚^[1]研究了混流装配线的多种优化算法对混流装配线进行了平衡的排序,开发了智能家居行业混流装配线平衡与排序系统。揣博君^[2]以生产节拍、平滑指数和操作工人成本为多目标函数,用改进的遗传算法来进行求解。宋春雷^[3]以最小化完工时间、零部件消耗速率均衡化和相邻产品相似度最大化为目标,建立了多目标投产排序模型,应用遗传算法和 Pareto 排序思想对模型进行了求解。张志文^[4,5]建立了以 MODA-PTS 为基础的标准工时系统,并结合线平衡理论建立了流水线平衡模型,再利用遗传算法求解出了最优化方案。戴韬^[6]对混流汽车液压电子制动控制单元工艺过程进行了数学建模,采用优化遗传算法进行了求解。苏平等人^[7]提出了一种综合运用遗传算法和仿真分析的混合装配线平衡问题的求解方法。胡罗克等人^[8]应用工业工程方法和仿真技术,对实际企业的装配线做了系统科学的平衡分析和改善。BUKCHIN Y 等人^[9]通过对工人安排方案进行编码,用遗传变异寻求最小化工人空闲时间,提高了生产线的效率。LI Z 等人^[10]为了解决双面混合装配线的平衡问题,采用了多种启发式算法综合比较,证明了改进的迭代贪婪算法具有有效性。RASHID M^[11]利用多目标离散粒子群优化算法来对整个混流装配线平衡和排序的问题进行了建模和优化。SAHU A 等

人^[12]提出了一种新的遗传算法,提高了求解最小工作站数、瓶颈时间等参数的效率。AKPINAR S, SUNGUR B, DELICE Y 等人^[13-15]对混流装配线的工人分配进行了优化。

但是上述学者关注的是混流装配线的研究,还很少有人对汽车电子液压制动控制单元的混流装配线进行研究。

笔者通过建立以工作站数最小、装配节拍最小以及各工作站装配时间的均方差最小的多目标数学模型,利用改进的自适应遗传算法加快收敛速度,得到满意解,降低设备与库存成本,减少各工作站装配时间的波动,平衡生产线。

1 混流装配线平衡理论和研究问题

1.1 基本概念

此处所涉及到的几个关键的混流装配线键术语,详细基本概念可以参见参考文献中的前 4 条文献。

(1) 工序优先关系图。表示产品装配过程中各个作业元素的先后关系。

A、B 两种产品的作业元素优先图如图 1 所示。

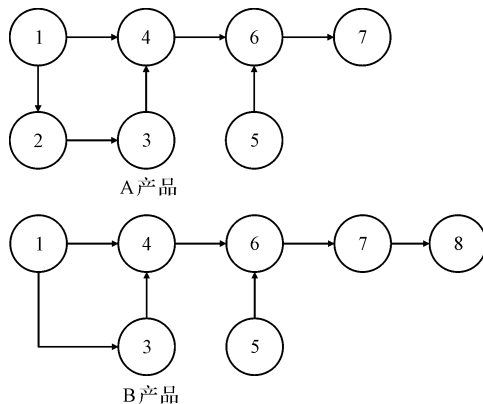


图 1 A、B 产品的作业元素优先图

(2) 优先关系矩阵。将产品的作业元素优先图用矩阵来表示。

上图中 A、B 产品的优先关系矩阵如图 2 所示。

$$A = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

图 2 A、B 产品的优先关系矩阵

(3) 综合作业优先关系图。A、B 两种产品的综合作业优先关系图如图 3 所示。

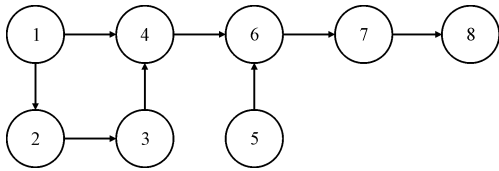


图 3 A、B 两种产品的综合作业优先关系图

1.2 电子液压制动控制单元的混流装配线

通过对某汽车零部件公司的电子液压制动控制单元的混流装配线进行研究发现,该公司主要装配 3 种不同型号的控制单元,包括零件的压装、电子控制-液压控制-马达合装、综合测试 3 大部分。其中,零件的压装是指利用压力机对组件进行压装;电子控制-液压控制-马达合装是指利用螺丝、压力机将液压控制单元、电子控制单元和电机三者连接;综合测试部分包括对传感器、油泵、制动器、电机等关键零件进行测试。

笔者绘制出 3 种制动控制单元的综合作业优先关系图,如图 4 所示。

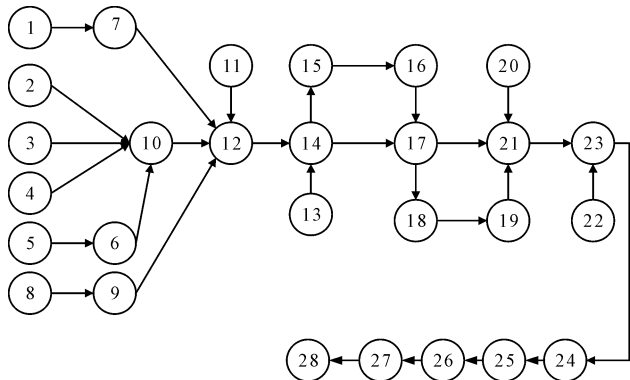


图 4 3 种制动控制单元的综合作业优先关系图

已知 A、B、C 3 种产品的需求量 D_1 、 D_2 、 D_3 分别为 400、100、100,所以各产品的需求比例 q_1 、 q_2 、 q_3 分别为 2/3、1/6、1/6;测得每个产品各工序的时间,并求得在综合作业优先关系图(即在混流装配线)中,装配这 3

种产品时各个工序的时间,如表 1 所示。

表 1 各工序的时间

工序	各产品 p 工			综合产品 工序时间	工序	各产品 p 工			综合产品 工序时间
	序时间 t_{pm}	序时间 t_{pm}	序时间 t_{pm}			序时间 t_{pm}	序时间 t_{pm}	序时间 t_{pm}	
1	8	8	8	8.00	15	10	10	10	10.00
2	13	13	13	13.00	16	10	0	0	6.67
3	21	17	12	18.83	17	10	10	10	10.00
4	10	0	0	6.67	18	8	8	0	6.67
5	8	0	0	5.33	19	12	0	0	8.00
6	12	0	0	8.00	20	12	12	12	12.00
7	13	13	13	13.00	21	12	10	8	11.00
8	0	0	6	1.00	22	10	8	8	9.33
9	0	0	8	1.33	23	16	12	12	14.67
10	12	0	0	8.00	24	16	10	10	14.00
11	12	12	12	12.00	25	13	13	13	13.00
12	18	18	18	18.00	26	13	13	13	13.00
13	10	8	4	8.67	27	20	15	15	18.33
14	6	6	6	6.00	28	13	13	13	13.00

由于综合作业优先顺序图中,各个作业元素的时间为 $t_m = \sum_{p=1}^P q_p t_{pm}$, 因此可以求得表中的各个工序时间 t_m 。

2 装配线平衡数学模型的建立

传统的混流装配线平衡主要包括 3 类:

- (1) 给定生产节拍,求解最小的工作站数;
- (2) 给定工作站数目,使得生产节拍最小;
- (3) 给定生产节拍和工作站数,使得各工作站负荷均衡。

汽车电子液压制动控制单元的混流装配线平衡问题在传统问题的基础上进一步拓展,笔者综合考虑生产节拍、工作站数和工作站间负荷均衡 3 个指标,对混流装配线进行统筹优化。

以下是具体的数学模型。

2.1 数学符号假设

TT —总有效生产时间;

D —所有产品的总需求量;

P —产品的种类;

D_p —第 p 种产品的需求量;

CT_{theory} —理论生产节拍, $CT_{theory} = TT/D$;

CT_{real} —实际生产节拍;

m —第 m 个作业元素, M —作业元素的总和;

q_p —第 p 种产品的比例, $q_p = D_p/D$;

w —第 w 个工作站, W_{real} —实际工作站总数,

W_{theory} —理论工作站数;

x_{mw} —作业任务的分配矩阵, $x_{mw} = \begin{cases} 1, & \text{第 } m \text{ 个作业元素被分配到第 } w \text{ 个工作站} \\ 0, & \text{第 } m \text{ 个作业元素未被分配到第 } w \text{ 个工作站} \end{cases}$;

t_{pm} —第 p 种产品第 m 个工序所花费的标准时间;

t_m —综合作业优先顺序图中各个作业元素的时间, $t_m = \sum_{p=1}^P q_p t_{pm}$ 。

2.2 目标函数

为了使得企业获得更多的利润,对混流装配线平衡的研究已经不仅仅局限于单目标、双目标甚至多目标以提升多方面效益,例如:工人成本费用最少、生产节拍最小、工作站数目最少、各个工作站的负荷差异最小、传送带停止时间最少、总切换时间最少,等等。此处主要根据企业的实际需求,设定了 3 个目标函数:工作站的数目最少、生产节拍最小、各工作站的负荷均衡。

(1) 工作站的数目最少

各工作站的数目最少为混流装配线平衡研究中的第一类问题,即在知道理论生产节拍时,确定最小的工作站的数目,来减少装配所需的空间、工人成本,等等。根据企业当前的有效生产时间和在有效时间内需装配的产品数量,来确定理论生产节拍,再根据综合作业优先图,得到总时间与理论的生产节拍的比值,得到理论上最少工作站的数目,即:

$$W_{theory} = \left\lceil \frac{\sum_{m=1}^M \sum_{p=1}^P q_p t_{pm}}{CT_{theory}} \right\rceil + 1 \quad (1)$$

在进行作业元素分配时,会发现 W_{theory} 太小,工作站的数目不能容纳所有的作业元素,这时再逐个增加工作站的数量得到实际工作站数目 W_{real} ,即:

$$W_{real} = W_{theory} + k \quad (2)$$

式中: k —新增的工作站数目。

(2) 生产节拍最小

求生产节拍最小是混流装配线平衡研究中的第二类问题,就是使得各工作站总时间的最大值最小化,它决定了能否在规定的时间内生产出足够的产品数量来满足客户的需求。

所以,实际的生产节拍最小化为:

$$\min CT_{real} = \max_{1 \leq w \leq W} \left\{ \sum_{p=1}^P \sum_{m=1}^M q_p t_{pm} x_{mw} \right\} \quad (3)$$

(3) 装配线平滑指数 L 最小

装配线平滑指数 L 用以衡量各工作地之间的作业负荷分布是否均匀,它是混流装配线中的第 3 类问题。

装配线平滑指数 L 越小,表明各个工作站之间的加工时间越接近,作业元素的分配越合理。

装配线平滑指数 L 为:

$$\min L = \sqrt{\frac{\sum_{w=1}^W \left(\sum_{p=1}^P \sum_{m=1}^M q_p t_{pm} x_{mw} - \sum_{w=1}^W \sum_{p=1}^P \sum_{m=1}^M q_p t_{pm} x_{mw} / W \right)^2}{W}} \quad (4)$$

2.3 约束条件

(1) 每一个作业元素只能分配给一个工作站,即:

$$\sum_{w=1}^W x_{mw} = 1 \quad (5)$$

(2) 每个工作站生产的时间不能超过 CT_{theory} ,即:

$$\sum_{p=1}^P \sum_{m=1}^M t_{pm} x_{mw} \leq CT_{theory} \quad (6)$$

(3) 必须要满足各个作业元素之间存在着产品工艺上的先后顺序,即:

$$\sum_{w=1}^W w x_{iw} - \sum_{w=1}^W w x_{mw} \leq 0, \forall i \in pre(m) \quad (7)$$

式中: $pre(m)$ —在工作元素 m 的紧前工序集合。

(4) 各个决策变量必须为 0 或 1,即:

$$x_{iw}, x_{mw} \in \{0, 1\} \quad \forall i, m, w \quad (8)$$

2.4 目标函数权重的确定

因此,这是一个多目标的优化问题,绝大多数情况下,多个目标函数不能同时取得最优解。对于决策者而言,当目标函数达到一定的范围即可认为是最优解。这里采用加权法来赋予 3 个目标函数不同的加权,使得多目标优化问题转化为单目标的优化问题,而加权由决策者的偏好决定。

这里采用层次分析法(AHP)的判断矩阵来确定各目标的权重,判断矩阵由 3 个目标重要性程度的比值组成。判断矩阵的元素一般用数字 1~9 来确定,代表不同的重要性程度。工作站的数目影响了混流装配线的占地面积、工人成本、空闲时间,等等。

实际生产节拍的大小决定了单位时间里生产的产品数量以及产品能否按时交付给客户,其重要性不言而喻,明显重要于各个工作站之间的负荷分配是否均衡。

因此,笔者构造多目标的判断矩阵如表 2 所示。

表 2 目标函数的判断矩阵

目标	工作站数	生产节拍	装配线平滑指数
工作站数	1	2	5
生产节拍	1/2	1	4
装配线平滑指数	1/5	1/4	1

根据判断矩阵求解,可得到 3 个目标的权重分别

为 0.57、0.33 和 0.1。此时,多目标优化就转化成了单目标优化,即:

$$\min f = 0.57W_{\text{real}} + 0.33CT_{\text{real}} + 0.1L \quad (9)$$

$$\begin{cases} \sum_{w=1}^W x_{mw} = 1 \\ \sum_{p=1}^P \sum_{m=1}^M t_{pm} x_{mw} \leq CT_{\text{theory}} \\ \sum_{w=1}^W wx_{iw} - \sum_{w=1}^W wx_{mw} \leq 0, \forall i \in \text{pre}(m) \\ x_{iw}, x_{mw} \in \{0, 1\}, \forall i, m, w \end{cases} \quad (10)$$

3 自适应遗传算法求解

该类数学模型属于 NP-hard 问题,一般利用启发式算法求解。遗传算法就是一种启发式算法,根据适者生存的原则,利用适应度函数来对方案进行评价,从而择优汰劣;各个方案之间再进行交叉、变异生成更好的方案,再次进行择优汰劣,如此迭代一定的次数得到最优方案^[16]。

遗传算法流程图如图 5 所示。

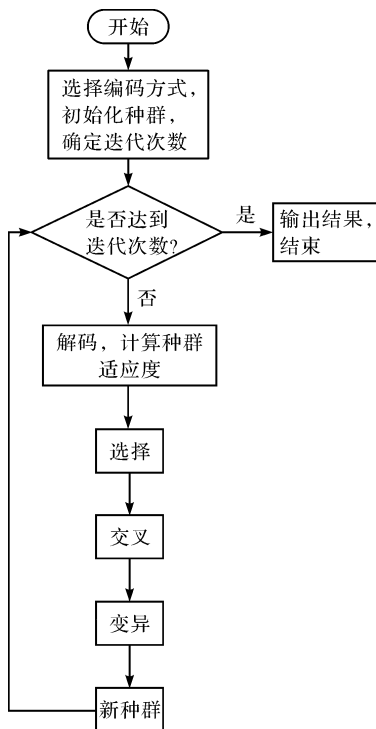


图 5 遗传算法流程图

3.1 编码方式

自适应遗传算法的第一步就是确定编码的方式,合理的编码方式便于计算适应度,简化交叉和变异操作,提升算法的执行效率。编码方式分为二进制编码、整数编码、实数编码等。不同于标准遗传算法的编码

方式,笔者采用基于综合作业优先图的整数编码,各个位置上的基因排列必须遵循作业元素的先后顺序,并且每个工序只能出现一次,染色体的长度与工序的个数相等。

根据综合作业关系优先矩阵来得到一条染色体的基因序列,其编码过程分为两步:

(1) 创建 $s = \text{Judge}(\mathbf{PA})$ 函数,用于生成可以分配的工序 s , s 是函数返回值,并且这些 s 满足当前综合作业关系优先矩阵条件。假设 \mathbf{PA} 为综合作业关系优先矩阵, y 代表工序的个数, i 代表工序的编号。具体操作过程如下:

Step1: $i = 1$, 判断 \mathbf{PA} 第 i 列是否全部为 0, 如果是则将工序 i 放入 s 中去, 否则执行 Step2;

Step2: $i = i + 1$, 重复 Step1, 直到 $i = y$ 结束, 将此时的 s 作为函数的返回值;

(2) 创建 $c = P_chrom(\mathbf{PA})$ 函数,用于生成当前 \mathbf{PA} 条件下的一条染色体 c , 染色体 c 是函数返回值。 y 代表工序的个数, j 代表 s 的长度, i 代表工序的编号。具体操作过程如下:

Step1: $i = 1$, 利用 $\text{Judge}(\mathbf{PA})$ 函数得到 s ;

Step2: 从 s 中随机抽取一个工序 t 放入染色体 c 中;

Step3: 使得 \mathbf{PA} 中的 t 列为 inf , t 行为 0;

Step4: $i = i + 1$, 重复 Step1、Step2、Step3, 直到 $i = y$ 结束, 将此时的 c 作为函数的返回值。

下面将图 2 中 A 产品的优先关系矩阵作为 \mathbf{PA} , 举例说明编码操作:

Step1: $i = 1$, 利用 $\text{Judge}(\mathbf{PA})$ 函数得到 $s = \{1, 5\}$;

Step2: 从 s 中随机抽取一个工序 5 放入染色体 c 中;

Step3: 使得 \mathbf{PA} 中第 5 列为 inf , 第 5 行为 0。此时, \mathbf{PA} 作业优先关系矩阵如图 6 所示。

$$\mathbf{PA} = \begin{bmatrix} 0 & 1 & 0 & 1 & \text{inf} & 0 & 0 \\ 0 & 0 & 1 & 0 & \text{inf} & 0 & 0 \\ 0 & 0 & 0 & 1 & \text{inf} & 0 & 0 \\ 0 & 0 & 0 & 0 & \text{inf} & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \text{inf} & 0 & 1 \\ 0 & 0 & 0 & 0 & \text{inf} & 0 & 0 \end{bmatrix}$$

图 6 \mathbf{PA} 作业优先关系矩阵

Step4: $i = 2$, 利用 $\text{Judge}(\mathbf{PA})$ 函数得到 $s = \{1\}$;

Step5: 从 s 中随机抽取工序 1 放入 c 中;

Step6: 使得 \mathbf{PA} 中第 1 列为 inf , 第 1 行为 0。此

时, PA 作业优先关系矩阵如图 7 所示。

$$PA = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \text{inf} & 0 & 1 & 0 & \text{inf} & 0 & 0 \\ \text{inf} & 0 & 0 & 1 & \text{inf} & 0 & 0 \\ \text{inf} & 0 & 0 & 0 & \text{inf} & 1 & 0 \\ \text{inf} & 0 & 0 & 0 & 0 & 0 & 0 \\ \text{inf} & 0 & 0 & 0 & \text{inf} & 0 & 1 \\ \text{inf} & 0 & 0 & 0 & \text{inf} & 0 & 0 \end{bmatrix}$$

图 7 PA 作业优先关系矩阵

Step7: 重复以上步骤直到安排完所有工序。

最终, 可以随机得到 A 产品的一条染色体如图 8 所示。

5	1	2	3	4	6	7
---	---	---	---	---	---	---

图 8 A 产品的一条随机染色体

3.2 种群初始化

种群初始化就是确定随机产生的染色体数目。种群数目过大会导致算法收敛过早, 并且造成算法运行时间变长, 效率降低。种群数目过小可能使得算法“早熟”, 导致适应度陷入局部最优解。因此, 要合理确定种群的数目。笔者设置初始种群 pop 的大小 n 为 80, 具体初始化种群的算法如下所示:

Step1: $i = 1$, 利用 $c = P_chrom(PA)$ 函数生成一条染色体作为 pop 的第 i 行;

Step2: $i = i + 1$, 当 i 到达种群的大小 n 时, 算法结束。

3.3 计算适应度与译码

在自适应遗传算法中, 可以根据适应度 f' 来判断染色体的优劣和靠近最优解的程度。因为此处选择了轮盘赌选择法, 适应度大的染色体, 遗传到下一代的概率大; 反之, 概率则小。适应度值就像是数学模型中的目标函数, 适应度越大的染色体越接近最优解。而在式(9)中, 目标函数的最优解是一个最小值, 因此需要对目标函数进行转化。该文以 $f' = 50 - 0.57W_{\text{real}} + 0.33CT_{\text{real}} + 0.1L$ 作为适应度函数。

当已知一个种群时, 并不能比较染色体之间的优劣。因为种群的染色体中包含的信息只有基因的排列顺序(即工序的顺序), 比较种群的优劣需要最小工作站数 W_{real} 、实际生产节拍 CT_{real} 和装配线平滑指数 L 的信息。译码的操作就是根据一条染色体排列的顺序求得 W_{real} 、 CT_{real} 、 L 。

3.4 选择

选择算子是为了保留优质染色体, 淘汰劣质染色

体。该文选择了轮盘赌法选择法作为选择算子, 它模拟博彩游戏的轮盘赌, 一个轮盘被划分为 n 个扇形(n 为种群的大小), 而每个扇形的面积与它所表示的染色体的适应度成正比; 为了选择种群中的个体, 设想有一个指针指向轮盘, 转动轮盘, 当轮盘停止后, 指针所指向的染色体被选择。因为一个染色体的适应度越大表示该染色体的扇形面积越大, 所以它被选择的可能性也就越大。

具体操作如下:

Step1: 计算种群各个染色体的适应度 f_i ;

Step2: 计算种群所有染色体的适应度所占的比例

$$f_i / \sum_{i=1}^n f_i;$$

Step3: 计算各个染色体的累积概率 $q_i = \sum_{j=1}^i f_j$;

Step4: 在 $[0, 1]$ 区间产生一个随机数 r ;

Step5: 若 $r < q_1$, 则选择染色体 1, 否则选择染色体 k , 使得 $q_{k-1} < r < q_k$;

Step6: 重复 Step4 和 Step5 共 n 次。

3.5 交叉

显然, 一定数量种群的染色体, 一般很难得到最优解。那么, 在种群数量固定的情况下, 要得到或者是接近最优解, 就要使染色体的基因相互交叉, 就是使种群多样化的一种算子。此处的交叉操作不同于一般遗传算法中染色体的交叉, 随意地交叉会破坏优秀染色体中符合综合作业优先关系的基因。因此, 笔者采用了两点交叉的方式。

为了方便理解, 下面用一个例子来说明本文交叉算子的思想, 假设父代 1 和父代 2 的基因排列顺序均满足某个作业装配工序的优先顺序, 如图 9 所示。

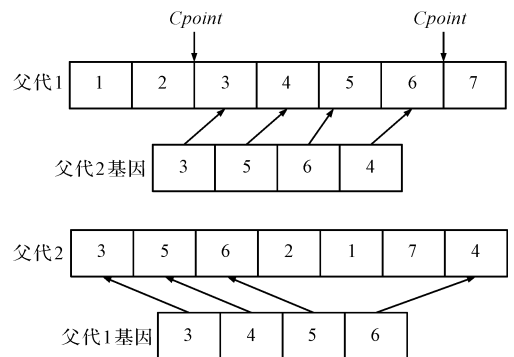


图 9 两点交叉算子的示意图

Step1: 当 $rand < p_c$ (交叉概率) 时, 首先在 $Chrom1$ (即父代 1) 中随机产生两个交叉点 $Cpoint1$ 和 $Cpoint2$, 图 9 中的 $\{3, 4, 5, 6\}$ 是 $Chrom1$ 在交叉算子中

提供的基因, 将其存入到变量 $temp$ 中去;

p_c 的表达式为:

$$p_c = p_{cmax} + \frac{p_{cmin} - p_{cmax}}{m} \cdot i \quad (11)$$

式中: p_{cmax} —最大交叉概率; p_{cmin} —最小交叉概率; i —当前迭代次数; m —总迭代次数。

Step2: 逐个找到变量 $temp$ 中的成员在 $Chrom2$ (即父代 2) 中的位置, 存入变量 $index$ 中 ($index$ 的含义是 $Chrom1$ 交叉的染色体在 $Chrom2$ 中的索引), 即 $index = \{1, 7, 2, 3\}$;

Step3: 将 $index$ 从小到大进行排序, 得到新的 $index = \{1, 2, 3, 7\}$ (新的 $index$ 即为 $temp$ 在 $Chrom2$ 中的排列顺序, 有 $Chrom2(index) = \{3, 5, 6, 4\}$, 即为图 9 中箭头所指父代 2 的基因);

Step4: 在 $Chrom1$ 中 $Cpoint1$ 到 $Cpoint2$ 基因位置存入 $Chrom2(index) = \{3, 5, 6, 4\}$, 生成子代 1。

Step5: 在 $Chrom2$ 中的索引位置即 $index = \{1, 2, 3, 7\}$, 分别存入 $temp = \{3, 4, 5, 6\}$ 。生成子代 2。

新生成的子代 1 和子代 2 不仅丰富了种群的多样性, 增加了搜索的范围, 还符合综合优先关系图的优先顺序。对于更加复杂染色体的交叉算子, 原理与这个例子一致。

3.6 变异

为了丰富种群的多样性, 增加搜索范围, 此处引入变异算子。变异就是使得染色体中的某个基因发生变化, 然而平凡的变化会破坏基因应符合的顺序, 产生不可行解。为了消除不可行解, 笔者采用了考虑综合作业优先关系矩阵的单点变异。

为了方便更好地理解变异算法, 下面举例说明, 已知某混流装配线的综合作业优先关系图和综合作业优先关系矩阵如图 10、图 11 所示。

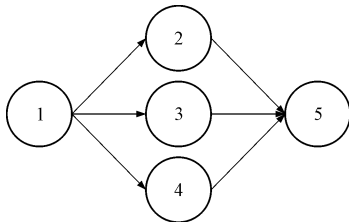


图 10 综合作业优先关系图

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

图 11 综合作业优先关系矩阵

假设已知一条满足该装配顺序的染色体 $c = \{1, 2, 3, 4, 5, 6\}$, 变异思想主要是随机产生一个变异点 $mpoint$; 前面的基因序列保持不变, 将 $mpoint$ 处基因及后面的基因序列重新排列。

解决这个例子算法的步骤如下:

Step1: 当 $rand < p_m$ (变异概率) 时, 产生一个随机变异点 $mpoint$, 假设在 3 处, 如图 12 所示。

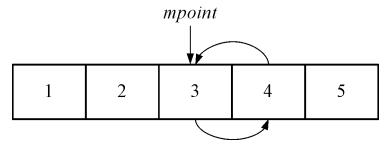


图 12 单点变异示意图

p_m 的表达式为:

$$p_m = p_{mmax} + \frac{p_{mmin} - p_{mmax}}{m} \cdot i \quad (12)$$

式中: p_{mmax} —最大变异概率; p_{mmin} —最小变异概率; i —当前迭代次数; m —总迭代次数。

Step2: 当基因 1、2 保持不变, 求出此时的综合作业优先矩阵 PA 如图 13 所示。

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ inf & inf & 0 & 0 & 1 \\ inf & inf & 0 & 0 & 1 \\ inf & inf & 0 & 0 & 0 \end{bmatrix}$$

图 13 变异后的综合作业优先关系矩阵

Step3: 用 3.1 中的 $Judge(PA)$ 函数得到 $s = \{3, 4\}$;

Step4: 随机选择一个基因, 这里假设选择基因 4 放入染色体 3 号位中;

Step5: 再保持基因 1、2、5 不变, 求出此时的综合作业优先矩阵 PA ;

Step6: 重复上述步骤, 最终得到一条变异后的染色体如图 14 所示。

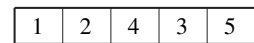


图 14 变异后的染色体

新生成的染色体再次丰富了种群的多样性, 也增加了算法的搜索范围, 防止算法陷入局部最优解。对于该文更加复杂染色体的变异算子原理与此例一致。

4 自适应遗传算法运行结果分析

4.1 参数设置

第一次试验将种群大小设定为 $n = 80$, 迭代次数 $m = \{100, 200, 300, 400\}$, 最大交叉概率为 0.9, 最小交叉概率为 0.5, 最大变异概率为 0.01, 最小变异概率为

0.005, 目标在 8 h 生产 A、B、C 3 种产品共 600 件。

MATLAB 的执行结果, 不同的迭代次数与适应度函数值的关系, 如图 15 所示。

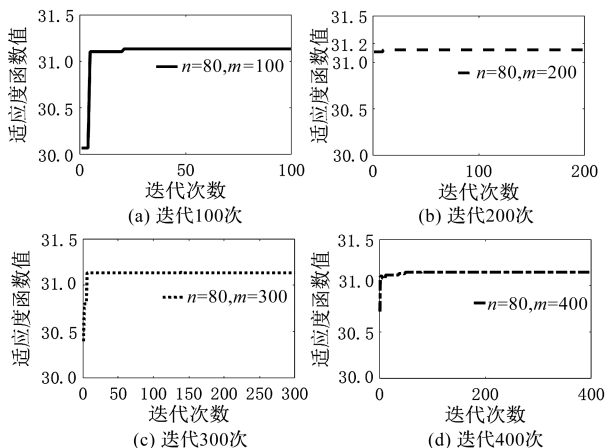


图 15 不同的迭代次数与适应度函数值的关系

图 15 是种群为 80 时, 不同的迭代次数与适应度函数值的关系。

MATLAB 的执行结果, 即不同的迭代次数对适应度函数值的影响, 如图 16 所示。

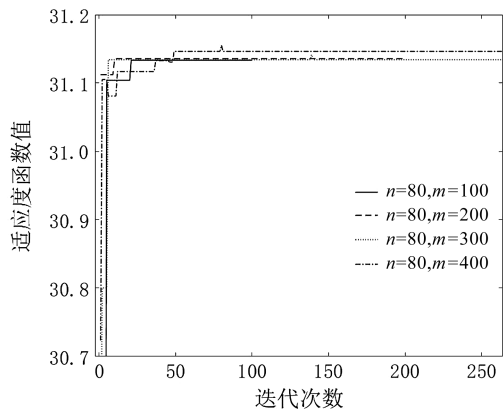


图 16 不同的迭代次数对适应度函数值的影响

图 16 将图 15 中的 4 条曲线放入一张图中, 从而更好地观察适应度函数收敛的重合程度。结果发现, 当迭代次数达到 60 左右时, 适应度函数值收敛, 再增加迭代的次数, 适应度函数值无明显的变化。因此, 综合考虑算法的效率和局部最优解的情况, 笔者将迭代的次数设定为 100。

第二次试验将种群大小设定为 $n = \{80, 160, 240, 320\}$, 迭代次数 $m = 100$, 最大交叉概率为 0.9, 最小交叉概率为 0.5, 最大变异概率为 0.01, 最小变异概率为 0.005, 目标在 8 h 生产 A、B、C 3 种产品共 600 件。

执行的结果, 即不同种群下适应度函数值和迭代次数的关系, 如图 17 所示。

当种群为 80 时, 适应度函数值在 100 次迭代过程

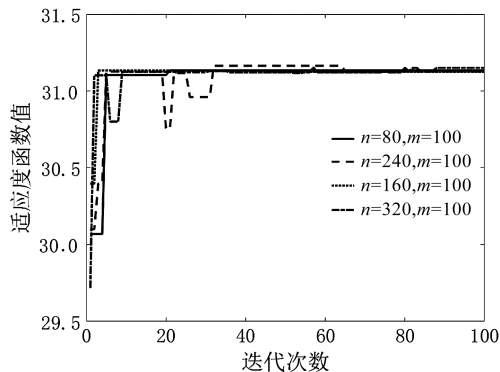


图 17 不同种群下适应度函数值和迭代次数的关系

中可以完全收敛, 而增加种群的数目到 160、240、320, 与种群为 80 时的收敛值几乎一致; 并且没有搜索到更优解, 只是略微增加了适应度函数的收敛速度。考虑到算法的效率和局部最优解, 将种群的大小设置为 80。

4.2 运行结果分析与对比

在具体固定了种群的大小和迭代的次数后, 就可以进行多次试验来求得最优解。

笔者进行了 30 次试验, 选取 4 次在 MATLAB 中的实验结果, 适应度函数值和迭代次数的关系曲线如图 18 所示。

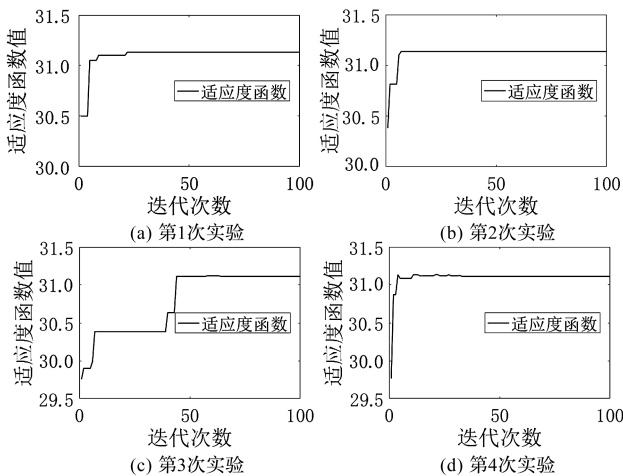


图 18 适应度函数值和迭代次数的关系曲线

从图 18 可以看出, 虽然初始种群的适应度函数值不一样, 但在迭代的过程中不断地选择、交叉、变异后, 适应度函数值都收敛于 31 与 31.5 之间的某个数。

笔者进行了 30 次试验, 选取 4 次在 MATLAB 中的实验结果, 目标函数值和迭代次数的关系图如图 19 所示。

在图 19 中可以发现与图 14 之间存在着联系, 因为在 3.3 节中设计适应度函数值时, 将目标函数值进行了转化, 即 $f' = 50 - f$, 所以适应度函数和目标函数

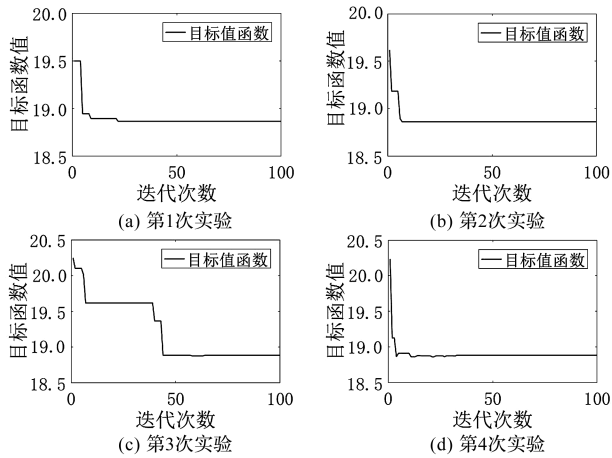


图 19 目标函数值和迭代次数的关系图

关于 $y = 25$ 对称。可以发现在迭代过程中,目标函数值收敛于 18.5 和 19 之间的某个数。

利用 MATLAB 中绘制传统遗传算法得到的工序分配方式,如图 20 所示。

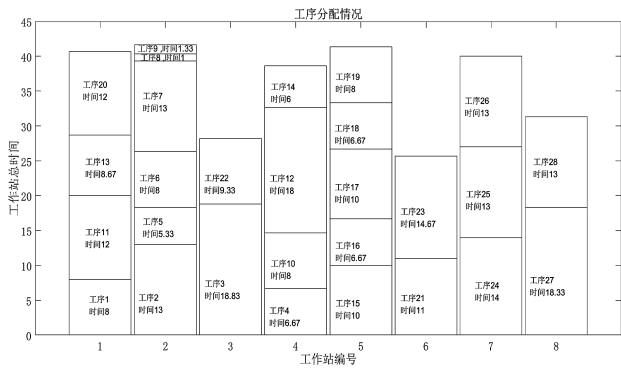


图 20 优化前各工序在 8 个工作站的分配柱状图

自适应遗传算法在 30 组实验中求得优化后的工序分配方式,如图 21 所示。

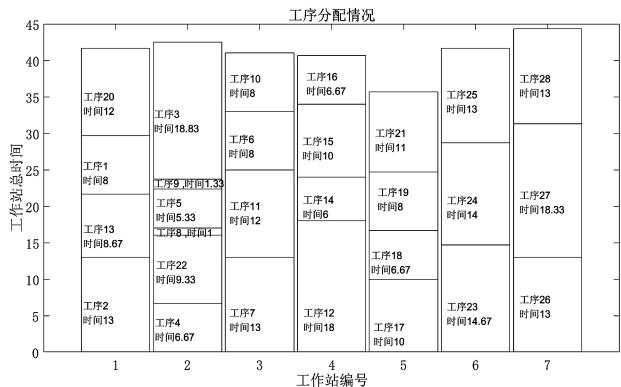


图 21 优化后各工序在 7 个工作站的分配柱状图

最优解显示最小工作站数 W_{real} 为 7,实际生产节拍 CT_{real} 为 44.33,装配线平滑指数为 2.47,目标函数值为 18.87,适应度函数值 f' 为 31.13,在规定时间内可以生产出 649 件产品,超出目标产量 49 件。

分别用传统遗传算法与自适应遗传算法来解决混流装配线平衡问题,得到优化前后的参数对比如表 3 所示。

表 3 优化前后混流装配线的参数对比

参数	工作 站数	实际 节拍	理论 节拍	平滑 指数	适应 度值	目标函 数值	目标 产量	实际 产量
遗传 算法	8	41.67	48	6.08	31.08	18.92	600	691
自适 应遗 传算法	7	44.33	48	2.47	31.13	18.87	600	649

由表 3 可知:

- (1) 减少了 1 个工作站,降低了工人与设备的费用;
- (2) 平滑指数降低了 59.36%,使得各个工作站之间的装配时间波动更小,装配作业的负荷更加均衡;
- (3) 实际生产节拍增加了 2.66 秒/件,使产量减少 42 件;
- (4) 在完成目标产量 600 件的基础上还降低了库存量 46.15%,降低了库存成本。

5 结束语

针对汽车电子液压制动控制单元的混流装配线平衡问题,笔者建立了装配线的多目标数学模型,利用 MATLAB 编写了改进后的自适应遗传算法,并对此进行了求解,通过调整参数设置得到了其最优解,最后将其与传统的遗传算法进行了对比分析。

研究表明:

- (1) 利用自适应遗传算法对汽车电子液压制动控制单元的混流装配线平衡进行优化,其效果优于传统遗传算法;
- (2) 优化后,减少了 1 个工作站,平滑指数降低了 59.36%,生产节拍增加了 2.66 秒/件,完成目标产量并降低库存,大大降低了混流装配线工人与设备的成本,提升了混流装配线的平衡率。

在后续的研究过程中,笔者将利用 NSGA-II 算法来解决多目标决策的问题,使得各个目标函数都尽可能能找到最优的解集,并将其应用于汽车电子液压制动控制单元的混流装配线平衡。

参考文献 (References):

[1] 姚午厚. 智能家居行业混流装配线平衡与排序方法研究 [D]. 哈尔滨: 哈尔滨工业大学, 2018.
 [2] 揣博君. 基于改进遗传算法的混流装配线多目标平衡研究 [D]. 哈尔滨: 东北林业大学, 2018.
 [3] 宋春雷. 混流装配线平衡与投产排序研究 [D]. 大连: 大

- 连理工大学,2012.
- [4] 张志文. 汽车座椅混合流水线平衡及仿真技术研究与应用[D]. 上海:上海交通大学,2015.
- [5] 常建娥,涂春慧,莫易敏,等. 汽车混流装配线工位复杂性研究[J]. 机械制造,2019,57(7):4.
- [6] 戴 韬,夏蓓鑫. 混流汽车液压电子制动控制单元装配线平衡问题设计与研究[J]. 计量与测试技术,2018,45(12):17-18.
- [7] 苏 平,于兆勤. 基于遗传算法和仿真分析的混合装配线平衡问题研究[J]. 计算机集成制造系统,2008,14(6):1121-1122.
- [8] 胡罗克,陈 勇,覃 涛. 线束装配线平衡改善与仿真[J]. 组合机床与自动化加工技术,2012,2:92-94.
- [9] BUKCHIN Y, RAVIV T. Constraint programming for solving various assembly line balancing problems [J]. **Omega**, 2018,78:59-61.
- [10] LI Z, KUCUKKOC I, NILAKANTAN J M. Comprehensive review and evaluation of heuristics and meta-heuristics for two-sided assembly line balancing problem[J]. **Computers & Operations Research**,2017,84:146-161.
- [11] RASHID M, TIWARI A, HUTABARAT W. Integrated optimization of mixed-model assembly sequence planning and line balancing using Multi-objective Discrete Particle Swarm Optimization [J]. **AI EDAM-ARTIFICIAL Intelligence for Engineering Design Analysis and Manufacturing**, 2019,33(3):333-335.
- [12] SAHU A, PRADHAN S K. Comparative analysis and optimization of Mixed Model assembly line using Genetic Algorithm [J]. **Material Today-Proceedings**, 2018, 5 (11): 25076-25079.
- [13] AKPINAR S, ELMI A, BEKTA? T. Combinatorial benders cuts for assembly line balancing problems with setups [J]. **European Journal of Operational Research**, 2017, 259 (2):529-532.
- [14] SUNGUR B, YAVUZ Y. Assembly line balancing with hierarchical worker assignment [J]. **Journal of Manufacturing Systems**, 2015,37:290-298.
- [15] DELICE Y, AYDO GAN E K, ÖZCAN U, et al. A modified particle swarm optimization algorithm to mixed-model two-sided assembly line balancing [J]. **Journal of Intelligent Manufacturing**, 2017,28(1):23-36.
- [16] 韩煜东,董双飞,谭柏川. 基于改进遗传算法的混装线多目标优化[J]. 计算机集成制造系统,2015,21(6):1476-1485.

[编辑:雷 敏]

本文引用格式:

蔡啟明,李伟伟. 电子液压制动控制单元的混流装配线平衡优化研究[J]. 机电工程,2022,39(1):77-86.

CAI Qi-ming, LI Wei-wei. Optimization of mixed-model assembly line balancing for hydraulic electronic control unit [J]. *Journal of Mechanical & Electrical Engineering*, 2022,39(1):77-86.

《机电工程》杂志;http://www.meem.com.cn

(上接第 64 页)

- [4] YAO J Y, JIAO Z X, SHANG Y X, et al. Adaptive nonlinear optimal compensation control for electro-hydraulic load simulator [J]. **Chinese Journal of Aeronautics**, 2010, 23 (6):725-731.
- [5] PLUMMER A R. Robust electrohydraulic force control [J]. **Proceeding of the IMechE, Part I: Journal of Systems and Control Engineering**, 2007, 221(4):719-730.
- [6] 高 强,朱玉川,王 睿,等. 高速开关阀控电液伺服加载系统自适应鲁棒控制 [J]. 航空动力学报,2019,34(2):505-510.
- [7] 裴忠才,王占林,王立国. 电液负载仿真台的理论分析 [J]. 北京航空航天大学学报,2000,26(6):658-661.
- [8] 郭 庆. 电液伺服非线性控制技术进展综述 [J]. 液压与气动,2018(3):3-8.
- [9] 成旭华,姚建勇,乐贵高. 电液伺服系统多模型鲁棒自适应控制 [J]. 西安交通大学学报,2018,52(11):158-161.
- [10] 焦宗夏,华 清,王晓东,等. 电液负载模拟器的复合控制 [J]. 机械工程学报,2002,38(12):35-39.
- [11] 邢宗义,张 媛,侯远龙,等. 电液伺服系统的建模方法研究与应用 [J]. 系统仿真学报,2009,21(6):24-26.
- [12] 姜继海,葛泽华,杨 晨,等. 基于微分器的直驱电液伺服系统离散滑模控制 [J]. 吉林大学学报,2018,48(5):1488-1495.
- [13] 吴振顺,付丙勤,冯玉宾,等. 自适应逆控制在电液伺服系统中的应用 [J]. 哈尔滨工业大学学报,2005,37(3):383-386.
- [14] PLETT G L. Adaptive inverse control of linear and nonlinear systems using dynamic neural networks [J]. **IEEE Transactions on Neural Networks**, 2003, 14 (2): 363-374.
- [15] XIE Y, MAHINDA V, TSENG K J, et al. Modeling and robust adaptive control of a 3-axis motion simulation [J]. **IEEE Transactions on Industry Applications**, 2001, 1 (1):555-561.
- [16] CHONG C S, ROZAIMI G, HAZRIQ I J, et al. Sliding mode controller design with optimized PID sliding surface using particle swarm algorithm [J]. **Procedia Computer Science**, 2017, 105(6):236-238.

[编辑:李 辉]