Journal of Sensor Technology and Application 传感器技术与应用, 2026, 14(2), 227-235
Published Online March 2026 in Hans. https://www.hanspub.org/journal/jsta
https://doi.org/10.12677/jsta.2026.142023

Hans汉斯

基于改进萤火虫算法的塑料激光焊接系统压力控制

彭鸿斌，张俊玲，涂　宇

江西理工大学电气工程与自动化学院控制工程专业，江西　赣州

收稿日期：2026年1月2日；录用日期：2026年1月26日；发布日期：2026年2月3日

摘　要

针对目前PID控制激光焊接系统的不足，本文尝试将FSO算法进行改进后加入塑料激光焊接系统中对于压力的控制部分，提高PID控制器的控制性能，实现更好地控制焊接过程中的压力。

关键词

塑料激光焊接，萤火虫算法，经验交换策略

Pressure Control of Plastic Laser Welding System Based on Improved Firefly Algorithm

Hongbin Peng, Junling Zhang, Yu Tu

Control Engineering Major, School of Electrical Engineering and Automation, Jiangxi University of Science and Technology, Ganzhou Jiangxi

Received: January 2, 2026; accepted: January 26, 2026; published: February 3, 2026

Abstract

Addressing the current shortcomings of the PID-controlled laser welding system, this paper attempts to enhance the FSO algorithm and incorporate it into the pressure control section of the plastic laser welding system, thereby improving the control performance of the PID controller and achieving better control over pressure during the welding process.

Keywords

Plastic Laser Welding, Firefly Algorithm, Experience Exchange Strategy

文章引用: 彭鸿斌, 张俊玲, 涂宇. 基于改进萤火虫算法的塑料激光焊接系统压力控制[J]. 传感器技术与应用, 2026, 14(2): 227-235. DOI: 10.12677/jsta.2026.142023

1. 引言

在当今社会，塑料已经成长为人类社会最重要的材料之一，当前，我国的塑料行业的产量约 8000 万吨，比 1949 年新中国成立时增长了 40 万倍。我国塑料加工行业的生产能力、实际产量、需求量、应用量、出口量目前在世界上均保持第一的位置[1]。在这种形势下，研究塑料相关技术的重要性不言而喻。

对于许多塑料制品而言，一次成型是几乎不可能实现的事情，而目前对于塑料而言主要以冶金连接技术(电阻焊、摩擦焊、电磁焊、扩散焊和超声波焊等)、机械连接技术(螺栓连接、自冲铆连接和压印连接等)或者粘接技术实现材料的有效连接[2]，在这三种中，机械连接是用紧固件将其连接，具有连接强度高的优点，但会增加零件重量，不适用于高轻量化需求零件的连接。黏合剂连接具有黏合表面应力分布均匀，不易产生变形等优点，然而也存在效率较低、连接强度低、易老化和环境适应性差等问题。焊接具有经济性好、操作简单、安全可靠等特点，适用于大规模的工业化生产，已经成为热塑性塑料零部件连接的主流加工方法[3]。目前，用于热塑性塑料组合件固接工艺中采用的焊接工艺技术主要有热板焊接、激光焊接、脉冲焊接以及超声波焊接等。其本质区别是热源不同[4]。目前来看，激光焊接技术在车灯的焊接领域属于主流的技术，拥有速度快，强度高，美观等多种优点。

而在激光焊接的过程中，压力的控制是非常重要的一部分，对于压力的控制直接决定了焊接的最终质量，在压力控制方面，PID 算法有着悠久的历史，考虑到 PID 参数调参较为复杂，使用经验中的数据往往无法实现更好的控制效果，因此，本文选择使用优化算法尝试寻找优秀的 PID 参数辅助控制。随着优化算法的不断发展，已经有越多越多的优化算法被提出来用于解决各类问题，比如由美国 Michigaan 大学的 Holland 教授提出的遗传算法[5]，由 R.Storn 和 K.Price 提出的差分进化算法[6]，由 Kennedy 和 Eberhart 提出的粒子群优化算法[7]等。萤火虫算法(FA)由英国学者 xin-she yang 教授提出(印度学者也提出过类似的萤火虫算法，被称作 GSO，这里选择使用 yang 教授的算法)，这个算法以自然界萤火虫为参考对象，具有原理直观，易于理解和实现；需要调节的参数较少，寻优能力强，等优点，适合应用于 PID 控制参数的优化中。

2. 算法简介

2.1. 萤火虫算法简介

Xin-She Yang 教授根据萤火虫的行为设计了以下几条主要的规则：

a. 萤火虫会根据自己所在位置的适应度发光。

b. 萤火虫所发出的光会随着距离而衰减，符合平方反比的规律。

c. 当萤火虫所接收到的来自某一只萤火虫发出的光大于自身发出的光时，萤火虫会朝着最亮的那只萤火虫的方向前进[8]。

萤火虫算法的主要思路是用搜索空间中的点模拟自然界中的萤火虫个体，将搜索和优化过程模拟成萤火虫个体的吸引和移动过程，将求解问题的目标函数度量成个体所处位置的优劣，将个体的优胜劣汰过程类比为搜索和优化过程中用好的可行解取代较差可行解的迭代过程[9]。

2.2. 萤火虫群算法简介

萤火虫群(FSO)算法在萤火虫算法(FA)的基础上，引入了性别机制，将萤火虫群分为雌性和雄性两部

分，每一只雄性萤火虫自带一些雌性的萤火虫，雌性萤火虫会围绕着自己的雄性萤火虫周围探索，而雄性萤火虫会基于附近最佳的雌性萤火虫和全局最优的雄性萤火虫更新自己的位置。

具体的算法流程为。

(1) 初始化阶段

系统会随机生成一定量的雄性萤火虫，并在雄性萤火虫附近的一个小区域中随机生成一定的雌性萤火虫，具体的生成数值根据情况修改。

(2) 第一阶段

雌性探索阶段中，对于每只雌性萤火虫，有

$$X^{n+1} = X^n + S_1 + S_2 \tag{1}$$

在这个式子中，X^{n+1} 是雌性萤火虫的新位置，X^n 则是萤火虫当前的位置，S_1 和 S_2 分别代表着雌性萤火虫的两个运动趋势。具体的公式如下。

$$S_1 = C_1\left(A^n - X^n\right) \tag{2}$$

$$S_2 = C_2\left(B^n - X^n\right) \tag{3}$$

C_1 和 C_2 是设定好的参数，决定了当前所属雄性萤火虫和随机选择的另一只雄性萤火虫对该雌性萤火虫的吸引力度，A^n 是当前雌性萤火虫所属的雄性萤火虫所处的位置，而 B^n 是随机选择的一只雄性萤火虫所处的位置，每次雌性萤火虫移动时，都会尝试重新随机选择一个 B^n。雌性萤火虫运动完后会尝试计算该雌性萤火虫的适应度函数 P，P 的计算方式将在后面给出。雌性萤火虫会持续运动共计 L_1 次。

当所有的雌性萤火虫运动完毕时，雄性萤火虫就会选择它所属的所有雌性萤火虫中适应度最高的一个，将它的适应度和雄性萤火虫本身的适应度进行对比，如果这个解更优秀，雄性萤火虫就会将这个位置作为它的新位置。

当雌性探索阶段结束，就开始进入雄性探索阶段。雄性萤火虫的位置更新公式为。

$$A^{n+1} = A^n + C_3\left(A^n - A_{best}^n\right) \tag{4}$$

C_3 和 C_1 C_2 一样，在算法运行前设定好，A_{best}^n 是所有的雄性萤火虫中的最优个体。当雄性探索阶段结束后，第一阶段会再次重复进行。

(3) 第二阶段

在第一阶段重复多次后，算法结束第一阶段，进入第二阶段。第二阶段中，算法抛弃了雌性萤火虫，专注于雄性萤火虫。

第二阶段中算法采用了传统萤火虫算法的吸引度机制，对于萤火虫 n 而言，他所接收到的来自萤火虫 a 的光照强度 T_a 由以下公式决定，P_a 是萤火虫 a 的适应度函数，γ 是吸引度系数，提前设定好。

$$T_a = \gamma \frac{P_a}{\sqrt{\left(x_a - x_n\right)^2 + \left(y_a - y_n\right)^2 + \left(z_a - z_n\right)^2}} \tag{5}$$

当萤火虫 n 受到的 T_a 大于 P_n 时，n 会选择受到光照最强的来源方向移动，移动公式为

$$A^{n+1} = A^n + C_4\left(A^n - A_a^n\right) + D \tag{6}$$

D 是随机额外扰动量，C_4 是提前设定好的运动系数。当萤火虫受到的所有光照都不如自身的适应度时，萤火虫会进行随机运动，随机运动的公式为。

$$A^{n+1} = A^n + D \tag{7}$$

3. 算法改进思路

在初始的 FSO 算法的基础上，进行了多次改进以尝试实现更好的优化效果，主要为 EES 经验策略，停滞重启，混沌初始化这几种。此外，为了节省运算成本，提高代码运行的效率和速度，还额外加入了并行计算的方案。

3.1. EES 经验策略

EES 经验策略是一种新兴的优化算法改进策略[10]，这个策略主要分为 esc，ecr，esh 三个阶段，三个阶段的划分和切换主要和当前迭代次数有关。三个阶段的划分与转换机制遵循原论文设计[10]，但是在具体的切换时间点上略有修改，当当前迭代次数 iter 与总迭代次数 S1max 比值低于 0.3 时，EES 经验策略处于第一阶段，比值位于 0.3 到 0.6 之间时，处于第二阶段，位于 0.6 之后，位于第三阶段。同时，在原策略的基础上增加了保险机制，当代码运行到中后期但经验池解过少时，允许强制切换为 esc 阶段进行进一步的积累。当代码结束 s1 阶段，进入 s2 阶段时，将锁定在 esh 阶段运行。

经验池的主要机制为

1) 保留记录整个算法运行中产生解，上限为 50 个，当达到上限时，会自动剔除经验池中适应度最低的解以替换成新的高适应度的解。

2) 相似度检测，为了提升经验的多样性，本文引入相似度检测的机制。当新收纳的解和已存在的解高度相似时(欧式距离小于 0.01%)，则只会保留其中适应度较高的解，适应度较低的不做保留。

3) 年龄检测，为了避免过于老旧的解的存在影响到 EES 经验策略的引导，本文加入了年龄检测的机制，当经验池中存在 30 个及以上的解时，对于过于老旧(已经经历的迭代次数超过 20 次)的解予以淘汰，特殊地，若是该解属于经验池中前 10%的优质解，那么它不会被淘汰。

esc 阶段是算法的初期，对于绝大部分的算法而言，esc 阶段的经验质量较差，没有足够的解用于引导种群，所以处于这个阶段时，这里主要以收集相对优质的解进入经验池，为之后做好准备。

第二个阶段是 ecr 阶段，进入到了 ecr 阶段后，经验交换策略在收集优质解的同时，也会发挥引导作用， 在这个阶段时，经验交换策略会在经验池中随机挑选两个解，然后生成引导向量，引导向量为带年龄权重的引导向量 1 和引导向量 2 的平均值，再加上少量的混沌扰动。

第三个阶段是 esh 阶段，这个阶段是经验交换策略主要工作的时间，在这个阶段中，经验交换策略开始对整个算法进行精英解的引导，精英解是在整个经验池中挑选出来的最多三个的历史最优解。经验引导向量 X_n 的计算公式为。

$$X_n = \frac{1}{3}X_a + \frac{1}{3}X_b + \frac{1}{3}X_c \tag{8}$$

3.2. 停滞重启机制

由于萤火虫算法容易陷入局部最优解的特性，这里选择加入停滞重启的机制，用于在一定程度上解决这个问题。在算法进行到中后期时，大量萤火虫会聚集到几个优质解附近进行探索，很容易误触发重启机制，因此重启机制将会在第一阶段结束后关闭。

停滞的检测主要有两个方面，即适应度值和参数值，当整个算法所有个体的最优适应度值与历史最优适应度值差值小于 5×10^{-7} 时，则认为适应度值陷入停滞；当所有个体平均的参数变化量小于 5×10^{-6} 时，则认为参数值陷入停滞。当这两个方面都陷入停滞状态持续超过十次迭代，且当前迭代的次数小于第一阶段最大迭代次数减 20 时(主要为了确保在重启后算法仍有足够的时间迭代)，将会进行重启。

重启会随机选择算法中 40%的个体，重启主要针对它们的雌性个体。对于这部分的雌性个体，其中

的 70%会在 EES 经验交换策略中的经验池中随机挑选一个解进行重启，12%会直接选择当前的历史最优解进行重启，18%会直接进行随机重启，用于探索新的区域，防止算法过早收敛。在选择好重启的位置后，这些个体还需要额外的小幅混沌扰动，以方便萤火虫探索。特殊的，如果停滞的过早，经验池中并未保留任何一个优质解，重启机制会将所有的雌性个体都进行随机重启。

3.3. 混沌初始化

为了提升算法的收敛速度和对整个解空间的遍历性，这里抛弃原始的经验起始点，选择使用混沌初始化起始点。在本算法中，采用的是 tent 混沌映射，tent 混沌映射的公式为。

$$x_{n+1}=\begin{cases}\mu x_n, & 0\le x_n<0.5\\ \mu\left(1-x_n\right), & 0.5\le x_n\le 1\end{cases} \tag{9}$$

在普通的混沌映射的基础上，还进行了一些改动，将整个算法的起始点分为三个部分，第一个部分以传统经验点为基础，辅以 0.3 强度的混沌映射。对于第二个部分的起始点，算法会使用混沌映射随机生成十个解，并将它们评估后放入 EES 策略的经验池，在经验池中选择解作为基础，辅以 0.5 强度的混沌映射。对于第三个部分的起始点，考虑到许多问题的最优解常常出现在解空间中相对中心的位置，所以选择解空间中 6.4% (即三个变量均中间 40%)的区域随机生成解，并辅以 0.7 强度的混沌映射。

3.4. FSO 算法运行流程

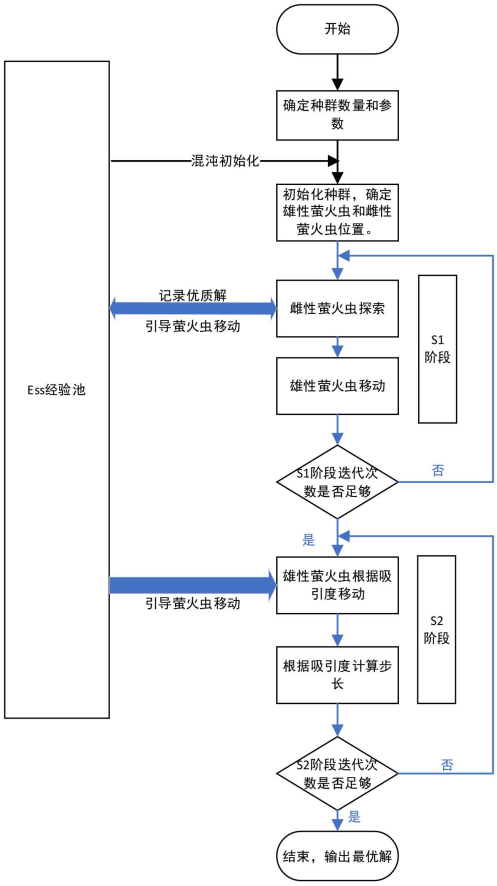

Figure 1. Flowchart of the improved FSO algorithm
图 1. 改进的 FSO 算法大致的运行流程图

在焊接过程中，对压力的控制需要比较稳定，超调量过大可能会导致瞬时压力过大，压坏产品，其次，调节时间决定了压力调节的速度，这对于提高焊接效率、保证焊缝一致性和适应高速连续生产至关重要。因此，整个算法的适应度函数 p 主要构成为：

ise (积分平方误差) = 0.15；

itae (时间乘绝对误差积分) = 0.20；

w_overshoot (超调量) = 0.25；

w_settling (调节时间) = 0.25；

w_rise (上升时间) = 0.15；

如果计算出来的 PID 参数无法稳定，则适应度函数直接设为极大值，此外，在适应度函数的计算之前，还会对 PID 参数做初步检测，对于 PID 参数过高的值(kp > 15, ki > 8, kd > 12)也会直接返回极大的适应度函数。

FSO 算法的大致流程如图 1 所示，省略了停滞重启机制等部分的内容。

4. 仿真实验结果分析

本次的仿真实验主要是在 matlab 的 Simulink 中进行，在对于上海某公司相关塑料激光焊接设备进行研究辨识得到的传递函数为

$$G(s)=\frac{3.625e^{-0.025s}}{s^2+0.81s+1} \tag{10}$$

使用未改进的 FSO 算法对该传递函数进行优化，重复试验五次，最终优化生成的适应度函数如表 1 所示。

Table 1. Results of the unimproved FSO algorithm run five times consecutively
表 1. 未改进的 FSO 算法连续运行五次的结果

次数	1	2	3	4	5
适应度函数	14.23	12.48	12.90	11.09	16.35

最优的适应度函数为 11.09，数据之间的方差为 3.955850。

改进后的 FSO 算法和原始的 FSO 算法，它们之间的对比如表 2 所示。

Table 2. Comparison between the improved FSO algorithm and the unimproved one
表 2. 改进后的 FSO 算法与未改进的对比

	1	2	3	4	5	方差
原 FSO 算法	14.23	12.48	12.90	11.09	16.35	3.955850
改进的 FSO 算法	9.91	9.72	9.73	9.65	9.53	0.019120

对比中可以清晰的看到，原 FSO 算法在面对这个问题时寻优能力尚可，但是不够稳定，需要多次运行才能找到相对合适的最优解，而经过改进后的 FSO 算法，不仅在寻优能力上有着进一步的突破，更重要的是依靠 EES 经验交换策略和混沌初始化，能够相对稳定地找到最优解，不容易陷入局部最优中。

FSO 算法(未改进)和 FSO 算法(改进后)函数评估次数对比和运行时间如图 2 和图 3 所示。

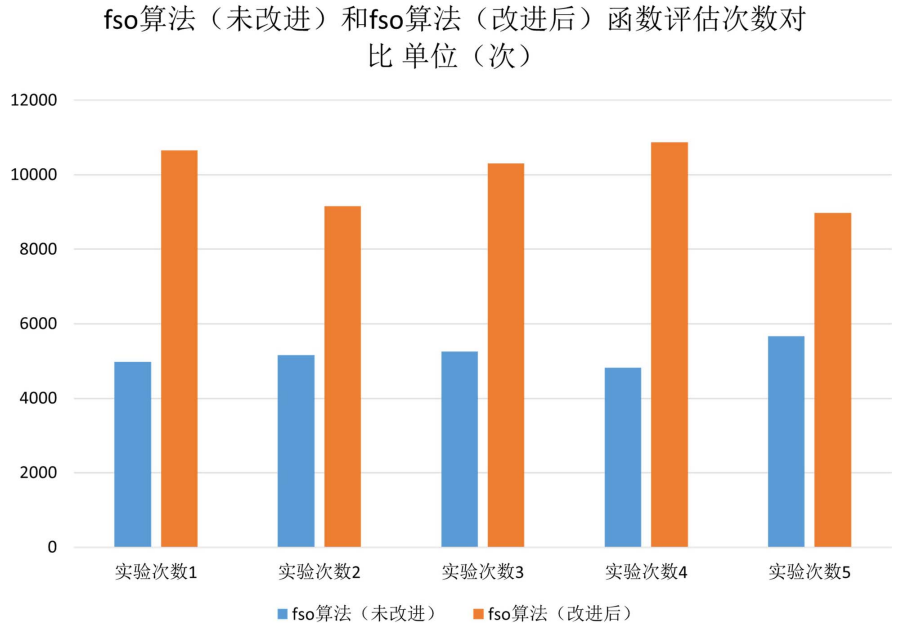

Figure 2. Comparison of the number of function evaluations between the improved FSO algorithm and the unimproved FSO algorithm
图 2. 改进的 FSO 算法和未改进的 FSO 算法之间的函数评估次数对比

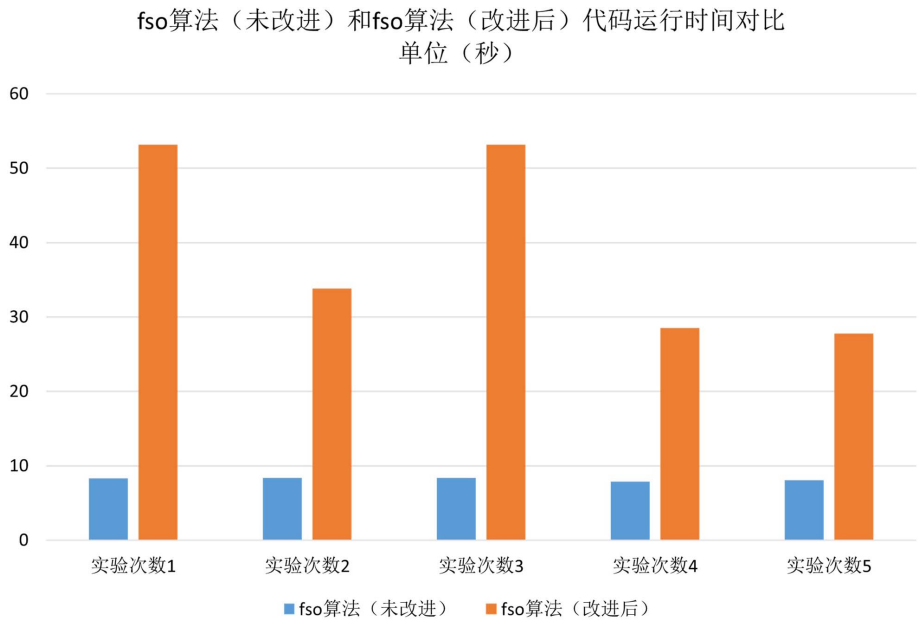

Figure 3. Comparison of code execution times between the improved FSO algorithm and the unimproved FSO algorithm
图 3. 改进的 FSO 算法和未改进的 FSO 算法之间的代码运行时间对比

可以看到，经过改进后的 FSO 算法无论是运行时间还是函数评估次数都远远高于初始版本的，但是处在完全可以接受的地步。

最后，将优化过的 FSO 算法和未优化过的 FSO 算法和传统经验整定出来的 PID 值进行仿真对比，仿真结果如图 4 所示。

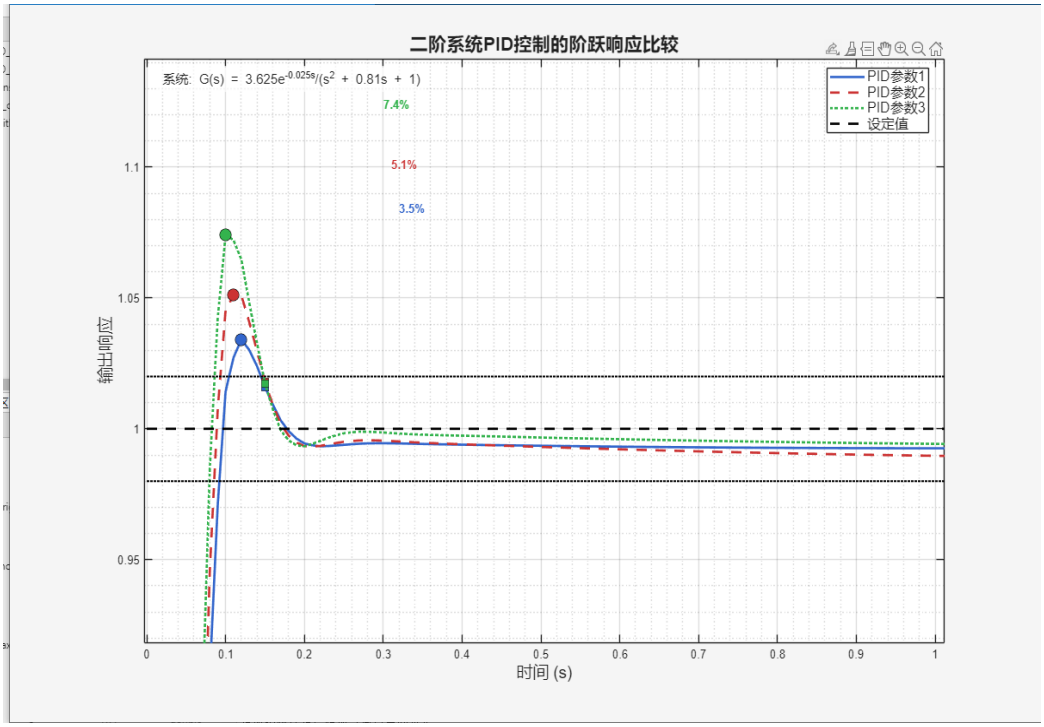

Figure 4. PID simulation comparison with three parameters
图 4. 三种参数进行 PID 仿真对比

图中，参数 1 为优化过的 FSO 算法多次运行中产生的最佳适应度解，参数 2 为普通 FSO 算法多次运行中产生的最佳适应度解，参数 3 为传统经验整定的 PID 参数。

可以看到，相比于普通 FSO 算法优化出来的 PID 参数，改进后的 FSO 算法在超调量方面降低了 29.41%，相比于传统经验整定出来的 PID 参数，在超调量方面降低了 52.7%。调节时间和响应速度有极小的降低但没有影响。由此可见，改进后的 FSO 算法在优化 PID 参数方面效果显著。

5. 结语

针对于塑料激光焊接系统中对于压力控制的需求，本文在 FSO 算法的基础上引入了 EES 经验交换策略和停滞重启机制、混沌初始化等改进，在 matlab 中进行运行过后，多次实验证明了无论是寻优能力还是算法的稳定性，都显著强于未改进的 FSO 算法。仿真出来的结果也显示改进后的 FSO 算法实现的 PID 控制远远强于经验整定出来的 PID 参数，因此，该算法在 PID 参数的优化中效果显著。

参考文献

[1] 刘学.《中国塑料》首届高分子材料技术创新与应用论坛上的新观点、新成果[J]. 中国塑料, 2024, 38(4): 124-128.
[2] 张越, 卢岩, 彭锐涛, 等. 轻量化材料新型连接工艺与应用现状[J]. 机械工程学报, 2024, 60(4): 259-283.
[3] 陈志, 张婉清, 颜昭君. 塑料激光透射焊接技术的研究动态和发展趋势[J]. 应用激光, 2020, 40(3): 556-563.
[4] 郭宗瑞, 张德华, 乔磊, 等. 超声波塑料焊接技术研究进展[J]. 新技术新工艺, 2024(8): 5-11.
[5] 边霞, 米良. 遗传算法理论及其应用研究进展[J]. 计算机应用研究, 2010, 27(7): 2425-2429+2434.
[6] 杨启文, 蔡亮, 薛云灿. 差分进化算法综述[J]. 模式识别与人工智能, 2008, 21(4): 506-513.
[7] 冯茜, 李擎, 全威, 等. 多目标粒子群优化算法研究综述[J]. 工程科学学报, 2021, 43(6): 745-753.
[8] 王晖, 王文君, 肖松毅. 萤火虫算法研究综述[J]. 南昌工程学院学报, 2019, 38(4): 71-77.

[9] 刘长平, 叶春明. 一种新颖的仿生群智能优化算法: 萤火虫算法[J]. 计算机应用研究, 2011, 28(9): 3295-3297.

[10] Jia, H. and Rao, H. (2025) Experience Exchange Strategy: An Evolutionary Strategy for Meta-Heuristic Optimization Algorithms. *Swarm and Evolutionary Computation*, **98**, Article 102082. https://doi.org/10.1016/j.swevo.2025.102082