

李徐, 张帆, 唐超, 等. 一种基于神经网络梯度下降的多目标优化算法[J]. 智能计算机与应用, 2024, 14(6): 224-229. DOI: 10.20169/j.issn.2095-2163.240634

一种基于神经网络梯度下降的多目标优化算法

李徐, 张帆, 唐超, 王文龙

(上海工程技术大学 机械与汽车工程学院, 上海 201620)

摘要: 针对多目标优化研究中各优化指标相互耦合相互牵制的问题, 提出一种基于神经网络梯度下降的多目标优化算法。该算法利用神经网络强拟合能力去模拟设计参数寻优的过程, 通过反向传播不断更新迭代神经网络, 并在这一过程中进行非支配解的判断。该算法通过一个刹车盘的优化问题进行了验证, 并和一种布谷鸟多目标优化算法进行了比较, 结果表明在求解边缘值上更具优势, 论文最后也指出了该算法的不足以及下一步的改进计划。

关键词: 多目标优化; 梯度下降; 神经网络

中图分类号: TP241

文献标志码: A

文章编号: 2095-2163(2024)06-0224-06

A multi-objective optimization algorithm based on neural network gradient descent

LI Xu, ZHANG Fan, TANG Chao, WANG Wenlong

(School of Mechanical and Automotive Engineering, Shanghai University of Engineering Science, Shanghai 201620, China)

Abstract: Aiming at the multi-objective optimization problem, a multi-objective optimization algorithm based on neural network gradient descent is proposed for the problem of coupling and pinning each other of optimization indicators. The algorithm uses the neural network excellent fitting ability to simulate the process of optimization of design parameters, continuously updates the iterative neural network through back propagation, and judges the nondominant solution in this process. The algorithm is verified by a brake disc optimization problem and compared with a cuckoo multi-objective optimization algorithm. The simulation results show that it is more advantageous in solving edge values. Furtherly, the shortcomings of the algorithm and the next improvement plan are also indicated in the end.

Key words: multi-objective optimization; gradient descent; neural networks

0 引言

多目标优化算法需要在处理目标冲突时找到一组能够平衡不同目标的解, 使得每个目标都能达到相对理想的状态。目前, 多目标优化算法主要分为基于进化算法、基于粒子群算法等多种类型, 这些算法都在一定程度上解决了多目标优化问题。下面将重点探讨几种目前应用较广泛的多目标优化算法。

遗传算法是多目标优化研究的典型算法之一, 其基本思想是利用生物遗传进化的方法, 在解空间中搜索最优解。Deb 等学者^[1]首次提出多目标遗传算法(MOGA), 通过生成和演化种群的方式, 寻找并保留帕累托最优解集合。此后, MOGA 也在不断地发展和改进, 例如 Tan 等学者^[2]提出了一种适用

于高维目标的基于取整的融合算子, 提升了性能。Deb 等学者^[3]提出了 NSGA-II 算法, 该算法能够在多目标问题中高效求解帕累托最优解集, 是进化算法中最为流行的一种。NSGA-II 算法基于非支配排序, 通过使用各项分布的概率密度函数追求帕累托最优前沿。NSGA-II 算法的缺点也比较明显, 主要是难以处理多约束问题, 并在处理高维问题时表现出较低效率。Reyes-Sierra 等学者^[4]提出 MOPSO 算法, 该算法是基于粒子群优化算法的改进, 通过将多个目标问题转化为一组彼此互不干扰的单目标问题, 并通过使用粒子群算法解决这些单目标问题, 以此达到多目标问题的最优解。Li 等学者^[5]提出 MOEA/D 算法, 该算法采用了交互式机制和集成策略等技术, 以此保证算法高效和解的优质性, 但使用

基金项目: 上海市科学技术委员会科技支撑医疗器械项目(17441901200)。

作者简介: 李徐(1997-), 男, 硕士研究生, 主要研究方向: 智能控制。

通讯作者: 张帆(1980-), 女, 博士, 副教授, 主要研究方向: 并联机器人, 医疗机器人, 先进制造技术。Email: pdssophia@qq.com

收稿日期: 2023-04-19

不同的编码方式可能会影响算法的性能和效果。Chen 等学者^[6]在对 MOEA/D 算法改进中利用机器学习技术,用网络结构对目标函数进行建模和优化,结果表明具有更高的搜索效率和可靠性。Yang 等学者^[7]提出一种布谷鸟多目标优化算法(MOCS),采用了自适应区搜索和多样化种群处理来增强算法的全局搜索能力和帕累托前沿集的收敛速度,并在压气机切削参数优化和冷却孔优化中,与其他3种最新的优化算法进行了比较,实验表明其性能效果更优,因此将其选作本文的比较算法,另外以上算法都是基于区域搜索的算法,当要求的设计精度较高时,其搜索效率会呈指数级下降。综合前文分析可知,如能设计一种算法可以通过梯度下降的方式进行多目标的寻优,则可解决上述问题。本文内容主要分为3部分:首先介绍了神经网络梯度下降以及参数更新原理,其次阐述了多目标优化算法原理,最后还进行了实验验证,得出了优化性能影响因素。

1 神经网络原理

神经网络是一种模拟人类神经系统的计算模型,由大量的人工神经元相互连接而成,可以通过学习和训练来实现各种复杂的任务,如图像识别、语音识别、自然语言处理等。神经网络的基本结构包括输入层、隐藏层和输出层。其中,输入层接收外部输入数据,隐藏层对输入数据进行处理和转换,输出层将处理后的结果输出如图1所示。神经网络的学习过程通常采用反向传播算法,通过不断调整神经元之间的连接权重,使得网络的输出结果与期望结果尽可能接近。神经网络的优点在于可以自动学习和适应数据的特征,无需手动设计特征提取器,同时具有较强的容错性和鲁棒性。

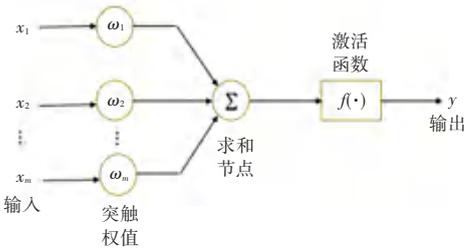


图1 单层神经网络模型

Fig. 1 Single-layer neural network model

网络的梯度传播和参数更新原理如下:

$$y = f\left(\sum_{i=1}^m w_i x_i\right) \quad (1)$$

神经网络的更新方式是通过反向传播,梯度下降的原理去更新神经网络的参数,其更新方式的数

学原理详见如下。

设输入样本为:

$$x(k) = (x_1(k), x_2(k), x_3(k), \dots, x_n(k)) \quad (2)$$

期望输出为:

$$d_o(k) = (d_1(k), d_2(k), d_3(k), \dots, d_n(k)) \quad (3)$$

隐藏层各神经元的输入输出如下:

$$h_{i_h}(k) = \sum_{i=1}^n w_{ih} x_i(k) - b_h \quad h = 1, 2, 3, \dots, p \quad (4)$$

$$ho_h(k) = f(h_{i_h}(k)) \quad h = 1, 2, 3, \dots, p \quad (5)$$

$$y_{i_o}(k) = \sum_{h=1}^p w_{ho} ho_h(k) - b_o \quad o = 1, 2, 3, \dots, q \quad (6)$$

$$yo_o(k) = f(y_{i_o}(k)) \quad h = 1, 2, 3, \dots, q \quad (7)$$

$$e = d_o(k) - yo_o(k) \quad (8)$$

求出期望值 $d_o(k)$ 与实际输出值 $yo_o(k)$ 的误差函数的偏导数 $\delta_o(k)$, 进一步推得如下系列公式。

(1) 输出层。运行计算公式如下:

$$\frac{\partial e}{\partial w_{ho}} = \frac{\partial e}{\partial y_{i_o}} \frac{\partial y_{i_o}}{\partial w_{ho}} \quad (9)$$

$$\frac{\partial y_{i_o}(k)}{\partial w_{ho}} = \frac{\partial\left(\sum_h w_{ho} ho_h(k) - b_o\right)}{\partial w_{ho}} = ho_h(k) \quad (10)$$

$$\begin{aligned} \frac{\partial e}{\partial y_{i_o}(k)} &= \frac{\partial\left(\frac{1}{2} \sum_{o=1}^q (d_o(k) - yo_o(k))^2\right)}{\partial y_{i_o}} = \\ &= -(d_o(k) - yo_o(k)) yo_o'(k) = \\ &= -(d_o(k) - yo_o(k)) f'(y_{i_o}(k)) = \\ &= \delta_o(k) \end{aligned} \quad (11)$$

(2) 隐藏层。运行计算公式如下:

$$\frac{\partial h_{i_h}(k)}{\partial w_{ih}} = \frac{\partial\left(\sum_h w_{ih} x_i(k) - b_h\right)}{\partial w_{ih}} = x_i(k) \quad (12)$$

$$\frac{\partial e}{\partial h_{i_h}(k)} = \frac{\partial\left(\frac{1}{2} \sum_{o=1}^q (d_o(k) - yo_o(k))^2\right)}{\partial h_{i_h}(k)} \times$$

$$\frac{\partial ho_h(k)}{\partial h_{i_h}(k)} = \frac{\partial\left(\frac{1}{2} \sum_{o=1}^q (d_o(k) - f(y_{i_o}(k)))^2\right)}{\partial h_{i_h}(k)} \times$$

$$\frac{\partial ho_h(k)}{\partial h_{i_h}(k)} = \frac{\partial\left(\frac{1}{2} \sum_{o=1}^q (d_o(k) - f\left(\sum_{h=1}^p w_{ho} ho_h(k) - b_o\right)\right)^2\right)}{\partial ho_h(k)} \times$$

$$\begin{aligned} \frac{\partial ho_h(k)}{\partial h_{i_h}(k)} &= - \sum_{o=1}^q (d_o - yo_o(k)) f'(y_{i_o}(k)) w_{ho} \frac{\partial ho_h(k)}{\partial h_{i_h}(k)} = \\ &= - \left(\sum_{o=1}^q \delta_o(k) w_{ho}\right) f'(h_{i_h}(k)) = \delta_h(k) \end{aligned} \quad (13)$$

利用输出层各神经元的 $\delta_o(k)$ 和隐藏层各神经

元的输出来修正链接权值 $w_{ho}(k)$:

$$\Delta w_{ho}(k) = -\mu \frac{\partial e}{\partial w_{ho}} = \mu \delta_o(k) h_{o_h}(k) \quad (14)$$

$$w_{ho}^{N+1} = w_{ho}^N + n\delta_o(k) h_{o_h}(k) \quad (15)$$

利用隐藏层各神经元的 $\delta_h(k)$ 和输入层各神经元的输入修正权连接:

$$\Delta w_{ih}(k) = -\mu \frac{\partial e}{\partial w_{ih}} = -\mu \frac{\partial e}{\partial h_i(k)} \cdot \frac{\partial h_i(k)}{\partial w_{ih}} = \delta_h(k) x_i(k) \quad (16)$$

$$w_{ih}^{N+1} = w_{ih}^N + n\delta_h(k) x_i(k) \quad (17)$$

其中, n 表示学习率。通过式(17)不断地迭代更新,可收敛到最优解的网络参数。

2 多目标优化算法原理

利用神经网络进行单目标优化已经在文献[8]中得到证明,因此为多目标优化算法提供了可行性支持。多目标优化问题是指在优化过程中需要同时考虑多个目标函数的最优化问题。这些目标函数通常是相互矛盾的,因此无法通过单一的优化方法来找到有效解决策略。多目标优化问题中除了多个目标函数的制约,还有一些边界函数的限制,设计参数的优化,不仅要满足设计参数在可选范围内的要求,还要满足边界函数的限制,基于以上思想,本文中基于梯度的多目标优化算法的基本思想是:在进行梯度下降最优化目标函数时,需要先满足边界函数的限制,其次再优化目标函数。

获得设计参数的策略网络的输入的维数和限制函数的个数相匹配,设边界函数满足条件时输入为1,反之为0。当边界函数不满足条件时就将以上一个输入作为本次输入,继续优化不满足条件的边界函数,当所有的边界函数都满足了条件后则优化目标函数。算法整体流程如图2所示。

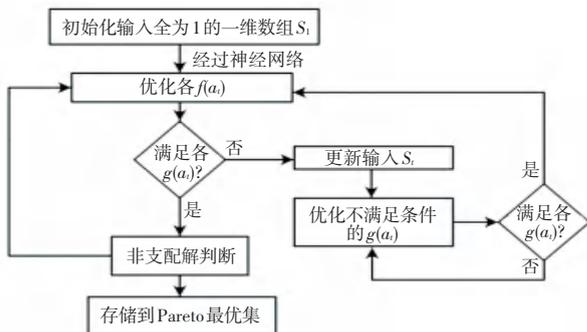


图2 算法流程图

Fig. 2 Flow chart of the algorithm

图2中, $f(a_i)$ 是各目标函数; $g(a_i)$ 是边界函数; a_i 是设计参数; S_1 是初始网络输入, 是一个全为1的一维数组; S_i 是优化过程中神经网络的输入。

3 优化实例验证

多盘式制动器的设计是多目标优化的另一个基准^[9-11]。目标是通过选择最佳设计变量来最小化总质量和制动时间: 制动盘的内径 r 、外径 R 、接合力 F 和摩擦表面 S 的数量。这是在扭矩、压力、温度和制动器长度等设计约束条件下进行的。研究推得该双目标设计问题可以写成如下形式。

(1) 最小化。可由式(18)~(19)来描述:

$$f_1(x) = 4.9 \times 10^{-5} (R^2 - r^2) (s - 1) \quad (18)$$

$$f_2(x) = \frac{9.82 \times 10^6 (R^2 - r^2)}{F_s (R^3 - r^3)} \quad (19)$$

(2) 边界函数。可由式(20)~(24)来描述:

$$g_1(x) = 20 - (R - r) \leq 0 \quad (20)$$

$$g_2(x) = 2.5(s + 1) - 30 \leq 0 \quad (21)$$

$$g_3(x) = \frac{F}{3.14(R^2 - r^2)} - 0.4 \leq 0 \quad (22)$$

$$g_4(x) = \frac{2.22 \times 10^{-3} F (R^3 - r^3)}{(R^2 - r^2)^2} - 1 \leq 0 \quad (23)$$

$$g_5(x) = 900 - \frac{0.026 6 F_s (R^3 - r^3)}{(R^2 - r^2)} \leq 0 \quad (24)$$

其中, 设计参数范围: $55 \leq r \leq 80, 75 \leq R \leq 110, 1\ 000 \leq F \leq 3\ 000, 2 \leq s \leq 20$ 。

在这项优化任务工作中, 设置了学习率是一个衰减过程, 其参数见表1。

表1 学习率参数

Table 1 Learning rate parameters

目标函数	初始学习率	衰减率	衰减步长	总步长
$f_1(x)$	0.01	0.5	5 000	30 000
$f_2(x)$	0.000 01	40.0	5 000	30 000

理论上, 2个待优化的目标函数在优化过程中是相互耦合的, 会相互牵制, 表现结果为求得的 $loss$ 曲线应该具有相反的走势, 如图3所示。

由图3可知, 整个优化算法会先倾向于优化 $f_1(x)$ 函数, 但总体上会随着 Pareto 前沿线开始向 $f_1(x)$ 函数优化, 这种图像走势符合 Pareto 前沿的分布状态, 及此消彼长的耦合关系。为了消除算法的偶然性, 进行了4次不同初始化网络参数值的优化, 其结果如图4所示。

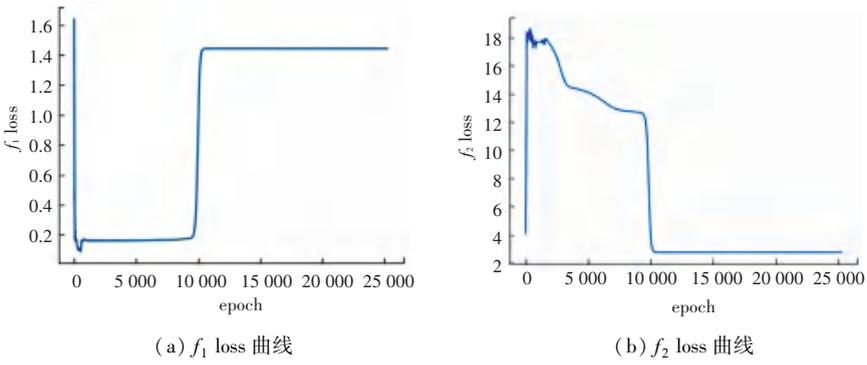


图 3 $f_1(x), f_2(x)$ loss 曲线
 Fig. 3 $f_1(x), f_2(x)$ loss curve

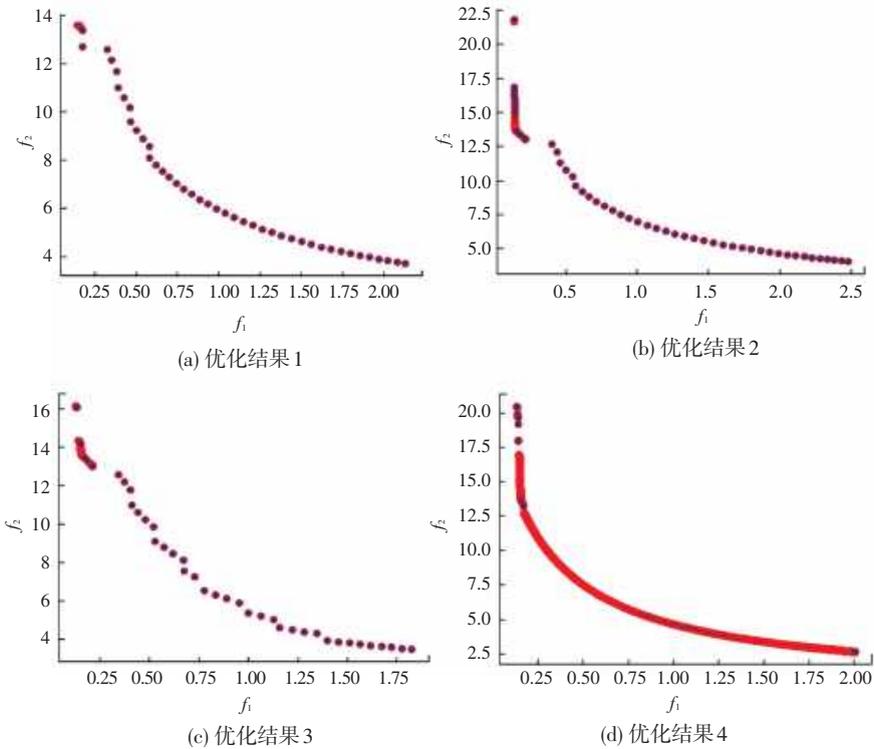


图 4 Pareto 前沿
 Fig. 4 Pareto fronts

由图 4 分析可知不同的初始化参数会影响优化结果,但是经过线性拟合,理论上其 Pareto 前沿可基本上一致。

4 影响因素

第 3 节中设计了一个具有衰减度的学习率,以及不同的初始化网络参数,对刹车盘进行了优化求解,研究结果也表明不同初始化网络参数会对优化求解有一定的影响,另外虽然得出了优化结果证明了其有效性,但对优劣程度却并未做出评价。基于以上因素设计了不同的学习率来验证其优化结果具有更精确的优势,对比算法是 MOCS 算法,先给出其优化结果如图 5 所示。这项工作中进行了 6 次特殊

的实验,每次优化迭代了 3 000 个步长,其学习率的参数见表 2。运行得到的优化结果如图 6 所示。图 6 中,侧重优化 $f_1(x)$ 见图 6(a),侧重优化 $f_2(x)$ 见图 6(b),同等重视见图 6(c)。

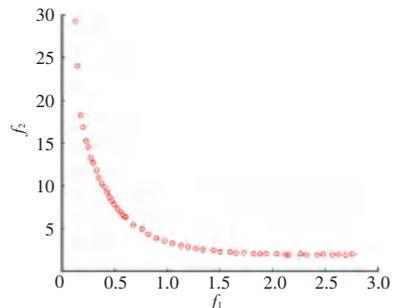


图 5 MOCS 算法的 Pareto 前沿^[7]
 Fig. 5 Pareto fronts of MOCS^[7]

表2 学习率参数

Table 2 Learning rate parameters

学习率类别	目标函数	侧重优化 $f_1(x)$	侧重优化 $f_2(x)$	同等重视
1	$f_1(x)$	0.050 0	0.000 1	0.050
	$f_2(x)$	0.000 1	0.050 0	0.005
2	$f_1(x)$	0.100 0	0.000 1	0.005
	$f_2(x)$	0.001 0	0.050 0	0.050

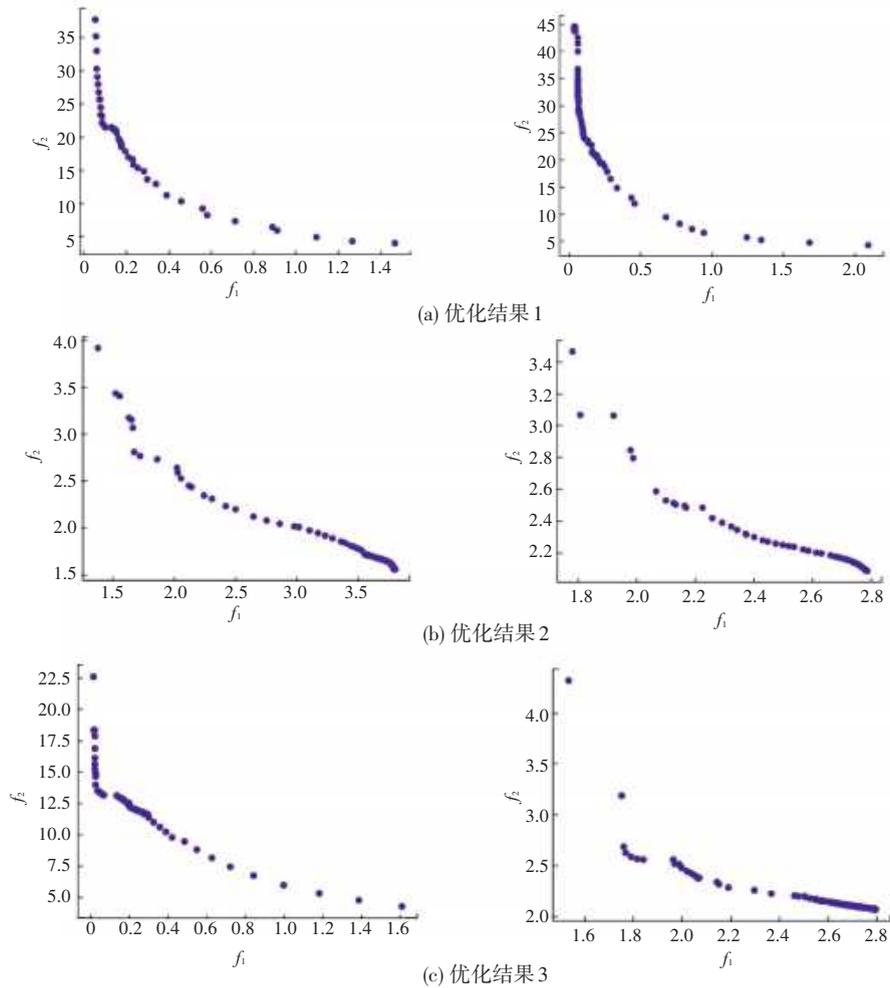


图6 运行后的优化结果

Fig. 6 Optimization results

由图6可知当增加 $f_1(x)$ 的学习率、降低 $f_2(x)$ 的学习率的时候,其优化结果更加偏向于 $f_1(x)$;当降低 $f_1(x)$ 的学习率、增加 $f_2(x)$ 的学习率的时候,其优化结果更倾向于 $f_2(x)$;当两者的学习率相差不大的时候,其优化侧重度被两者均摊呈现各目标函数的值分布在解集的中间位置。对比MOCS算法可知,本算法的优势在于更容易获取到边缘位置的优化值,这一点可以从 $f_1(x)$ 、 $f_2(x)$ 最大值分布的位置看出。设计者可以从经验出发,选择侧重优化的目标函数来获取理想的目标设计参数,从而增加

了在设计时的选择性。

5 结束语

本文利用神经网络去拟合设计参数的优化策略,通过梯度下降的原理,以求极值的方式求解最优解。且在求极值的过程中是一种线性下降的方式,因此对比于全局搜索内算法,其搜索效率在一定程度上得到了提高。仿真结果也表明了多目标优化的可行性,以及在求解边缘值方面的优越性。但是,同时也会看到,研究中需要调节学习率。因此在下一

步工作中,会将学习率的调整这一步融合成一个多目标优化算法,理论上这种方式可以一次性得到更优的 Pareto 前沿解集。

参考文献

- [1] DEB K, AGRAWAL S. Simulated binary crossover for continuous search space[J]. *Complex Systems*, 1995,9(2):115-148.
- [2] TAN Y, WANG X. A hybrid integer-coded MOGA for high-dimensional problems [J]. *Computational Optimization and Applications*, 2016,63(1): 73-94.
- [3] DEB K, PRATAP A, AGARWAL S, et al. A fast and elitist multi objective genetic algorithm: NSGA-II[J]. *IEEE Transactions on Evolutionary Computation*, 2002,6(2):182-197.
- [4] REYES-SIERRA M, COELLO C C. Multi-objective particle swarm optimizers: A survey of the state-of-the-art [J]. *International Journal of Computational Intelligence Research*, 2006,2(3): 287-308.
- [5] LI Z, YANG S, TIAN Y, et al. Multi-objective evolutionary algorithm based on decomposition [J]. *IEEE Transactions on Evolutionary Computation*, 2009,13(5): 941-958.
- [6] CHEN J, ZHANG L, PENG W, et al. Multi-objective evolutionary algorithm based on deep reinforcement learning [J]. *Information Sciences*, 2019,478: 278-297.
- [7] YANG Xinshe, DEB S. Multi objective cuckoo search for design optimization [J]. *Computers & Operations Research*, 2013, 40(6):1616-1624.
- [8] GONG Wenyin, CAI Zhihua, ZHU Li. An effective multi objective differential evolution algorithm for engineering design [J]. *Structural and Multidisciplinary Optimization* 2009, 38(2): 137-157.
- [9] OSY CZKA A, KUNDU S. A genetic algorithm-based multi criteria optimization method [C]// *Proceedings of First World Congress on Structural and Multi-disciplinary Optimization*. Goslar, Germany :Elsevier Science,1995:909-914.
- [10] RAY L, LIEW K M. A swarm metaphor for multiobjective design optimization [J]. *Engineering Optimization*, 2002, 34(2): 141-153.
- [11] 李徐,张帆. 受强化学习思想启发的一种结构优化算法[J]. *智能计算机与应用*,2022,12(5):136-140.