

基于自适应控制参数的改进水波优化算法

刘 翱^{1,2} 邓旭东¹ 李维刚^{3,4}

(武汉科技大学管理学院 武汉 430081)¹

(智能信息处理与实时工业系统湖北省重点实验室 武汉 430065)²

(武汉科技大学信息科学与工程学院 武汉 430081)³

(冶金工业过程系统科学湖北省重点实验室 武汉 430081)⁴

摘 要 水波优化算法(Water Wave Optimization, WWO)是最近被提出的一种新型的群智能优化算法。它尽管具有控制参数少、操作简单、容易实现等优点,但是也存在收敛较慢、搜索精度低等不足。针对水波优化算法的不足,首先,从理论上分析并揭示算法收敛时控制参数应满足的条件;然后,提出满足上述条件的改进水波优化算法,改进算法采取自适应机制来调节算法参数,进一步增强了全局探索和局部开发的平衡能力;最后,对4种算法(ApWWO, WWO, FA, MVO)在10个标准测试函数上的寻优性能进行仿真实验和统计比较。结果表明,ApWWO在搜索精度、速度和鲁棒性等方面均显著优于WWO和FA,在5个测试函数上优于MVO;与PSO和GA的对比结果表明,ApWWO具有较好的寻优性能。进一步分析了维数和种群规模对ApWWO的影响,并使用ApWWO来求解置换流水线调度问题,结果表明ApWWO能够取得较好的求解效果。

关键词 进化算法,水波优化算法,自适应控制参数,置换流水线调度

中图分类号 TP183 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.07.036

Improved Water Wave Optimization Algorithm with Adaptive Control Parameters

LIU Ao^{1,2} DENG Xu-dong¹ LI Wei-gang^{3,4}

(School of Management, Wuhan University of Science and Technology, Wuhan 430081, China)¹

(Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial System, Wuhan 430065, China)²

(College of Information Science and Engineering, Wuhan University of Science and Technology, Wuhan 430081, China)³

(Hubei Province Key Laboratory of Systems Science in Metallurgical Process, Wuhan 430081, China)⁴

Abstract Water wave optimization algorithm(WWO) is a novel population-based optimization algorithm. Despite of its advantages with few controllable parameters, simple operations, and easy implementation, it still risks slow convergence rate and low search precision. To mitigate the aforementioned risks, firstly, a concise yet powerful theoretical analysis was carried out to derive the convergence condition that the control algorithm parameters should satisfy. Then, an improved WWO was proposed to meet the above conditions by incorporating an adaptive algorithm parameters tuning strategy, and it's expected to further enhance the capability of balancing between global exploration and local exploitation via this strategy. Finally, simulation experiments and statistical comparisons between four algorithms(ApWWO, WWO, FA, MVO) on 10 benchmark functions were conducted. The results show that ApWWO performs significantly better than WWO and FA in terms of search accuracy, speed and robustness, as well as outperforms MVO in five test function. Compared with PSO and GA, ApWWO can yield good performance, which can be affected by the problem's dimension and population size, and ApWWO also performs well on the permutation flow shop scheduling problem.

Keywords Evolutionary algorithm, Water wave optimization algorithm, Adaptive controlling parameter, Permutation flow shop scheduling

到稿日期:2016-06-16 返修日期:2016-10-18 本文受国家自然科学基金(11271356),教育部人文社会科学研究青年基金项目(16YJ CZH056),湖北省教育厅人文社会科学研究青年项目(17Q034),智能信息处理与实时工业系统湖北省重点实验室开放基金(2016znss18B),冶金工业过程系统科学湖北省重点实验室(武汉科技大学)开放基金(Z201501),武汉科技大学青年科技骨干培育计划项目(2016xz017)资助。

刘 翱(1987—),男,博士,讲师,CCF会员,主要研究方向为智能计算、优化调度,E-mail:liuao@amss.ac.cn;邓旭东(1956—),男,硕士,教授,主要研究方向为系统工程,E-mail:dengxudong@wust.edu.cn;李维刚(1978—),男,博士,教授,主要研究方向为工业复杂系统建模、控制与优化,E-mail:liweigang.luck@foxmail.com(通信作者)。

1 引言

优化问题因在工程实践中广泛应用而备受研究人员和工程人员的关注。求解优化问题的算法通常包括两类:1)经典的数学规划方法;2)受自然、社会、物理等现象的启发而设计的群智能优化算法^[1]。后者通常是随机搜索算法,能够有效求解复杂的优化问题。经典的群智能优化算法包括遗传算法^[2]、粒子群优化^[3]、蚁群优化^[4]等。

近年来,诸如萤火虫算法、蝙蝠算法、布谷鸟搜索算法、生物地理学优化算法、水波优化算法等新兴的群智能优化算法不断涌现。与经典的群智能优化算法相比,这些算法通过设计独特的搜索机制来获得优良的搜索性能,更好地平衡了全局探索和局部开发,因此被越来越多的研究人员关注并广泛应用于工程实践问题。例如,田梦楚等^[5]研究了基于萤火虫算法的粒子滤波问题。Rahmani等^[6]利用蝙蝠算法对机器人进行鲁棒自适应控制,结果表明基于蝙蝠算法的机器人控制具有较好的鲁棒性。贾云璐等^[7]利用种群特征对布谷鸟搜索算法进行反馈改进,其优化效果优于改进前的布谷鸟搜索算法。Wang等^[8]结合GOBL和NEH等策略来改进标准布谷鸟搜索算法,并将其应用于流水车间调度问题的求解。Lin等^[9]则通过引入局部搜索、NEH等策略研究了混合离散型生物地理学优化算法,并将其应用到分布式装配流水线调度问题,结果表明融合混合策略的生物地理学优化算法能较好地求解分布式装配车间流水调度问题。

水波优化算法(Water Wave Optimization, WWO)是受浅水波理论的启发而设计的一种新型的群智能优化算法^[10]。该算法通过模拟水波的传播、折射、碎浪等运动方式来实现对高维空间的搜索寻优,具有可调参数少、操作简单、易于实现等优点。

2015年Zheng^[10]首次提出了水波优化算法,并利用IEEE CEC 2014基准测试问题集对水波优化的寻优性能进行了测试,结果表明水波优化的性能表现良好,优于IWO, BBO和GSO等进化算法;同时也将它应用到高铁列车调度问题中,结果表明WWO的效果优于CPSO, GSA, HuS。随后,张蓓等^[11]研究了水波优化算法的收敛性,从理论上分析了执行传播操作和折射操作时算法的收敛条件,并利用数值实验验证了上述两种收敛条件的正确性。进一步,郑宇军等^[12]研究了软件形式化开发关键部件的选取问题,建立了该问题的0-1约束规划模型,并针对该问题设计了离散水波优化算法来求解大型软件系统的应用实例,结果表明离散水波优化算法具有可行性且其性能优于GA, BPSO, IBPSO和BDE等算法。

尽管已有的研究结果初步验证了WWO算法的可行性和优势,但如文献^[10]所指出,WWO算法是新近提出的,仍存在收敛较慢、精度不高等不足,在理论分析和算法性能上仍有改进空间。

鉴于此,本文首先从理论上分析并揭示算法收敛时控制

参数应满足的条件;然后结合自适应参数更新机制,提出了满足上述条件的改进水波优化算法,改进算法在搜索早期能更好地执行全局探索,而在搜索后期能更多地进行搜索局部开发,进一步增强了全局探索和局部开发的平衡能力。

2 优化问题

不失一般性,将优化问题定义为:

$$\begin{aligned} \max f(x) \\ \text{s. t. } x_i \in [Lb_i, Ub_i], i=1, 2, \dots, n \end{aligned} \quad (1)$$

其中, n 为变量维数, $x=(x_1, x_2, \dots, x_n)$ 为 n 维决策变量。

3 水波优化算法概述

3.1 基本原理

水波优化算法(Water Wave Optimization, WWO)通过模拟水波运动来求解优化问题。以最大化问题为例,根据文献^[10]所述,该算法将优化问题的搜索空间类比为海床,每个解对应于一个水波,水波具有波高 h 和波长 λ ,水波的适应度值与其到海床的垂直距离成反比:离海平面越近的解越优,对应的水波能量越高, h 越大, λ 越小(见图1),这就保证了算法在较小范围内搜索较优解,在较大范围内搜索较差解,最终促使水波群不断向更优的解靠近。

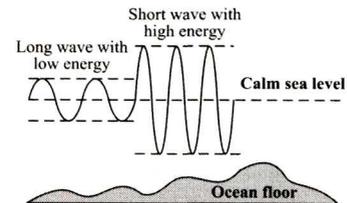


图1 水波优化算法示意图^[1]

在WWO算法中,初始化时设定每个水波的波高和波长分别为 $h=h_{\max}, \lambda=0.5$,其中 h_{\max} 为整数常量。在进化过程中,WWO执行3种操作:传播、折射和碎浪。

3.2 传播

在WWO算法的每代进化中,水波 x 都按如下方式执行一次传播操作,传播后的解被更新为 x' :

$$x'_d = x_d + \text{rand}_{[-1,1]} \cdot \lambda L_d, d=1, 2, \dots, n \quad (2)$$

其中, $\text{rand}_{[-1,1]}$ 为 $[-1,1]$ 内均匀分布的随机数, L_d 表示第 d 维变量的搜索宽度。若更新后的位置 x' 超过搜索范围,则将其随机重置为搜索范围内的一个位置。

如果 $f(x') > f(x)$,则用 x' 替代 x ,重置波高 $h=h_{\max}$;否则,保留 x ,更新波高 $h=h-1$ 。

每次进化后,按如下方式更新水波 x 的波长 λ :

$$\lambda = \lambda \alpha^{-(f(x) - f_{\min} + \epsilon) / (f_{\max} - f_{\min} + \epsilon)} \quad (3)$$

其中, f_{\max}, f_{\min} 分别为水波群中的最大适应度值和最小适应度值, α 是波长的衰减系数, ϵ 是很小的正数(避免分母为0)。

3.3 折射

当水波 x 传播多次而没有得到改善时,其波高将递减为 $h=0$,按如下方式对其执行折射操作以避免搜索停滞。

$$x_d' = x_d^* + N\left(\frac{x_d^* + x_d}{2}, \left|\frac{x_d^* - x_d}{2}\right|\right), d=1, 2, \dots, n \quad (4)$$

其中, x^* 表示当前的最优解, $N(\mu, \sigma^2)$ 表示均值为 μ 、方差为 σ^2 的高斯随机数。

折射后重置波高 $h = h_{\max}$, 波长更新为:

$$\lambda^{new} = \lambda \frac{f(x)}{f(x')} \quad (5)$$

3.4 碎浪

当水波能量持续增加时, 水波的波峰越来越陡峭, 直至破碎成一连串的孤立波。WWO 算法一旦搜索到新的最优水波 x^* , 则按如下方式执行碎浪操作: 随机选择 $k \in [1, k_{\max}]$; 对每一维进行搜索得到 k 个孤立波 x' :

$$x_d' = x_d^* + N(0, 1) \cdot \beta L_d, d=1, 2, \dots, k \quad (6)$$

其中, β 是碎浪系数。如果 k 个孤立波的适应度值都低于原水波 x^* , 则保留 x^* ; 否则用一个最优水波替代 x^* 。

4 改进水波优化算法

WWO 算法通过执行传播、折射和碎浪操作来进行搜索。可以预见, 随着进化次数的增加, 水波群 x 收敛到最优水波 x^* 。因此, 以下等式成立:

$$\lim_{t \rightarrow \infty} X_i(t) = \lim_{t \rightarrow \infty} X_j(t), \forall i \neq j \quad (7)$$

$$\lim_{t \rightarrow \infty} X_i(t+1) = \lim_{t \rightarrow \infty} X_i(t) \quad (8)$$

其中, $X_i(t)$ 表示第 t 代水波群中的第 i 个水波。

4.1 衰减系数 α 的取值范围

α 是波长的衰减系数, 根据式(3)可知水波的适应度值越高, 波长越短, 其搜索范围越小; 反之, 搜索范围越大。

令 $L = L_d$, r 为随机数, \bar{f} 是前 t 项 $(f(X(t)) - f_{\min}(t) + \epsilon) / (f_{\max}(t) - f_{\min}(t) + \epsilon)$ 之和的平均值。

根据式(2)、式(7)、式(8)可知:

$$\lim_{t \rightarrow \infty} X_i(t+1) = \lim_{t \rightarrow \infty} (X_i(t) + rand \cdot \lambda(t)L) \Rightarrow \lim_{t \rightarrow \infty} \lambda(t) = 0 \quad (9)$$

进一步, 根据式(3)可知:

$$\lambda(t) = \lambda(0)\alpha^{-\bar{f}}, \lim_{t \rightarrow \infty} \lambda(t) = 0 \Rightarrow \lim_{t \rightarrow \infty} \alpha^{-\bar{f}} = 0 \quad (10)$$

由于 $0 < \bar{f} \leq 1$, 为保证式(10)在任何情况下都成立, 衰减系数 α 应满足:

$$\alpha > 1 \quad (11)$$

文献[10]的研究表明, 当 $\alpha \in [1.001, 1.01]$ 时, WWO 算法的优化效果较好。

4.2 碎浪系数 β 的自适应控制策略

β 代表碎浪系数, 用于控制在最优水波附近密集搜索孤立子波的范围。 β 越小, 搜索范围越小; 反之, 搜索范围越大。

根据式(6)一式(8), 可知:

$$\lim_{t \rightarrow \infty} X_i(t+1) = \lim_{t \rightarrow \infty} (X_i(t) + rand \cdot \beta(t)L) \Rightarrow \lim_{t \rightarrow \infty} \beta(t) = 0 \quad (12)$$

由式(12)可知, 为保证 WWO 算法的收敛性, β 应该逐渐趋近于 0。直观上看, 若 β 过大, 则难以保证算法收敛; 若 β 过小, 又限制了搜索精度, 难以更好地执行全局探索。

合理的碎浪系数 β 应该满足:

- 1) 在进化早期, 较大的碎浪系数能更好地执行全局探索;
- 2) 在进化后期, 较小的碎浪系数能更多地局部开发, 进一步增强全局探索和局部开发的平衡能力。

与原始 WWO 算法使用固定碎浪系数不同, 改进 WWO 算法 (Adaptive Water Wave Optimization, ApWWO) 采取如下策略来自适应地控制碎浪系数 β (见图 2)。

$$\beta(t) = \beta_{\max} \cdot \left(\frac{\beta_{\min}}{\beta_{\max}}\right)^{\frac{t}{T_{\max}}} \quad (13)$$

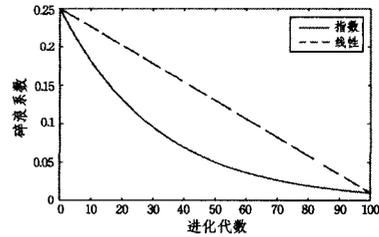


图 2 自适应碎浪系数

如图 2 所示, 式(13)所示的自适应碎浪系数比线性碎浪系数下降得更缓慢, 能更好地平衡进化过程中算法的全局探索和局部开发。

ApWWO 算法基本步骤的描述如下:

Step 1 随机初始化一个规模为 n 的水波群, 计算每个水波 x 的适应度值 $f(x)$, 并找到最优水波 x^* 。

Step 2 若满足终止条件, 则算法结束, 并返回最优水波 x^* 。

Step 3 对每个水波 x 执行传播、折射、碎浪操作:

Step 3.1 对水波 x 执行传播操作, 得到一个新波 x' 。

Step 3.2 若 $f(x') > f(x)$, 执行 Step 3.2.1—Step 3.2.2, 否则转 Step 3.3:

Step 3.2.1 若 $f(x') > f(x^*)$, 按式(6)和式(13)执行碎浪操作, 用 x' 替代 x^* ;

Step 3.2.2 用 x' 替代 x 。

Step 3.3 将 x 的 h 减 1; 若 $h = 0$, 则对 x 执行折射操作。

Step 3.4 按式(3)更新所有水波的波长。

Step 4 计算每个水波 x 的适应度值 $f(x)$, 更新最优水波 x^* , 转 Step 2。

5 仿真实验与分析

5.1 测试函数

为验证改进水波优化算法 (ApWWO) 的寻优性能, 本文对 10 个标准测试函数^[13-14]进行测试, 并将其与 WWO, Firefly Algorithm(FA)^[15], Multi-Verse Optimizer(MVO)^[16]进行对比。

F1 和 F2 为单峰函数, 具有一个全局极值点; F3—F6 是多峰函数, 具有多个局部极值点; F7—F10 是 F1—F3 和 F5 经过平移和旋转后的多峰函数。

F1—F6 的信息和形态参见表 1 和图 3。

表1 标准测试函数 F1-F10

编号	函数	公式	特征	范围	理论最优值
F1	Sphere	$f_1(x) = \sum_{i=1}^n x_i^2$	单峰	$[-100, 100]$	0
F2	Rosenbrock	$f_2(x) = \sum_{i=1}^n [100(x_{i+1} - x_i)^2 + (x_i - 1)^2]$	单峰	$[-30, 30]$	0
F3	Rastrigrin	$f_3(x) = \sum_{i=1}^n (x_i^2 - 10\cos(2\pi x_i) + 10)$	多峰	$[-5.12, 5.12]$	0
F4	Ackley	$f_4(x) = -20(\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2})) - \exp(\frac{1}{n}\sum_{i=1}^n \cos(2\pi x_i)) + 20 + e$	多峰	$[-32, 32]$	0
F5	Griewank	$f_5(x) = -\frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(2\pi \frac{x_i}{\sqrt{ f_i}}) + 1$	多峰	$[-600, 600]$	0
F6	Shaffer	$f_6(x) = \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2} - 0.5$	多峰	$[-100, 100]$	-1
F7	Shifted Sphere	$f_7(x) = f_1(z) - 180, z = M(X - O), M^T M = I$	复合	$[-100, 100]$	-450
F8	Shifted Rosenbrock	$f_8(x) = f_2(z) - 180, z = M(X - O), M^T M = I$	复合	$[-30, 30]$	390
F9	Shifted Rastrigrin	$f_9(x) = f_3(z) - 180, z = M(X - O), M^T M = I$	复合	$[-5.12, 5.12]$	-330
F10	Shifted Griewank	$f_{10}(x) = f_5(z) - 180, z = M(X - O), M^T M = I$	复合	$[-600, 600]$	-180

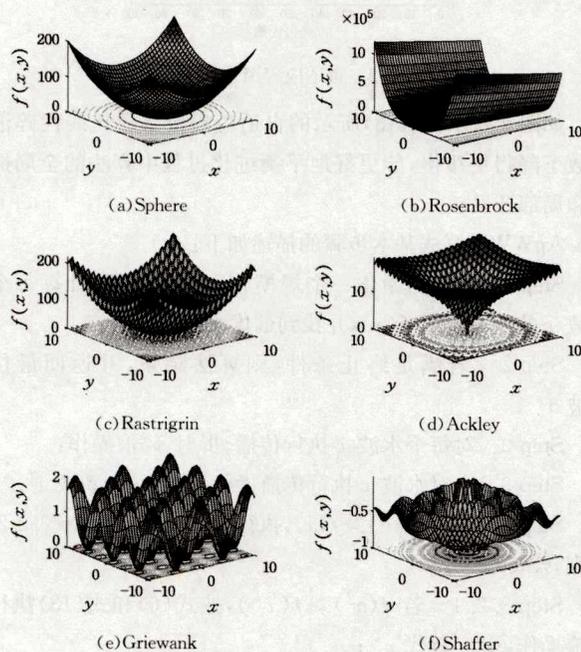


图3 F1-F6 函数形态

5.2 实验环境

实验环境如下: Window 7, CPU 奔腾 T4400, 主频 2.2 GHz, 内存为 2GB, 编程语言为 Matlab 2014b.

5.3 参数设置

为保证结果的公平性及客观性, 设 WWO, ApWWO, FA 和 MVO 等的种群规模为 10, 最大函数评价次数为 50000.

1) WWO 和 ApWWO 的参数来源于文献[10], 参数设置为 $\lambda_0 = 0.5, h_{\max} = 6, \alpha = 1.003, \beta \in [0.001, 0.25]$;

2) FA 和 MVO 的算法参数来源于文献[15]和文献[16];

3) 每个算法在 F1-F10 上分别独立运行 30 次.

5.4 结果分析

为了全面地评价 ApWWO 算法的性能, 接下来将从平均适应度值、标准差和非参数检验等方面来比较 ApWWO, WWO, FA 和 MVO 等 4 种算法在 F1-F10 上的寻优性能.

5.4.1 性能分析

表 2 汇总了 4 种算法在标准测试函数 F1-F10 上的平均适应度值、标准差等统计结果. 4 种算法中的最优平均用

黑体加下划线表示, 如果 ApWWO 在某个函数上的平均适应度值排名第二, 则用黑体表示.

表 2 不同算法在 F1-F10 上的优化结果统计

函数	均值/方差	WWO	ApWWO	FA	MVO
F1	均值	1.38E+02	<u>2.13E-28</u>	6.89E+04	5.50E-01
	标准差	1.65E+02	1.76E-28	3.63E+04	3.36E-01
F2	均值	1.22E+05	<u>2.35E+03</u>	2.08E+08	<u>8.70E+02</u>
	标准差	1.69E+05	3.62E+03	1.09E+08	1.02E+03
F3	均值	1.80E+02	<u>1.39E+02</u>	5.07E+02	1.86E+02
	标准差	1.21E+02	7.60E+01	2.74E+02	9.74E+01
F4	均值	1.17E-12	<u>1.57E-15</u>	1.32E+00	9.15E-04
	标准差	2.04E-12	1.61E-15	1.07E+00	8.14E-04
F5	均值	6.85E-01	<u>2.12E-02</u>	1.82E+01	6.51E-02
	标准差	3.77E-01	3.24E-02	9.79E+00	4.26E-02
F6	均值	1.68E+01	<u>1.66E+01</u>	1.79E+01	<u>1.66E+01</u>
	标准差	9.04E+00	8.92E+00	9.88E+00	9.00E+00
F7	均值	2.00E+04	<u>4.87E+03</u>	1.10E+05	<u>-3.59E+02</u>
	标准差	1.84E+04	<u>5.93E+03</u>	5.85E+04	2.01E+02
F8	均值	3.46E+09	<u>1.25E+09</u>	7.28E+10	<u>5.90E+03</u>
	标准差	4.39E+09	2.14E+09	3.73E+10	6.20E+03
F9	均值	-9.79E+01	<u>-1.24E+02</u>	2.39E+02	-5.28E+01
	标准差	1.16E+02	9.64E+01	1.25E+02	9.49E+01
F10	均值	5.55E+01	<u>5.51E+01</u>	4.68E+02	<u>-1.44E+02</u>
	标准差	6.63E+01	6.87E+01	2.43E+02	8.03E+01

从表 2 中可以看出, ApWWO 在 F1, F3-F6 和 F9 上取得了最好的平均值, 在 F2, F7, F8 和 F10 上取得了次好的平均值, 而且相应的标准差也是 4 种算法中最小或者次小的, 表现出了更强的鲁棒性和适应性.

特别地, 除 F5 和 F10 外, ApWWO 无论是在平均值还是标准差方面都比 WWO 和 FA 高了若干数量级, 这表明了 ApWWO 算法寻优性能的高效性和鲁棒性.

与 MVO 相比, ApWWO 在单峰和多峰函数上的性能优于 MVO, 而在复合函数上的性能劣于 MVO. 值得指出的是, MVO 采取了若干搜索操作来改进其整体性能, 这表明需进一步结合高效的搜索机制或操作来改进 ApWWO 的寻优性能.

图 4 示出了 F1-F6 的进化曲线图. 从中可以看出, 对大多数测试函数而言, ApWWO 用较少的计算量 (20000 次函数评价, $20000/50000 = 40\%$) 就能收敛到较优解. 这表明, ApWWO 在收敛速度方面也优于 WWO 和 FA.

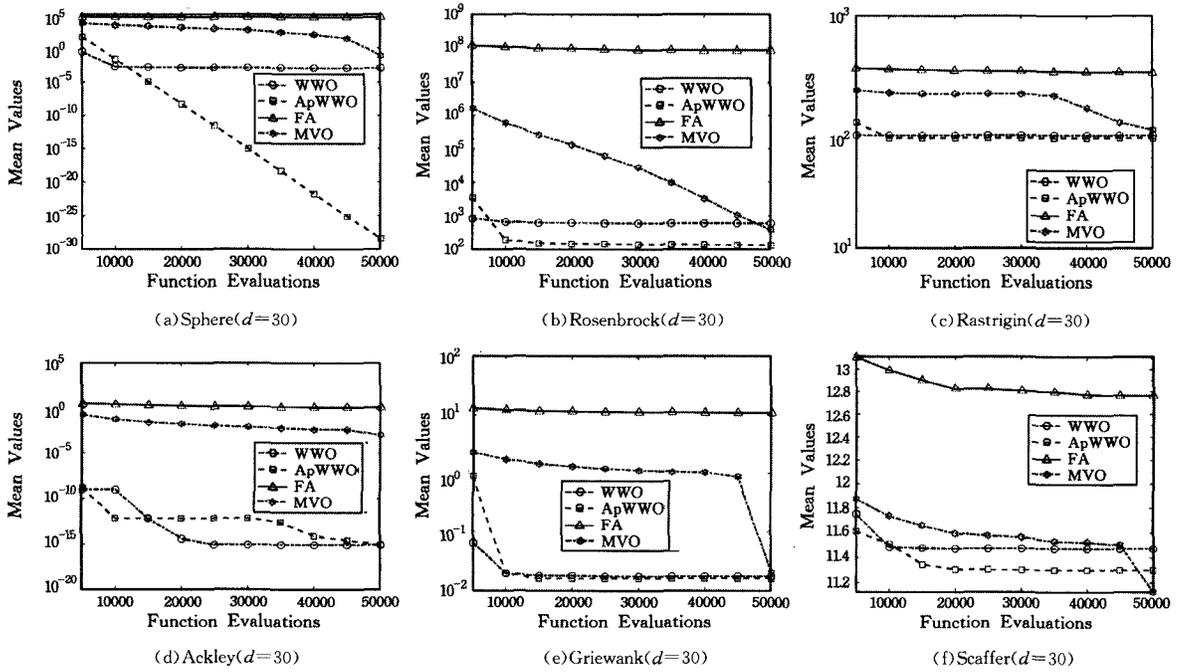


图 4 F1-F6 的进化曲线图

5.4.2 统计比较

为进一步比较不同算法的寻优性能,对 4 种算法在 10 个标准测试函数上的性能进行 Sign Test 和 Wilcoxon Signed Ranks Test^[17],详细的检验结果见表 3 和表 4。1/0/-1 表示 ApWWO 的性能优于/等于/劣于对比算法,+=/- 统计了 ApWWO 优于/等于/劣于对比算法的次数。

表 3 Sign Test 结果汇总

函数	ApWWO vs. WWO	ApWWO vs. FA	ApWWO vs. MVO
F1	1	1	1
F2	1	1	1
F3	1	1	-1
F4	1	1	1
F5	1	1	1
F6	1	1	0
F7	1	1	-1
F8	1	1	-1
F9	1	1	1
F10	1	1	-1
+=/-	10/0/0	10/0/0	5/1/4
p value	0.002	0.002	1.000

从表 3 可以看出,对于 Sign Test 而言,当显著性水平为 0.01 时,ApWWO 的性能都显著优于 WWO 和 FA,但与 MVO 相当。

表 4 Wilcoxon Signed Ranks Test 结果汇总

对比组	+=/-	R+	R-	p value
ApWWO vs. WWO	10/0/0	55	0	0.005
ApWWO vs. FA	10/0/0	55	0	0.005
ApWWO vs. MVO	5/1/4	30	15	0.374

从表 4 可以看出,对于 Wilcoxon Signed Ranks Test 而言,当显著性水平为 0.01 时,ApWWO 的性能都显著优于 WWO 和 FA,但是也与 MVO 相当。

上述统计比较的结果进一步验证了 ApWWO 的寻优性能与 WWO 和 FA 相比是显著高效和鲁棒的。

5.4.3 与 PSO 和 GA 的比较

为进一步探讨 ApWWO 的优化效果,本节将 ApWWO 与 PSO^[3,18-19] 和 GA^[2] 进行对比。该组实验的维数为 30,函数的最大评价次数为 50000,每组实验独立运行 30 次。实验比较结果如表 5 所列。

表 5 ApWWO 与 PSO 和 GA 的比较结果汇总

函数	均值/方差	ApWWO	PSO	GA
F1	均值	<u>2.13E-28</u>	3.16E-02	2.58E+00
	标准差	1.76E-28	1.51E-01	1.14E+00
F2	均值	2.35E+03	6.73E+01	3.03E+02
	标准差	3.62E+03	4.51E+01	1.17E+02
F3	均值	1.39E+02	5.61E+01	3.21E+00
	标准差	7.60E+01	1.59E+01	1.11E+00
F4	均值	<u>1.57E-15</u>	8.88E-15	4.26E-04
	标准差	1.61E-15	0.00E+00	7.78E-04
F5	均值	<u>2.12E-02</u>	5.24E-02	1.20E-01
	标准差	3.24E-02	1.46E-02	5.33E-02
F6	均值	1.66E+01	8.33E+00	1.73E+00
	标准差	8.92E+00	1.09E+00	3.86E-01
F7	均值	4.87E+03	-4.50E+02	-4.47E+02
	标准差	5.93E+03	3.41E-02	1.13E+00
F8	均值	1.25E+09	8.16E+02	4.09E+03
	标准差	2.14E+09	1.12E+03	3.71E+03
F9	均值	-1.24E+02	-1.12E+02	-1.20E+02
	标准差	9.64E+01	1.64E+01	1.45E+00
F10	均值	5.51E+01	-1.80E+02	-1.80E+02
	标准差	6.87E+01	6.32E-02	1.58E-14

从表 5 可以看出,在 10 个测试函数上,与 PSO 和 GA 相比,ApWWO 尽管在寻优性能和精度方面没有表现出显著优越性,但是其寻优效果与 PSO 和 GA 基本一致,这保证了 ApWWO 应用于优化问题的可行性。与 PSO 等改进研究类似,可以预期,通过融入高效的搜索算子或者局部搜索策略能较好地提高 ApWWO 的寻优性能。

5.4.4 维数对 ApWWO 的影响分析

本节探讨不同维数对 ApWWO 的影响,该组实验的维数

依次为 10,30,50,70 和 100,函数的最大评价次数为 50000, 每组实验独立运行 30 次。实验结果汇总于表 6 中。

表 6 WWO 与 ApWWO 在不同维数上的优化结果汇总

Table with 12 columns: 问题维数, 函数, F1, F2, F3, F4, F5, F6, F7, F8, F9, F10. Rows include dimensions 10, 30, 50, 70, 100 and functions F1-F10, comparing WWO and ApWWO algorithms.

从表 6 可以看出,随着维度的增大,WWO 和 ApWWO 的寻优性能均有不同程度的下降,但在同等条件下,ApWWO 的寻优性能整体上比 WWO 更好。这些结果表明,ApWWO 寻优性能的鲁棒性和最优性均优于 WWO。

5.4.5 种群规模对 ApWWO 的影响分析

为探讨种群大小对 ApWWO 寻优性能的影响,本节进一步研究不同种群规模下 WWO 和 ApWWO 的寻优性能的变化规律。

该组实验种群大小依次为 10,20,30,40 和 50,函数最大评价次数为 50000,每组实验独立运行 30 次。从表 7 的汇总结果可知,随着种群规模的增大,WWO 和 ApWWO 的寻优性能均有不同程度的下降,但在相同的种群规模和评价次数下,除了少部分情况外,ApWWO 寻优平均值和标准差均优于 WWO。这些结果也表明,ApWWO 寻优性能的鲁棒性和最优性均优于 WWO。

表 7 WWO 与 ApWWO 在不同种群规模下的优化结果汇总

Table with 12 columns: 种群规模, 函数, F1, F2, F3, F4, F5, F6, F7, F8, F9, F10. Rows include population sizes 10, 20, 30, 40, 50 and functions F1-F10, comparing WWO and ApWWO algorithms.

5.4.6 ApWWO 求解置换流水线调度问题

本节应用 ApWWO 来求解经典的置换流水线调度问题,以初步验证 ApWWO 求解工程优化问题的可行性。置换流

水线问题的测试案例采用标准测试集[20],具体结果如表 8 所列。car1_11_5 代表问题 car1 共有 11 个工件在 5 台机器上加工,应用 ApWWO 对每个测试问题独立求解 30 次,C* 为最

优 makespan,第 i 次运行的结果记为 C_i 。

定义 3 个指标:最好相对误差 (Best Relative Error, BRE)、平均相对误差 (Average Relative Error, ARE)、最坏相对误差 (Worst Relative Error, WRE)^[20]:

$$\begin{cases} BRE = \min_{i=1,2,\dots,n} \frac{C_i}{C^*} \\ ARE = \frac{1}{n} \sum_{i=1}^n \frac{C_i - C^*}{C^*} \\ WRE = \max_{i=1,2,\dots,n} \frac{C_i}{C^*} \end{cases} \quad (14)$$

表 8 ApWWO 求解置换流水线调度问题的结果汇总

s 问题	C^*	BRE	ARE	WRE
car1_11_5	7038	0	0	0
car2_13_4	7166	0	0	0
car3_12_5	7312	0	0	0
car4_14_4	8003	0	0	0
car5_10_6	7720	0	0	0
car6_8_9	8505	0	0	0
car7_7_7	6590	0	0	0
car8_8_8	8366	0	0	0
rec01_20_5	1247	0.002	0.002	0.002
rec03_20_5	1109	0.002	0.002	0.003
rec05_20_5	1242	0.012	0.012	0.021
rec07_20_10	1566	0.011	0.011	0.011
rec09_20_10	1537	0.013	0.013	0.014
rec11_20_10	1431	0.029	0.029	0.041
rec13_20_15	1930	0.018	0.018	0.022
rec15_20_15	1950	0.019	0.019	0.027
rec17_20_15	1902	0.027	0.027	0.036
rec19_30_10	2093	0.044	0.044	0.059
rec21_30_10	2017	0.070	0.070	0.066
rec23_30_10	2011	0.031	0.031	0.037
rec25_30_15	2513	0.055	0.055	0.052
rec27_30_15	2373	0.027	0.027	0.027
rec29_30_15	2287	0.064	0.064	0.060
rec31_50_10	3045	0.062	0.062	0.072
rec33_50_10	3114	0.034	0.034	0.034
rec35_50_10	3277	0.021	0.021	0.022
rec37_75_20	4951	0.092	0.092	0.093
rec39_75_20	5087	0.084	0.084	0.087
rec41_75_20	4960	0.096	0.096	0.097

从表 8 可以看出, ApWWO 在 car1—car8 上均能找到最优结果,在 rec01—rec41 上的求解效果也较优,最坏相对误差均不超过 0.01,这些结果初步验证了 ApWWO 求解实际工程优化问题的可行性和较优性。

结束语 水波优化算法是一种新型的群智能优化算法,尽管具有控制参数少、操作简单、容易实现等优点,但是也存在收敛较慢、搜索精度低等不足。鉴于此,首先从理论上分析并导出了算法收敛时控制参数应满足的条件,为算法的参数设置提供理论指导,同时通过借鉴自适应机制提出了一种基于自适应控制参数的改进水波优化算法。该算法在搜索的早期使用较大的参数来进行全局搜索,而在后期使用较小的参数进行局部搜索并加速收敛,更好地平衡了全局探索和局部开发能力。进一步,对 4 种算法 (ApWWO、WWO、FA、MVO) 在 10 个标准测试函数上进行对比仿真实验和统计比较,结果表明, ApWWO 在搜索精度、速度和鲁棒性等方面均显著优于 WWO 和 FA,在 5 个测试函数上优于 MVO,这表明引入自适应控制参数策略有助于提高 WWO 的性能。进

一步的分析结果表明,维数和种群规模对 ApWWO 优化效果有一定影响。同时,本文还探讨了 ApWWO 与 PSO 和 GA 的寻优性能对比、种群规模和维数对 ApWWO 寻优性能的影响以及 ApWWO 求解置换流水线调度问题的可行性。上述结果均表明, ApWWO 在求解优化问题时具备可行性和一定优越性。

需要指出的是, ApWWO 在复合函数上的性能劣于 MVO,根据“无免费午餐”定理^[21],试图设计一种在每个问题上的性能都最优的全能算法是不现实的。因此,一方面需要进一步引入更多的搜索机制和操作来提高 ApWWO 的性能;另一方面,本文主要目的是通过理论分析引入自适应控制参数策略,将 ApWWO 作为 WWO 的一种补充来弥补 WWO 的一些不足,为解决复杂优化问题提供一种新思路。

进一步的工作包括:1)从 WWO 的传播、折射和碎浪定义出发,深入研究 WWO 的优化机理,据此改进并设计更有效的搜索机制和操作;2)深入研究算法参数对 WWO 寻优性能的影响;3)研究 WWO 与其他局部搜索算子混合的 Memetic 算法,并将其应用到车间调度等实际工程问题^[20]中。

参 考 文 献

- [1] LIU B, WANG L, LIU Y, et al. A unified framework for population-based metaheuristics[J]. Annals of Operations Research, 2011, 186(1): 231-262.
- [2] GOLDBERG D E, HOLLAND J H. Genetic algorithms and machine learning[J]. Machine Learning, 1988, 3(2): 95-99.
- [3] POLI R, KENNEDY J, BLACKWELL T. Particle swarm optimization[J]. Swarm Intelligence, 2007, 1(1): 33-57.
- [4] DORIGO M, BIRATTARI M, STUTZLE T. Ant colony optimization[J]. Computational Intelligence Magazine, IEEE, 2006, 1(4): 28-39.
- [5] TIAN M C, BO Y M, CHEN Z M, et al. Firefly Algorithm Intelligence Optimized Particle Filter[J]. Acta Automatica Sinica, 2016, 42(1): 89-97. (in Chinese)
田梦楚, 薄煜明, 陈志敏, 等. 萤火虫算法智能优化粒子滤波[J]. 自动化学报, 2016, 42(1): 89-97.
- [6] GHANBARI A, ETTEFAGH M M. Robust adaptive control of a bio-inspired robot manipulator using bat algorithm [J]. Expert Systems with Applications, 2016, 56(c): 164-176.
- [7] JIA Y L, LIU S, SONG Y H. Cuckoo search algorithm based on swarm feature feedback [J]. Control and Decision, 2016, 31(6): 969-975. (in Chinese)
贾云璐, 刘胜, 宋颖慧. 基于种群特征反馈的布谷鸟搜索算法 [J]. 控制与决策, 2016, 31(6): 969-975.
- [8] WANG H, WANG W, SUN H, et al. A new cuckoo search algorithm with hybrid strategies for flow shop scheduling problems [OL]. <https://link.springer.com/article/10.1007/s00500-016-2062-9>.
- [9] LIN J, ZHANG S. An effective hybrid biogeography-based optimization algorithm for the distributed assembly permutation flow-shop scheduling problem[J]. Computers & Industrial Engineering, 2016, 97: 128-136.

参考文献

- [1] HO H C, FANN W J D, CHIANG H J, et al. Application of Rough Set, GSM and MSM to Analyze Learning Outcome-An Example of Introduction to Education [J]. *Journal of Intelligent Learning Systems and Applications*, 2016, 8(1): 23-38.
- [2] GOTLIB D, MARCINIAK J. Potential Application of the Rough Set Theory in Indoor Navigation [J]. *Lecture Notes in Computer Science*, 2014, 8537: 301-308.
- [3] CHEN C S, LIANG W Y, HSU H Y. A cloud computing platform for ERP applications[J]. *Applied Soft Computing*, 2015, 27(C): 127-136.
- [4] CHEN X, WANG X M, HUANG Y, et al. Fault diagnosis for tilt-rotor aircraft flight control system based on variable precision rough set-OMELM [J]. *Control and Decision*, 2015, 35(3): 433-440. (in Chinese)
陈晓, 王新民, 黄誉, 等. 倾转旋翼机飞控系统的变精度粗糙集-OMELM故障诊断方法[J]. *控制与决策*, 2015, 35(3): 433-440.
- [5] CHEN D, WANG C, HU Q. A new approach to attribute reduction of consistent and inconsistent covering decision systems with covering rough sets[J]. *Information Sciences*, 2007, 177(17): 3500-3518.
- [6] WU K, PAN W, WU L, et al. Incremental imputation method for incomplete decision system[J]. *Industrial Marketing Management*, 2011, 40(2): 267-277.
- [7] YANG X P. Completing incomplete data based on maximum similarity in Rough sets [J]. *Computer Engineering and Applications*, 2012, 48(36): 164-166. (in Chinese)
杨小平. 粗集中最大相似度的不完备数据补齐[J]. *计算机工程与应用*, 2012, 48(36): 164-166.
- [8] SHU W, SHEN H. A rough-set based incremental approach for updating attribute reduction under dynamic incomplete decision systems[C]//*Proceedings of the 2013 IEEE International Conference on Fuzzy Systems, FUZZ*, 2013: 1-7
- [9] ZHANG Q, ZHENG X, XU Z. Efficient Attribute Reduction Algorithm Based on Incomplete Decision Table[C]//*2009 Second International Conference on Intelligent Computation Technology and Automation*. IEEE Computer Society, 2009: 192-195.
- [10] LIU F, LI T R. Method for Attribute Reduction Based on Rough Sets Boundary Regions[J]. *Computer Science*, 2016, 43(3): 242-245, 284. (in Chinese)
刘芳, 李天瑞. 基于边界域的不完备信息系统属性约简方法[J]. *计算机科学*, 2016, 43(3): 242-245, 284.
- [11] LI R, ZHANG D, ZHAO Y, et al. Incremental Core Computing for Incomplete Decision Tables [C]//*International Symposium on Computational Intelligence and Design*. IEEE Computer Society, 2008: 270-273.
- [12] QIAN W, SHU W, XIE Y, et al. Feature Selection using Compact Discernibility Matrix-based Approach in Dynamic Incomplete Decision System [J]. *Journal of Information Science & Engineering*, 2015, 31(2): 509-527.
- [13] QIAN W B, YANG B R, XU Z Y, et al. Efficient Algorithm for Computing Tolerance Classes of Incomplete Decision Table [J]. *Journal of Chinese Computer Systems*, 2013, 34(2): 345-350. (in Chinese)
钱文彬, 杨炳儒, 徐章艳, 等. 基于不完备决策表的容差类高效求解算法[J]. *小型微型计算机系统*, 2013, 34(2): 345-350.
- [14] SHU W, SHEN H. Updating attribute reduction in incomplete decision systems with the variation of attribute set [J]. *International Journal of Approximate Reasoning*, 2014, 55(3): 867-884.
- [15] ZHANG T, YANG X, MA F. Improved algorithm for attribute core computing based on binary discernibility matrix [C]//*33rd Chinese Control Conference(CCC)*. 2014: 7400-7404.
- [16] GE H, YANG C J, LI L S. An Improved Attribute Reduction Algorithm Based on Binary Discernibility Matrix [J]. *Computer Technology and Development*, 2008, 18(8): 12-15. (in Chinese)
葛浩, 杨传健, 李龙澍. 一种改进的基于二进制可分辨矩阵属性约简算法[J]. *计算机技术与发展*, 2008, 18(8): 12-15.
- [17] SHU W H, XU Z Y, QIAN W B, et al. Quick Attribution Reduction Algorithm Based on Incomplete Decision Table [J]. *Journal of Chinese Computer Systems*, 2011, 32(9): 1867-1871. (in Chinese)
舒文豪, 徐章艳, 钱文彬, 等. 一种快速的不完备决策表属性约简算法[J]. *小型微型计算机系统*, 2011, 32(9): 1867-1871.
- (上接第 209 页)
- [10] ZHENG Y J. Water wave optimization : a new nature-inspired metaheuristic [J]. *Computers & Operations Research*, 2015, 55: 1-11.
- [11] ZHANG B, ZHENG Y J. Convergence Analysis of Water Wave Optimization Algorithm [J]. *Computer Science*, 2016, 43(4): 41-44. (in Chinese)
张蓓, 郑宇军. 水波优化算法收敛性分析[J]. *计算机科学*, 2016, 43(4): 41-44.
- [12] ZHANG Y J, ZHANG B, XUE J Y. Selection of Key Software Components for Formal Development Using Water Wave Optimization[J]. *Journal of Software*, 2016, 27(4): 933-942. (in Chinese)
郑宇军, 张蓓, 薛锦云. 软件形式化开发关键部件选取的水波优化方法[J]. *软件学报*, 2016, 27(4): 933-942.
- [13] PAN Q K, SANG H Y, DUAN J H, et al. An improved fruit fly optimization algorithm for continuous function optimization problems [J]. *Knowledge-Based Systems*, 2014, 62(5): 69-83.
- [14] RUEDA J L, ERLICH I. Testing MVMO on learning - based real-parameter single objective benchmark optimization problems [C]//*2015 IEEE Congress on Evolutionary Computation*. New York: IEEE Press, 2015: 1025-1032.
- [15] YANG X S. Firefly algorithm, stochastic test functions and design optimisation[J]. *International Journal of Bio-Inspired Computation*, 2010, 2(2): 78-84.
- [16] MIRJALILI S, MIRJALILI S M, HATAMLOU A. Multi-verse optimizer: a nature-inspired algorithm for global optimization [J]. *Neural Computing and Applications*, 2015, 149(3): 29-38.
- [17] ZHANG H, LI B, ZHANG J, et al. Parameter estimation of nonlinear chaotic system by improved TLBO strategy[J]. *Soft Computing*, 2016, 20(12): 4965-4980.
- [18] ZHAO Z S, FENG X, LIN Y Y, et al. Improved rao blackwellized particle filter by particle swarm optimization[J]. *Journal of Applied Mathematics*, 2013, 10(4): 15-22.
- [19] ZHAO Z S, FENG X, LIN Y Y, et al. Evolved neural network ensemble by multiple heterogeneous swarm intelligence [J]. *Neurocomputing*, 2015, 149(3): 29-38.
- [20] LIU B, WANG L, JIN Y H. An effective PSO-based memetic algorithm for flow shop scheduling [J]. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 2007, 37(1): 18-27.
- [21] WOLPERT D H, MACREARY W G. No free lunch theorems for optimization [J]. *IEEE Transactions on Evolutionary Computation*, 1997, 1(1): 67-82.