一种基于混沌和单纯形法的水波优化算法

吴秀丽[1] 周永权[2]
(广西大学计算机与电子信息学院 南宁 530004)[1] (广西民族大学信息科学与工程学院 南宁 530006)[2]

摘 要 水波优化(Water Wave Optimization,WWO)算法是一种基于浅水波理论的新兴元启发式优化算法,通过模拟水波的传播、碎浪、折射操作在解空间中进行全局搜索。为提高算法的收敛速度和精度,提出了一种基于混沌(Chaotic)优化和单纯形法(Simplex Method,SM)的水波优化算法,简称为CSMWWO。在CSMWWO算法中,引入了混沌优化策略来降低随机初始化的种群对收敛速度和求解精度的影响,在混沌优化策略的基础上又引入了局部搜索能力较强的单纯形法来提高WWO算法的收敛速度。将CSMWWO与包括WWO在内的4个启发式算法在12个基本测试函数上进行了测试,结果表明改进后的算法在计算精度和收敛速度上都有一定程度的提高,所提出的混合水波优化算法能改进水波优化算法的整体性能。

关键词 水波优化,混沌策略,单纯形法
中图法分类号 TP183 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.05.039

Improved Water Wave Optimization Algorithm Based on Chaos Optimization and Simplex Method

WU Xiu-li[1] ZHOU Yong-quan[2]
(School of Computer,Electronic and Information,Guangxi University,Nanning 530004,China)[1]
(College of Information Science and Engineering,Guangxi University for Nationalities,Nanning 530006,China)[2]

Abstract Water wave optimization (WWO) algorithm,which is based on shallow water wave theory,is a new nature-inspired metaheuristic,it simulates mimicking shallow water wave motions including propagation,breaking,refraction for globally search in solution space. In order to improve the convergence speed and accuracy of the algorithm,an improved version of water wave optimization which is based on chaos optimization and simplex method(SM) was represented in this paper. It is named CSMWWO. In CSMWWO algorithm,in order to reduce the population which is random initialization influences on the convergence speed and precision of the algorithm,the chaotic optimization strategy was introduced,meanwhile the simplex method which has strong local searching ability was introduced based on chaos optimization strategy to improve the convergence speed of WWO algorithm. CSMWWO is compared with 4 heuristic algorithms including WWO in 12 benchmark functions,experimental results show that the improved algorithm have a certain degree of improvement on calculation accuracy and convergence speed,the proposed hybrid wave optimization algorithm can improve the overall performance of the water wave optimization algorithm.

Keywords Water wave optimization (WWO),Chaos strategy,Simplex method(SM)

水波优化(Water Wave Optimization,WWO)[1]算法是我国学者郑宇军于2014年提出的,它以浅水波理论[2-3]为基础,在搜索空间中进行优化。该算法具有简单易懂、易于实现和参数较少等优点。目前,WWO作为一种新的启发式优化方法,已被成功应用于高铁调度[1]和TSP求解[4]等优化问题中。

为进一步提高WWO的性能,Zhang等人对WWO进行了一些改进,提出了种群规模可变的水波优化算法[5],在折射操作中引入了全面学习机制。Zheng又提出了另外一种种群规模可变的水波优化算法[6],在该算法中,删除折射操作,但为平衡探测和开采操作并减少删除折射操作后的影响,引入了种群规模可变的策略。虽然WWO算法具有简单易懂、易于实现和参数较少等优点,但作为一种启发式优化算法,其收敛性和求解精度与随机初始化的种群密切相关[7],使用随机初始化的参数序列在算法的不同执行中可能会产生不同的结果。所以,提高初始化时种群分布的多样性,有助于避免算法陷入局部最优,并获得较好的结果。目前的WWO算法中初始种群是随机初始化的,随机初始化的种群没有任何特定的分布,因此初始种群的分布是不理想的。混沌序列可以用来弥补随机初始化的缺点。混沌的本质是随机的、不可预测的,但是也有一些规律性,于是,本文提出了基于混沌策略的水波优化算法。为进一步提高算法的收敛速度,将混沌策略与局

到稿日期:2016-03-16 返修日期:2016-08-20 本文受国家自然科学基金项目(61463008)资助。
吴秀丽(1989—),女,硕士生,主要研究方向为启发式优化算法;周永权(1962—),男,教授,博士生导师,主要研究方向为智能优化、神经网络,E-mail:yongquanzhou@126.com。

部搜索能力较强的单纯形法[8]相结合，提出了一种混沌优化策略和单纯形法相结合的水波优化算法来进一步提高收敛速度和求解精度，避免陷入局部最优。

1　水波优化算法

WWO 是一种新兴的启发式优化算法[1]，它从应用浅水波模型解决优化问题中获得启发，将种群中每个个体类比为一个“水波”对象，通过模拟水波的传播、碎浪、折射操作在解空间中进行全局搜索。不失一般性，假设有一个求最大值的函数 F，这里可以把要处理的实际问题与浅水波模型进行类比，对应关系如图 1 所示。

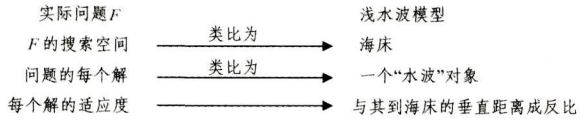

图 1　实际问题与浅水波模型的对应关系

每个水波都有一个波高 h 和波长 λ，在初始化种群时，把 h 设置成一个常数 h_{max}，λ 一般设置为 0.5。每个水波的适应度值与其到海床的垂直距离成反比，由此可知水波距离海床越近，适应度越大，其波高越高，波长越短，如图 2 所示。解决优化问题的过程中是以传播、碎浪和折射这 3 种操作来进行全局搜索的，以此来模拟自然界中的水波在流动过程中的行为。

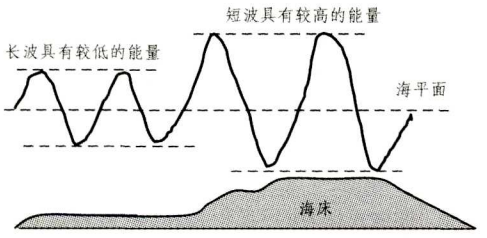

图 2　深水和浅水区域的不同波浪形状

1.1　传播

凡水波都要进行传播操作。假设原始的水波为 x，传播后的水波为 x'，求最大值函数 F 的维度为 D，传播操作就是在 x 波的每一维上进行如式(1)所示的操作来产生 x'。

$$x'(d)=x(d)+rand(-1,1)\cdot\lambda L(d) \tag{1}$$

其中，$d\in D$，$x(d)$ 表示原始水波的第 d 维，$x'(d)$ 表示传播后水波的第 d 维，$rand(-1,1)$ 是 $[-1,1]$ 之间均匀分布的随机数，λ 是水波的波长，$L(d)$ 表示搜索空间第 d 维的长度。如果 $L(d)$ 的长度超过了搜索空间第 d 维的长度，则将其按照式(2)随机设置为有效范围内的一个新位置。

$$L(d)=lb(d)+rand(\,)\cdot(ub(d)-lb(d)) \tag{2}$$

其中，$lb(d)$ 表示搜索空间的第 d 维的下界，$ub(d)$ 表示搜索空间第 d 维的上界，$rand(\,)$ 表示 $[0,1]$ 之间的随机数。

传播操作结束后，用适应度函数 f 计算 x' 的适应度值 $f(x')$。如果 $f(x')>f(x)$，在种群中用 x' 代替 x，这时 x' 的波高重置为 h_{max}；否则，保留原始水波 x，同时为了模拟水波在传播过程中的能量耗散，对波高 h 进行减 1 操作。

在每一轮迭代结束后，WWO 会按式(3)重新计算每个水波的波长。

$$\lambda=\lambda\cdot\alpha^{-(f(x)-f_{min}+\varepsilon)/(f_{max}-f_{min}+\varepsilon)} \tag{3}$$

其中，α 为波长衰减系数，f_{max} 和 f_{min} 分别是当前种群中最大和最小的适应度值，ε 为一极小的正数以避免分母为 0。通过观察式(3)可知，适应度值越大，波长越短，相应的水波传播的范围也越小，从而使适应度值较大的水波加强局部搜索，如图 2 所示。

1.2　碎浪

在水波理论中，水波能量的不断增加会使波峰越来越陡峭，当波峰的速度超过水波的传播速度时，这个水波就会碎裂成一系列的孤立波。在 WWO 算法中，对传播后新找到的最优解 x^* 进行碎浪操作来生成一系列的孤立波，以此来提高种群的多样性。对传播后的最优解进行碎浪操作的方法如下：先随机选择 k 维(这里 k 是介于 1 和一个预定义参数 k_{max} 之间的一个随机数)，在选择的每一维上按式(4)产生一个孤立波 x'：

$$x'(d)=x(d)+N(0,1)\cdot\beta L(d) \tag{4}$$

其中，β 是碎浪系数(介于 0.001 与 0.01 之间)。碎浪操作过后如果所有的孤立波的适应度值都小于 x^* 的适应度值，则保留 x^*；否则，在种群中用孤立波中的最优解来代替 x^*。

1.3　折射

水波在传播的过程中流经山底、山谷这些具有深海特征的陡峭地方时就会绕射，也就是所说的折射，发生折射时的波线分布如图 3 所示。通过观察图 3 可知，水波收敛在浅水区而发散在深水区。为了模拟水波在传播过程中的能量耗散，传播后波高会进行减 1 操作。在 WWO 中只对波高为 0 的水波进行折射操作，以避免搜索停滞。用式(5)来更新水波折射后每一维的位置：

$$x'(d)=N\left(\frac{(x^*(d)+x(d))}{2},\frac{|x^*(d)-x(d)|}{2}\right) \tag{5}$$

其中，x^* 是当前种群中的最优值，$N(\mu,\sigma)$ 表示均值为 μ、标准差为 σ 的高斯随机数。折射操作后，x' 的波高仍重置为 h_{max}，同时按式(6)对水波的波长进行更新：

$$\lambda=\lambda\frac{f(x)}{f(x')} \tag{6}$$

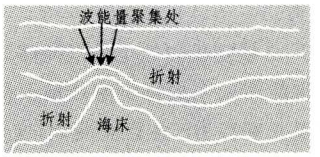

图 3　水波的折射

WWO 算法的框架结构如算法 1 所示。

算法 1　WWO 算法

输入：待求解的优化问题(设其维度为 D，适应度函数为 f)

输出：算法找到的最优解 x*

Step 1　初始化相关参数，随机初始化一个具有 n 个水波的种群 P，计算每个水波的适应度 f(x)，找出最优解 x*；

Step 2　当终止条件不满足时，转 Step 3，否则算法结束，返回 x*；

Step 3　对每个水波 x 执行如下操作：

Step 3.1 根据式(1)进行传播操作；

Step 3.2 如果生成的水波 x′满足 f(x′)>f(x),转 Step 3.2.1;否则,转 Step 3.3;

Step 3.2.1 用 x′代替 x;

Step 3.2.2 如果 f(x′)>f(x*),根据式(4)进行碎浪操作,并用 x′代替 x*,转 Step 3.4;

Step 3.3 对水波 x 的波高 h 执行减 1 操作;

Step 3.3.1 如果 h=0,根据式(5)和式(6)对 x 进行折射操作;

Step 3.4 根据式(2)更新每个水波的波长,转 Step 2。

2 基于混沌优化和单纯形法的水波优化算法 CSMWWO

水波优化算法的收敛性和求解精度与随机初始化的种群密切相关。为提高 WWO 算法的求解精度,引入了混沌优化策略;同时,为进一步提高 WWO 算法的收敛速度,引入了局部搜索能力较强的单纯形法[8]。

2.1 混沌优化策略

启发式优化算法一般都是对初值敏感的,如果种群中的个体都是初始化在最优解的邻域内,则算法的收敛速度快、求解精度高;如果种群初始化时离最优解较远,则算法的收敛速度就会下降,得到的全局最优解的精度也会相应降低。很多情况下,随机初始化的种群分布都不是那么理想,混沌序列可以用来弥补随机初始化种群分布不理想的缺点。混沌是非线性动力系统中普遍存在的一种现象,混沌的过程看似混乱,但其实具有一定的规律性,能在一定范围内按其自身规律不重复地遍历所有状态。混沌优化算法的基本思想是将待优化变量通过混沌映射规则映射到混沌空间([0,1]之间),利用混沌变量的随机性、遍历性、规律性[9]特点在混沌空间进行搜索,再将获得的解映射回原解空间。用混沌序列初始化种群,可以保证种群中的个体对解空间进行遍历搜索,克服随机初始化种群分布较差的问题。当前许多启发式算法都结合了混沌优化的策略[10-13]来改进算法的性能。在当前的研究中,产生混沌序列的混沌映射有多种,本文采用 Logistic Map[14]来产生混沌序列的初始种群。Logistic Map 函数的形式如式(7)所示。

$$X_{k+1}=\alpha X_k(1+X_k) \tag{7}$$

其中,α 是混沌系数,$\alpha\in[0,4]$,本文取 $\alpha=3$。

在式(7)中,X_{k+1}是随机初始化的混沌序列,X_{k+1}的范围在 0 至 1 之间。混沌序列中的每个值依赖于初始值,初始值的微小改变会使其长期行为产生巨大变化,这是混沌的特征,因此生成的混沌种群具有混沌的特性。混沌初始化初始种群的过程如算法 2 所示。

算法 2 混沌初始化初始种群

Step 1 随机初始化一个具有 n 个水波的种群 P;

Step 2 利用式(7)产生与随机初始化种群对应的混沌种群 CP;

这种方法得到的混沌初始化种群不可避免地还会存在随机初始化的个体。为了进一步减小随机初始化的个体对求解精度和收敛速度的影响,利用式(7)对每次迭代后的最优解进行混沌扰动。混沌扰动的过程如算法 3 所示。

算法 3 混沌扰动最优解

Step 1 将每一代中的最优解映射到 Logistic 方程的定义域[0,1];

Step 2 由 Logistic Map 得到当前最优解对应的混沌解;

Step 3 将该混沌解与当前最优解进行比较,若混沌解较优,则用混沌解代替当前最优解,否则保留当前最优解。

2.2 单纯形法

单纯形法(Simplex Method,SM)也可以称为可变多面体搜索法,具有原理简单、搜索速度快、计算量小和局部搜索能力强等优点,目前许多启发式算法[15-17]都引入了单纯形法来加快算法的收敛速度。为了进一步提高 WWO 算法的收敛速度,本文引入了局部搜索能力较强的单纯形法。

单纯形法的搜索原理[18]是在由 $n+1$ 个顶点构成的多面体所在的 n 维空间中,首先由适应度评价函数确定各个顶点的适应度值,其次确定其中的最优点、次优点和最差点,接着通过反射、扩张、收缩或压缩等策略找到一个较优的点取代最差点,从而构成新的多面体,如此反复迭代便可以找到或逼近最优点。单纯形法的搜索点如图 4 所示。

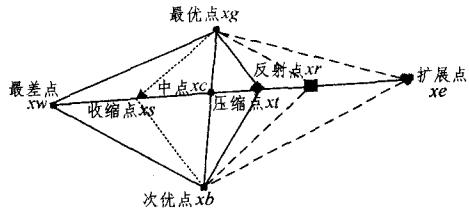

图 4 单纯形法搜索的不同点

单纯形法的搜索步骤如下:

Step 1 计算所有顶点对应的适应度函数值,确定其中的最优点 xg、次优点 xb 和最差点 xw,适应度函数为 f,根据式(8)计算最优点 xg 和次优点 xb 的中点位置 xc。

$$xc=\frac{xg+xb}{2} \tag{8}$$

Step 2 根据式(9)执行反射操作,将最差点 xw 依据中心点 xc 进行反射。

$$xr=xc+\alpha(xc-xw) \tag{9}$$

其中,xr 为反射点;α 为反射系数,本文中取 1。

Step 3 如果 $f(xr)>f(xg)$,说明反射方向正确,接着根据式(10)进行扩张操作。

$$xe=xc+\gamma(xr-xc) \tag{10}$$

其中,xe 为扩张点;γ 为扩张系数,本文中取 2。计算扩张点的适应度值 $f(xe)$,如果 $f(xe)>f(xg)$,用扩张点 xe 代替最差点 xw;否则,用反射点 xr 代替最差点 xw。

Step 4 如果 $f(xr)<f(xg)$,说明反射方向有误,按式(11)进行压缩操作。

$$xt=xc+\beta(xw-xc) \tag{11}$$

其中,xt 为压缩点;β 为压缩系数,本文中取 0.5。如果 $f(xt)>f(xw)$,则用 xt 取代最差点。

Step 5 如果 $f(xw)<f(xr)<f(xg)$,用式(12)进行收缩操作。

$$xs=xc-\beta(xw-xc) \tag{12}$$

其中,xs 为收缩点;β 为收缩系数,与式(11)中取值相同。如果 $f(xs)>f(xw)$,则用 xs 取代最差点 xw;否则,用反射点 xr 代替最差点 xw。

在改进后的 WWO 即 CSMWWO 中，算法在每次混沌扰动最优解之后、逐个更新波长之前实施单纯形法优化最差解。

2.3 基于混沌优化策略和单纯形法的水波优化算法框架结构

CSMWWO 算法的框架结构见算法 4，算法流程图见图 5。

算法 4 CSMWWO 算法

输入：待求解的优化问题(设其维度为 D，适应度函数为 f)

输出：算法找到的最优解 x^*

Step 1　初始化相关参数。

Step 2　用算法 2 混沌初始化一个具有 n 个水波的种群 P，计算每个水波的适应度 f(x)，找出最优解 x^*。

Step 3　当终止条件不满足时，转 Step 4；否则算法结束，返回 x^*。

Step 4　对每个水波 x 执行如下操作：

Step 4.1 根据式(1)进行传播操作；

Step 4.2 如果生成的水波 x' 满足 $f(x')>f(x)$，转 Step 4.2.1；否则，转 Step 4.3；

Step 4.2.1 用 x' 代替 x；

Step 4.2.2 如果 $f(x')>f(x^*)$，根据式(4)进行碎浪操作，并用 x' 代替 x^*，转 Step 4.4；

Step 4.3 对水波 x 的波高 h 执行减 1 操作；

Step 4.3.1 如果 h=0，根据式(5)和式(6)对 x 进行折射操作；

Step 4.4 根据算法 3 对最优解 x^* 进行混沌扰动；

Step 4.5 根据单纯形法优化较差解；

Step 4.6 根据式(2)更新每个水波的波长，转 Step 3。

注：Step 2，Step 4.4 和 Step 4.5 是与原 WWO 算法不同的部分。

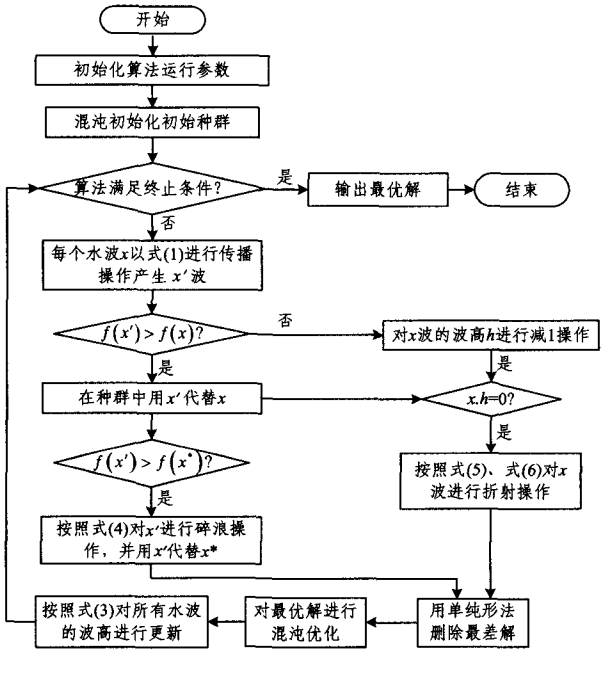

图 5　CSMWWO 算法流程图

3　实验结果与分析

3.1　实验设置和比较方法

为了测试 CSMWWO 算法的性能，将 CSMWWO 算法与花授粉算法(Flower Pollination Algorithm，FPA)[19]、布谷鸟搜索(Cuckoo Search，CS)[20]算法、蝙蝠算法(Bat Algorithm，BA)[21]和基本的水波算法 WWO 在 12 个测试函数上进行了比较。相关测试函数如表 1 所列。

表 1　测试函数(其中 *Scope* 为函数的搜索空间，f_{min} 为函数的最优值)

Function	Name	Benchmark function	*Scope*	f_{min}
f_1	*Sphere*	$f(x)=\sum_{i=1}^{n}x_i^2$	$[-100,100]$	0
f_2	Rastrigin	$f(x)=\sum_{i=1}^{n}[x_i^2-10\cos(2\pi x_i)+10]$	$[-5.12,5.12]$	0
f_3	Rosenbrock	$f(x)=\sum_{i=1}^{n-1}[(x_i-1)^2+100(x_i^2-x_{i+1})^2]$	$[-2,2]$	0
f_4	Ackley	$f(x)=-20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}-\exp(\frac{1}{n}\sum_{i=1}^{n}\cos 2\pi x_i))+20+e$	$[-32,32]$	0
f_5	Griewank	$f(x)=\frac{1}{4000}\sum_{i=1}^{n}(x_i^2)-\prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})+1$	$[-600,600]$	0
f_6	Salomon	$f(x)=-\cos(2\pi\sqrt{\sum_{i=1}^{D}x_i^2})+0.1\sqrt{\sum_{i=1}^{D}x_i^2}+1$	$[-100,100]$	0
f_7	Rotated hyper-ellipsoid	$f(x)=\sum_{i=1}^{n}\sum_{j=1}^{i}(x_j)^2$	$[-100,100]$	0
f_8	Alpine	$f(x)=\sum_{i=1}^{n}\lvert x_i\sin(x_i)+0.1x_i\rvert$	$[-10,10]$	0
f_9	Easom	$f(x_1,x_2)=-\cos(x_1)\cos(x_2)\exp(-(x_1-\pi)^2-(x_2-\pi)^2)$	$[-100,100]$	−1
f_{10}	Shubert	$f(x)=\sum i\cos((i+1)x_1+i)\cdot\sum_{i=1}^{5}i\cos((i+1)x_2+i)$	$[-10,10]$	−186.731
f_{11}	Shekel	$f(x)=-\sum_{i=1}^{m}(\sum_{j=1}^{n}[(x_j-a_{ij})^2+c_j])^{-1}, m=30$	$[0,10]$	−10.5364
f_{12}	Michalewicz	$f(x)=-\sum_{i=1}^{n}(\sin x_i)\times(\sin(\frac{i\times x_i^2}{\pi}))^{2m}, m=10$	$[0,\pi]$	−9.6615

算法中参数的微小改变都有可能会影响算法的性能，为便于比较和测试结果的公平性，对 FPA，CS，BA 和 WWO 在表 1 所列的测试函数上进行测试时，将这些算法参数设置如下。

1)FPA：转换概率 $P=0.8$。

2)CS：交换概率 $Pa=0.25$。

3)BA：频率最大、最小值分别为 $f_{max}=2$，$f_{min}=0$，脉冲发射率 $r=0.5$，音量 $A=0.25$。

4)WWO：波长 $\lambda=0.5$，波高 $h_{max}=12$，波长衰减系数 $\alpha=1.0026$，碎浪系数 β 的最大、最小值分别为 0.25，0.001，$k_{max}=\min(12,D/2)$(D 为问题的维度)。

5)CSMWWO：波长 $\lambda=0.5$，波高 $h_{max}=12$，波长衰减系

数 $\alpha=1.0026$，碎浪系数 β 的最大、最小值分别为 0.25 和 0.001，$k_{max}=\min(12,D/2)$（D 为问题的维度），混沌系数 $\alpha_1=3$，反射因子 $\alpha_2=1$，扩张因子 $r=2$，压缩因子 $\beta_1=0.5$，收缩因子 $\beta_2=0.5$。

实验环境如下：处理器为 Intel(R) Xeorn 3.5GHz，内存为 8GB，Windows 7 系统；应用软件为 Matlab 2012a。在实验中，用 30 维的问题来进行测试，同时每个算法的种群规模为 50，最大迭代次数设置为 1000，独立运行次数为 30，每个算法在 f_1-f_{12} 这些函数上的评价次数如表 2 所列（同一种算法在每个函数上采用相同的评价次数）。

表 2 不同算法在测试函数 f_1-f_{12} 上的评价次数

算法	FPA	CS	BA	WWO	CSMWWO
评价次数	1000	2000	1000	1000	2000

3.2 实验结果

表 3 列出了包括 CSMWWO 在内的 5 个算法的测试结果，其中“Max”，“Min”和“Median”分别表示 30 次独立运行结果中的最大值、最小值和中值；“Std.”表示 30 次运行结果的标准方差；“Rank”表示就 Median 值而言 5 个算法的排名情况；D 表示测试函数的维度。在比较的过程中，最优的“Min”，“Median”和“Std.”用粗体显示。

表 3 不同算法在测试函数 f_1-f_{12} 上的测试结果

Benchmark functions	Algorithm	Results				
		Max	Min	Median	Std.	Rank
f_1 (D=30)	FPA	2420	812.42	1635.56	480.46	5
	CS	0.03	4.23E−3	1.06E−2	5.55E−3	4
	BA	0.002	1.07E−03	1.32E−04	1.52E−04	3
	WWO	2.52E−05	3.97E−06	9.34E−06	4.84E−06	2
	CSMWWO	3.2E−65	**8.55E−69**	**1.71E−67**	**6.70E−66**	1
f_2 (D=30)	FPA	280.39	153.52	246.49	28.10	5
	CS	97.32	56.40	86.00	11.05	4
	BA	44.04	12.15	25.11	**6.69**	2
	WWO	147.09	29.94	84.93	27.94	3
	CSMWWO	47.76	**11.94**	**23.38**	**8.04**	1
f_3 (D=30)	FPA	416.37	90.48	236.93	83.24	5
	CS	26.75	23.81	25.23	0.69	2
	BA	29.52	25.73	28.20	1.33	4
	WWO	82.55	22.22	27.64	10.12	3
	CSMWWO	0.12	**7.52E−06**	**8.73E−03**	**0.03**	1
f_4 (D=30)	FPA	10.10	5.87	8.40	1.07	5
	CS	2.66	1.20	2.31	0.40	3
	BA	2.02	0.02	**1.34**	0.66	1
	WWO	6.99	2.58	3.69	1.29	4
	CSMWWO	2.66	**6.83E−08**	2.01	**0.50**	2
f_5 (D=30)	FPA	25.12	7.44	15.97	3.89	5
	CS	0.17	2.12E−2	7.66E−2	4.21E−2	4
	BA	9.15E−05	4.91E−05	7.29E−05	1.14E−05	2
	WWO	0.03	6.34E−05	6.18E−04	7.73E−03	3
	CSMWWO	5.55E−16	**0**	**1.11E−16**	**1.54E−16**	1
f_6 (D=30)	FPA	7.70	4.00	6.40	0.84	5
	CS	1.60	1.04	1.40	0.14	4
	BA	0.50	0.20	0.40	0.06	2
	WWO	1.20	0.50	0.86	0.16	3
	CSMWWO	0.40	**0.20**	**0.30**	**0.05**	1
f_7 (D=30)	FPA	21335.67	5616.91	13702.78	4236.71	5
	CS	658.88	310.52	486.60	95.73	3
	BA	0.01	2.70E−03	4.58E−02	1.55E−03	2
	WWO	3150.44	167.70	1550.55	760.72	4
	CSMWWO	6.20E−12	**7.16E−14**	**5.00E−13**	**1.50E−12**	1
f_8 (D=30)	FPA	31.71	16.10	25.90	3.61	5
	CS	8.47	4.76	6.86	0.87	4
	BA	0.02	0.01	0.02	1.44E−03	2
	WWO	9.28	0.19	3.71	2.96	3
	CSMWWO	2.53E−03	**6.26E−10**	**1.17E−06**	**7.10E−04**	1
f_9 (D=2)	FPA	**−1**	**−1**	**−1**	**0**	1
	CS	**−1**	**−1**	**−1**	**0**	1
	BA	−1	−1	−1	1.57E−08	5
	WWO	**−1**	**−1**	**−1**	**0**	1
	CSMWWO	**−1**	**−1**	**−1**	**0**	1
f_{10} (D=2)	FPA	−186.73	−186.73	−186.73	1.35E−6	4
	CS	−186.73	−186.73	−186.73	2.81E−7	3
	BA	−186.73	−186.73	−186.73	1.99E−05	5
	WWO	−186.73	−186.73	−186.73	5.22E−14	2
	CSMWWO	−186.73	**−186.73**	**−186.73**	**4.48E−14**	1

（续表）

Benchmark functions	Algorithm	Results				
		Max	Min	Median	Std.	Rank
f_{11} (D=4)	FPA	−10.54	−10.54	−10.54	4.52E−10	4
	CS	−5.13	−5.13	−5.13	1.82E−15	2
	BA	−5.13	−5.13	−5.13	2.20E−05	5
	WWO	−10.54	**−10.54**	**−10.54**	**1.71E−15**	1
	CSMWWO	−10.54	−10.54	−10.54	2.78E−15	3
f_{12} (D=10)	FPA	−6.50	−9.48	−8.01	0.64	4
	CS	−8.42	−9.19	−8.75	**0.23**	3
	BA	−3.88	−6.05	−4.63	0.62	5
	WWO	−7.72	−9.53	−8.77	0.50	2
	CSMWWO	−6.95	**−9.57**	**−8.91**	0.62	1

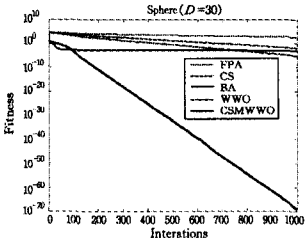

图 6　函数 f_1 的收敛曲线

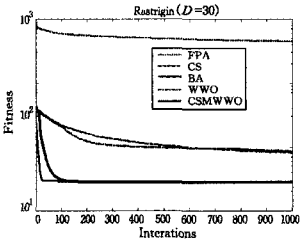

图 7　函数 f_2 的收敛曲线

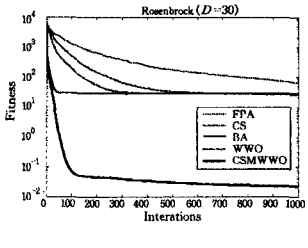

图 8　函数 f_3 的收敛曲线

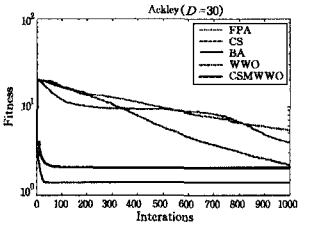

图 9　函数 f_4 的收敛曲线

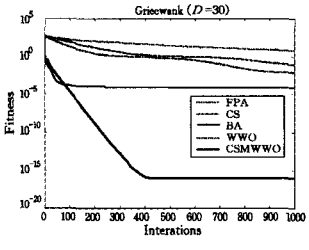

图 10　函数 f_5 的收敛曲线

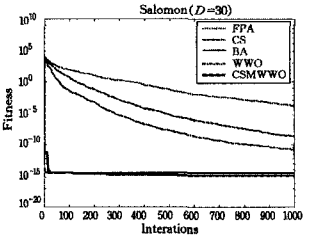

图 11　函数 f_6 的收敛曲线

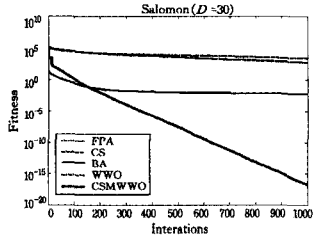

图 12　函数 f_7 的收敛曲线

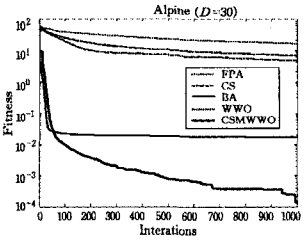

图 13　函数 f_8 的收敛曲线

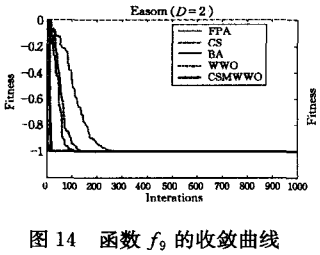

图 14　函数 f_9 的收敛曲线

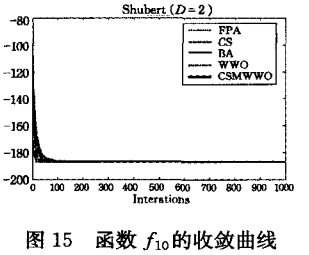

图 15　函数 f_{10} 的收敛曲线

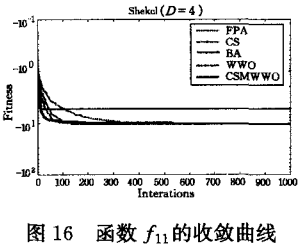

图 16　函数 f_{11} 的收敛曲线

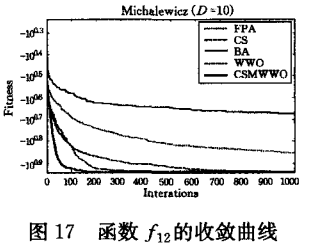

图 17　函数 f_{12} 的收敛曲线

图 6—图 17 分别示出 FPA，CS，BA，WWO，CSMWWO 求函数 f_1-f_{12} 的最小值时函数值随迭代次数的变化情况即收敛曲线。为了测试结果的公平性，收敛曲线是 20 次运行结果的平均值。

3.3　结果分析

从表 3 的结果中可以看出，相比 FPA，CS，BA，WWO，基于混沌优化和单纯形法的 CSMWWO 搜索最优值的能力更强，求解的最优值更加接近理论值。在 f_1-f_8 这 8 个 30 维函数中，CSMWWO 在除了 f_4 之外的 7 个函数上都获得了最优值。现将 f_1-f_8 的实验结果分析如下。

与 BA 和 WWO 相比，CSMWWO 的最优精度提高了 66 个和 63 个数量级，与 FPA 相比提高了 71 个数量级，并且获得了最小的标准差。这说明对于函数 f_1，CSMWWO 不仅探测最优解的能力较好，而且稳定性较好。

对于函数 f_2，CSMWWO 也获得了最优精度，其最优精度比 WWO 小了一半多。由于 f_2 是一个多峰、具有多个局部极小值的函数，CSMWWO 在 f_2 上获得了最优值，说明 CSMWWO 算法具有较强的全局搜索性能。

对于函数 f_3，f_7，f_8，CSMWWO 的最优精度分别为 e^{-6}，e^{-14} 和 e^{-10}，其最优求解精度与 FPA 相比分别提高了 7 个、16 个和 11 个数量级，与 CS 相比分别提高了 7 个、16 个和 10 个数量级，与 BA 相比分别提高了 7 个、17 个和 8 个数量级，与 WWO 相比分别提高了 7 个、16 个和 9 个数量级。在函数 f_3，f_7，f_8 上，CSMWWO 都取得了最小的标准差，进一步证明了 CSMWWO 的稳定性较好。函数 f_3 是一个多峰、多局部函数，查找全局最优解较困难，CSMWWO 在 f_3 上的测试结果显示了 CSMWWO 在处理多峰、多局部函数上的优势。函数 f_7 是一个单峰函数，由图 12 可知 CSMWWO 在 f_7 上搜索最小值时具有较快的下降速度。

对于函数 f_4，BA 的结果最优，CSMWWO 的结果次之，但是 CSMWWO 的结果明显优于 FPA，CS，WWO 的结果，并

且 CSMWWO 在函数 f_4 上获得了最小的标准差。

在函数 f_5 中，CSMWWO 获得了理论最优值 0，其平均求解精度比 FPA，CS，BA，WWO 分别高了 17 个、14 个、11 个、12 个数量级，且获得了最小的标准差。函数 f_5 有许多局部极小值点，是一个典型的非线性多模态函数，并且搜索空间比较广泛，一般的优化算法很难找到全局最优解，CSMWWO 在 f_5 上求解到最优精度，体现了 CSMWWO 处理此类问题的优势以及其较强的全局搜索能力。

在函数 f_6 中，CSMWWO 的最优值、中位数和标准差都是最小的，说明 CSMWWO 不仅求解精度高，而且稳定性也较好。

在 f_9-f_{12} 这 4 个低维的函数上，CSMWWO 都获得了最优解。在函数 f_9 上，FPA，CS，BA 和 CSMWWO 都获得了理论最优值且方差都为 0。在 f_{10} 上，各算法的最优精度一样，但是 CSMWWO 的方差最小，说明在求解 f_{10} 时 CSMWWO 稳定性较好。在 f_{11} 上，FPA，WWO，CSMWWO 求得的最优精度一样，WWO 的方差较小。f_{12} 是一个多峰函数，在函数 f_{12} 上 CSMWWO 获得了最优值，CS 取得了最小的标准差。通过 f_9-f_{12} 这 4 个低维函数的分析，说明 CSMWWO 在处理低维函数上也有一定的优势。

由以上分析可知，由于引入混沌策略和单纯形法，对于大部分测试函数来说，CSMWWO 的求解精度要优于 FPA，CS，BA 和 WWO。除了在函数 f_4 上，CSMWWO 的求解精度低于 BA，在其他函数上的求解精度都明显优于 FPA，CS，BA，WWO，并且在取得最优值的同时标准差较小，显示了 CSMWWO 较高的求解精度和稳定性。CSMWWO 在 f_2，f_5 这些多峰、具有多个局部极小值的函数上也获得了最优精度，体现了 CSMWWO 较好的全局搜索性能。

通过观察图 6—图 17 的收敛曲线，发现在收敛速度上，CSMWWO 在函数 f_2，f_3，f_4，f_6，f_9，f_{11}，f_{12} 上的收敛速度均快于 FPA；CSMWWO 在函数 f_2，f_3，f_4，f_5，f_6，f_9，f_{12} 上的收敛速度均快于 CS；CSMWWO 在函数 f_4，f_{12} 上的收敛速度快于 BA；CSMWWO 在函数 f_2，f_3，f_4，f_5，f_6，f_9，f_{10}，f_{11}，f_{12} 上的收敛速度均快于 WWO，说明改进后的算法在提高求解精度的同时也在一定程度上提高了算法的收敛速度。从图 7 可看出，CSMWWO 收敛曲线的下降速度比 WWO 快很多，而且很快找到了最优值。图 17 中，CSMWWO 的求解精度和收敛速度均优于其他算法。从图 8—图 11 可以看出，CSMWWO 的收敛速度明显快于 WWO，WWO 在一定的迭代中可能陷入局部最优，说明加入混沌策略和单纯形法之后能够帮助算法跳出局部最优，同时也能提高算法的求解精度。

结束语 文中针对基本水波优化算法 WWO 的求解精度和收敛速度还有待提高的问题，提出了一种基于混沌优化策略和单纯形法的水波优化算法 CSMWWO。该算法将混沌优化的特点和水波优化算法的优点相结合，不仅保持了水波优化算法易于实施的优点，而且从一定程度上降低了随机初始化的种群分布较差的问题。另外又引入了局部搜索能力较强的单纯形法，进一步提高了算法的求解精度和收敛速度。关于算法的参数优化（如波长衰减系数 α 和碎浪系数 β 等系数之间的取值范围和相互间的关系）、实际应用以及算法的拓扑结构优化等问题仍是研究的重点，也是下一步将要开展的工作。

参 考 文 献

[1] ZHENG Y J. Water wave optimization: A new nature-inspired metaheuristic [J]. Computers & Operations Research, 2015, 55: 1-11.

[2] The WAMDI Group. The WAM model-a third generation ocean wave prediction model [J]. Journal of Physical Oceanography, 1988, 18(12): 1775-1810.

[3] BOOIJ N, RIS R C, HOLTHUIJSEN L H, et al. A third-generation wave model for coastal regions, Part I, Model description and validation [J]. Journal of Geophysical Research Oceans, 1999, 104(C4): 7649-7666.

[4] WU X B, LIAO J, WANG Z C. Water Wave Optimization for the Traveling Saleman Problem [J]. Intelligent Computing Theories and Methodologies, 2015, 9225: 137-146.

[5] ZHANG B, ZHANG M X, ZHANG J F, et al. A Water Wave Optimization Algorithm with Variable Population Size and Comprehensive Learning [C]// Intelligent Computing Theories and Methodologies, 2015, 9225: 124-136.

[6] ZHENG Y J, ZHANG B. A simplified water wave optimization algorithm[C]// Evolutionary Computation. IEEE, 2015.

[7] AFRABANDPEY H, GHAFFARI M, MIRZAEI A, et al. A Novel Bat Algorithm Based on Chaos for Optimization Task [C]// Iranian Conference on Intelligent Systems. 2014: 1-6.

[8] NEUBURGER M, ROTZINGER M, KAISER H. A Simplex Method for Function Minimization [J]. Computer Journal, 1965, 7(4): 308-313.

[9] SADRI H. Nonlinear Dynamics and Chaos [J]. Journal of Statistical Physics, 2015, 78(5/6): 1635-1636.

[10] ALATAS B, AKIN E. Chaotically encoded particle swarm optimization algorithm and its applications [J]. Chaos Solitons & Fractals, 2009, 41(41): 939-950.

[11] ARUNKUMAR R, JOTHIPRAKAS V. Chaotic Evolutionary Algorithms for Multi-Reservoir Optimization [J]. Water Resources Management, 2013, 27(15): 5207-5222.

[12] CHEN H, ZHOU Y Q, ZHAO G W. Multi-population invasive weed optimization algorithm based on chaotic sequence [J]. Journal of Computer Applications, 2012, 32(7): 1958-1961. (in Chinese)
陈欢，周永权，赵光伟. 基于混沌序列的多种群入侵杂草算法 [J]. 计算机应用，2012，32(7)：1958-1961.

[13] BHARTI K K, SINGH P K. Chaotic gradient artificial bee colony for text clustering [J]. Soft Computing, 2016, 20(3): 1113-1126.

[14] MAY R M. Simple mathematical models with very complicated dynamics [J]. Nature, 1976, 261(5560): 459-467.

[15] WANG F, QIU Y H. A Novel Particle Swarm Algorithm Using the Simplex Method operator [J]. Information and Control,

2005，34(5)：517-522. (in Chinese)
王芳，邱玉辉. 一种引入单纯形法算子的新颖粒子群算法[J]. 信息与控制，2005，34(5)：517-522.

[16] CHEN X，ZHOU Y Q. Hybrid Algorithm Based on Monkey Algorithm and Simple Method [J]. Computer Science，2013，40(11)：248-254. (in Chinese)
陈信，周永权. 基于猴群算法和单纯法的混合优化算法[J]. 计算机科学，2013，40(11)：248-254.

[17] AMGED S，EL-WAKEEL. Design optimization of PM couplings using hybrid Particle Swarm Optimization-Simplex Method (PSO-SM) Algorithm [J]. Electric Power Systems Research，2014，116：29-35.

[18] REN X K，HAO R Z，SUN Z X，et al. Quantum Behaved Particle Swarm Optimization Algorithm Based on Simplex Method [J]. Microelectronics & Computer，2010，27(1)：154-157. (in Chinese)
任小康，郝瑞芝，孙正兴，等. 基于单纯形法的量子粒子群优化算法[J]. 微电子学与计算机，2010，27(1)：154-157.

[19] YANG X S. Flower pollination Algorithm for Global optimization [M]//Unconventional Computation and Natural Computation. 2012：240-249.

[20] YANG X S，DEB S. Cuckoo search via Lévy flights[C]//World Congress on Nature & Biologically Inspired Computing，2009 (NaBIC 2009). IEEE，2009：210-214.

[21] YANG X S. A New Metaheuristic Bat-Inspired Algorithm [J]. Science，2010，284：65-74.

(上接第 210 页)

表 3　论域 U 中所有对象的优势类

U	x 的优势类	U	x 的优势类
x_1	x_1, x_7	x_{11}	x_{11}, x_{16}
x_2	$x_2, x_7, x_8, x_9, x_{12}$	x_{12}	x_{12}
x_3	x_3, x_8, x_9	x_{13}	x_{13}
x_4	$x_4, x_5, x_6, x_7, x_8, x_9$	x_{14}	x_{13}, x_{14}
x_5	x_5	x_{15}	x_9, x_{15}
x_6	x_6, x_8	x_{16}	x_{16}
x_7	x_7	x_{17}	x_{17}
x_8	x_8	x_{18}	x_{18}
x_9	x_9	x_{19}	x_{19}
x_{10}	x_{10}	x_{20}	x_{20}

随机抽取一部分对象集 $X=\{x_1, x_2, x_3, x_7, x_8, x_{12}, x_{14}, x_{17}, x_{19}\}$ 作为研究对象，X 的上、下近似分别为：

$$\overline{R^{\geqslant}_{\beta\wedge k}}(X)=\{x_1, x_2, x_3, x_4\}$$

$$\underline{R^{\geqslant}_{\beta\wedge k}}(X)=\{x_1, x_2, x_5, x_7, x_8, x_{12}, x_{17}, x_{19}\}$$

进而，X 的正域、负域、上边界域、下边界域和边界域分别为：

$$posR^{\geqslant}_{\beta\wedge k}(X)=\{x_1, x_2\}$$

$$negR^{\geqslant}_{\beta\wedge k}(X)=\{x_6, x_9, x_{10}, x_{11}, x_{13}, x_{14}, x_{15}, x_{16}, x_{18}, x_{20}\}$$

$$UbnR^{\geqslant}_{\beta\wedge k}(X)=\{x_3, x_4\}$$

$$LbnR^{\geqslant}_{\beta\wedge k}(X)=\{x_5, x_7, x_8, x_{12}, x_{17}, x_{19}\}$$

$$bnR^{\geqslant}_{\beta\wedge k}(X)=\{x_3, x_4, x_5, x_7, x_8, x_{12}, x_{17}, x_{19}\}$$

结束语　本文引入加权得分函数并定义了一种新的排序规则，基于此排序规则定义了该排序规则下的直觉模糊序信息系统，通过“逻辑且”的方式把该序信息系统下的变精度粗糙集与程度粗糙集联合起来，使信息系统的量化更加精确。最后，通过董事会主席选举的实例对本文提出的定义与定理进行分析。文中工作有助于减小实际工程应用中因信息量化不全而导致的误差，从而提高信息量化的精度。

参 考 文 献

[1] 张文修，吴伟志，梁吉业，等. 粗糙集理论与方法[M]. 北京：科学出版社，2001.

[2] 杨善林，倪志伟. 机器学习与智能决策支持系统[M]. 北京：科学出版社，2004.

[3] PAWLAK Z. Rough Sets[J]. International Journal of Computer and Information Sciences，1982，11(5)：341-356.

[4] LIU W J，GUO Q. An Attribute Discretization Algorithm Based on Rough Sets and Fuzzy Sets[J]. Fuzzy Systems and Mathematics，2001，25(3)：147-153. (in Chinese)
刘文军，郭庆. 基于粗糙集与模糊集的连续值域决策表的离散化算法[J]. 模糊系统与数学，2001，25(3)：147-153.

[5] YANG X B，YU D J. Dominance-based rough set approach to incomplete interval-valued information system [J]. Data and Knowledge Engineering，2009，68(11)：1331-1347.

[6] ZHANG Z M. A rough set approach to intuitionistic fuzzy soft set based decision making [J]. Applied Mathematical Modelling，2012，36(10)：4605-4633.

[7] YEE L，MANFRED M F，WU W Z，et al. A rough set approach for the discovery of classification rules in interval-valued information systems [J]. International Journal of Approximate Reasoning，2008，47(1)：233-246.

[8] HU J H，CHEN X H. Multi-criteria decision making method based on dominance relation and variable precision rough set [J]. Systems Engineering and Electronics，2010，32(4)：759-763. (in Chinese)
胡军华，陈晓红. 基于优势关系和可变精度粗糙集的多准则决策方法[J]. 系统工程与电子技术，2010，32(4)：759-763.

[9] GRECO S，MATARAZZO B，SLOWINSKI R. Rough approximation by dominance relations [J]. International Journal of Intelligence Systems，2002，17(2)：153-171.

[10] GRECO S，MATARAZZO B，SLOWINSKI R. Rough sets theory for multi-criteria decision analysis [J]. European Journal of Operational Research，2001，129(1)：1-47.

[11] LUO G Z，YANG X J. Rough analysis model of multi-attribute decision making based on limited similarity dominance relation [J]. System Engineering：Theory and Practice，2009，29(9)：134-140.

[12] LUO G Z，YANG X B，YANG X J. Limited dominance - based rough fuzzy set and knowledge reductions [J]. Systems Engineering and Electronics，2010，32(8)：1657-1661. (in Chinese)
骆公志，杨习贝，杨晓江. 基于限制优势关系的粗糙模糊集及知识约简[J]. 系统工程与电子技术，2010，32(8)：1657-1661.

[13] ATANASSOV K. Fuzzy sets[J]. Fuzzy Sets and Systems，1986，20(1)：87-96.