文章编号:1000-2367(2023)05-0057-10 DOI:10.16366/j.cnki.1000-2367.2023.05.006

基于混合鸡群优化算法的 WSN 节点部署优化

韦修喜[a],郑宝峰[b]

(广西民族大学 a.人工智能学院;b.电子信息学院,南宁 530006)

摘 要:无线传感网络(Wireless Sensor Network,WSN)技术,在节点随机部署的情况下,容易出现节点分布不均的问题.为了提高节点部署的覆盖率,提出了一种基于混合鸡群优化算法(Hybrid Chicken Swarm Optimization Algorithm,HCSO)的 WSN 节点部署优化方法.首先为了平衡算法的全局搜索和局部搜索的能力,提出了一种自适应种群分配策略;其次,结合正余弦算法思想,改进了公鸡种群更新公式,提高其收敛速度和精度;最后,优化小鸡种群学习方式,使其不仅学习母鸡,也向公鸡和最优个体学习,提高小鸡粒子的质量.将 HCSO 算法在基准函数上测试,实验结果表明,本算法比其他算法的法收敛精度提高了 10%,收敛速度均比原算法提高了 0.05~0.10 s.最后将 HCSO 算法应用于 WSN 节点部署优化中,结果表明所提出的方法所得到的覆盖率高于其他算法,提高了 0.2~13.1 个百分点,充分证明了基于 HCSO 算法的 WSN 节点部署优化方法的优越性.

关键词:无线传感网络;节点部署;鸡群优化算法;自适应分配;正余弦算法

中图分类号:TP18 **文献标志码**:A

无线传感网络(WSN)是由一组轻量级、自主的空间分布节点组成,这些节点相互协调,形成一个感知特定应用任务的通信网络[1],具有成本低,适应度高等特点.随着科技的不断发展,科学家对于 WSN 的研究也越来越多,并已经应用到许多领域,如:路径调度目标跟踪、灾难预警和可穿戴设备[2]等.其中 WSN 的覆盖率问题是关键问题也是难点之一,覆盖率的大小反映了 WSN 网络的性能以及效率的好坏.在传统 WSN 网络中,传感器的部署位置大多是采用随机抛撒进行决定的,通过这种方法部署的网络容易出现节点分布不均的问题,使得一部分空间内的节点密度过高,从而导致网络的覆盖率降低,传感速度和效率变差.而解决这一问题就需要对节点部署方法进行优化.

随着群智能优化算法的出现,如:教与学优化算法(Teaching-learning-based Optimization,TLBO)[3],麻雀搜索算法(Sparrow Search Algorithm,SSA)[4],旗鱼优化算法(Sail Fish Optimizer,SFO)[5]等.近年来,群智能优化算法的研究人员越来越多,由于这类优化算法对于解决寻优问题有着显著的优点,许多国内外学者将群智能优化算法融到 WSN 节点部署优化中,如:MIAO 等[6]等将增强层次结构的灰狼优化算法应用其中,DEEPA 等[7]提出了利用 Lévy 飞行增强的鲸鱼优化算法,并应用到 WSN 覆盖优化中,并将 KNN(K-Nearest Neighbor,KNN)变体整合到 WSN 中,得到了效率更高、收敛度更好的覆盖优化方法.宋明智等[8]等提出了改进虚拟力-粒子群算法于 WSN 节点随机部署中的应用,最终实验表明其提出的方法覆盖率较其他优化算法有 3%~5%的提升.

虽然,将群智能优化算法运用到 WSN 节点部署优化中能使其覆盖率有一定的提升,但使用基本算法所得到的结果仍然还有许多不足,故这些算法还有许多可以提升和改进的空间.为了获得更好的 WSN 网络性能,还需要进一步开拓群智能优化算法的能力,更好地改善 WSN 节点部署优化问题,提高覆盖率.

鸡群优化算法(Chicken Swarm Optimization,CSO),于 2014 年由 MENG 等[9]提出,该算法作为群智能优化算法中的一员,一经提出就备受研究人员及学者关注.目前 CSO 算法及其改进算法已经应用到了光伏发

收稿日期:2022-06-21;**修回日期**:2022-07-21.

基金项目:国家自然科学基金(62266007;61662005);广西自然科学基金(2021GXNSFAA220068;2018GXNSFAA294068).

作者简介(通信作者):韦修喜(1980—),男,广西百色人,广西民族大学副教授,博士,研究方向为机器学习、计算智能,E-mail:weixiuxi@163.com.

电[10],电动汽车充电站规划[11],电网控制[12]等领域中,但应用到 WSN 节点部署优化的案例还不是很成熟,因此这一方向具有一定的研究意义.并且,WU 等[13]根据随机搜索算法的收敛准则,证明 CSO 算法满足两个收敛准则,保证了全局收敛性,但也指出了该算法的不足.由于算法的学习方法等原因,导致该算法在收敛速度和收敛精度上有所损失,同时易过早陷入局部最优解,因此 CSO 算法还有许多提升的空间.故 DEB 等[14]等将 TLBO 算法与 CSO 算法相结合,提出了一种基于教与学算法的鸡群优化算法.LIANG 等[15]提出了一种改进 CSO 算法用于机器人路径寻优,该算法引入了具有 Lévy 飞行特性的搜索策略和非线性递减策略,寻找到了较好的路径.虽然已经有许多学者和专家对 CSO 算法的改进和应用进行了大量的研究,但仍有许多开发的空间.

针对上述不足,本文提出了一种混合鸡群优化算法(HCSO)用于 WSN 节点部署优化的应用.

1 WSN 节点部署优化概念与模型

1.1 问题假设

假设 WSN 里面的每个节点对应一个传感器,传感器的覆盖范围都是半径固定的圆.假设每个传感器都可以实时感受到其感知半径内的其他传感器位置,并且所有的传感器的结构都是相同,这样就可以将 WSN 节点覆盖优化问题看为一个二维问题.若整个监测区域划为 $M\times N$ 个像素,如果 WSN 网络能够覆盖 x 个像素点,则覆盖率为$\frac{x}{M\times N}$.

1.2 节点覆盖模型

设监测区域内分布有 N 个传感器节点,每个节点的感知半径为 r,通信半径为 R,并且满足 $2r\leq R$.设 $T=\{t_1,t_2,\cdots,t_N\}$,其中 t_i 为第 i 个传感器,其位置坐标表示为(x_i,y_i),若第 j 个像素点为h_j,其位置坐标为(x_j,y_j). 则任意像素点与传感器之间的欧氏距离可用式(1)来表示:

$$d(t_i,h_j)=\sqrt{(x_i-x_j)^2+(y_i-y_j)^2}. \tag{1}$$

对于传感器与像素点的感知概率,本文使用二元感知模型进行建模,如式(2)表示:

$$p(t_i,h_j)=\begin{cases}1, & d(t_i,h_j)\leq r,\\ 0, & d(t_i,h_j)>r,\end{cases} \tag{2}$$

其中 $p(t_i,h_j)$ 为感知概率,r 为传感器的感知半径,若 $p(t_i,h_j)$ 为 1 则表示像素点 h_j 被传感器t_i 所覆盖,反之则没有被覆盖.

在整个监测范围中,会出现同一个像素点被不同的多个传感器同时感应到的现象,故需要进行联合感知概率的计算.像素点与多个传感器的联合感知概率如式(3)所展示:

$$p(T,h_j)=1-\prod_{t_i\in T}(1-p(t_i,h_j)), \tag{3}$$

其中 $p(T,h_j)$ 为联合感知概率. 最终,覆盖率由式(4)求出:

$$p_{\mathrm{cov}}=\frac{\sum_{j=1}^{M\times N}p(T,h_j)}{M\times N}, \tag{4}$$

其中 p_{cov}表示为覆盖率.综上所述,式(4)为 HCSO 应用到 WSN 节点部署优化的适应度函数,其目标为提高 p_{cov}数值.

2 鸡群优化算法

CSO 算法是通过观察鸡群的等级秩序以及鸡群的觅食行为而提出的.该算法中设置有公鸡、母鸡、小鸡 3 种等级不同的群体.其中公鸡群体的适应度最优,是其子群的领导者.小鸡群体的适应度最差,是种群中的弱势群体.剩下的粒子则为母鸡种群.假设整个鸡群中总共有 N 只鸡,则 $x_{i,j}^{t}$ 为第 t 次迭代中,第 i 只鸡在第 $j(j\in(1,D),D$ 为解的维度) 维度上的位置.

公鸡为适应度较好的前 N_R 只鸡,根据上述表示方法,公鸡的位置更新公式为:

$$x_{i,j}^{t+1}=x_{i,j}^{t}\times[1+N(0,\sigma^2)], \tag{5}$$

$$\sigma^2=\begin{cases}1, f_i\le f_k,\\ \exp(\dfrac{f_k-f_i}{|f_i|-\varepsilon}), f_i>f_k,\\ k\in[1,N_R], k\ne 1,\end{cases} \tag{6}$$

其中 $x_{i,j}^{t+1}$ 为下一次迭代中第 i 只鸡在第 j 维度上的位置;$N(0,\sigma^2)$ 代表均值为 0,方差为 σ^2 的正态分布;k 与 i 均代表一只公鸡,且 k 与 i 不相等,公鸡的适应度值分别由 f_i 和 f_k 对应;ε 为计算机中最小的常数,保证了式子分母不为零,防止出现错误.

N_H 代表母鸡的总数,母鸡的适应度适中,母鸡的位置更新公式如下所示:

$$x_{i,j}^{t+1}=x_{i,j}^{t}+S_1R_1(x_{r_1,j}^{t}-x_{i,j}^{t})+S_2R_2(x_{r_2,j}^{t}-x_{i,j}^{t}), \tag{7}$$

$$S_1=\exp(\frac{f_i-f_{r_1}}{abs(f_i)+\varepsilon}), \tag{8}$$

$$S_2=\exp(f_{r_2}-f_i), \tag{9}$$

其中,R_1 和 R_2 均为[0,1]的随机数;r_1 为与母鸡同一子群的公鸡;r_2 为另一只随机选取的鸡,且 $r_2\ne i$,$r_2\ne r_1$,$f_{r_2}<f_i$.

适应度最差的个体将被划分为小鸡种群,N_C 代表种群中小鸡的个数.由于小鸡是种群中的弱势群体,故指定其只能跟随鸡妈妈进行觅食.式(10)展示了这一现象的位置更新公式:

$$x_{i,j}^{t+1}=x_{i,j}^{t}+FL(x_{m,j}^{t}-x_{i,j}^{t}), \tag{10}$$

其中 $x_{m,j}^{t}$ 表示鸡妈妈的位置,FL 表示小鸡跟随其鸡妈妈的调节参数,通常为[0,2]上的随机数.

3　混合鸡群优化算法

3.1　自适应种群分配

CSO 算法是一个全局算法,为了加强其后期的局部搜索能力,本文加入了自适应种群分配策略.该策略使前期公鸡种群的粒子个数增加,随着迭代的过程,慢慢减少;而母鸡种群的粒子个数与之相反,是处于慢慢增加的状态.由于每 G 次迭代 CSO 会重新规划一次种群秩序,故种群数量的改变也是每 G 次进行一次.该策略的具体实现方法如式(11)所示:

$$\begin{cases}P_R=C_1\tan(\dfrac{\pi}{4}-\dfrac{t\pi}{4T})+P_{R_{\min}},\\ P_H=C_2\tan(\dfrac{\pi}{4}-\dfrac{t\pi}{4T})+P_{H_{\max}},\\ P_C=1-P_R-P_H,\\ P_R<P_H,\end{cases} \tag{11}$$

其中 P_R,P_H,P_C 分别代表每一个种群个体数量占鸡群总数量的百分比;t 代表当前迭代次数;T 代表最大迭代次数;C_1,C_2 用来控制每次迭代种群个体数量百分比增长或减小的速度;$P_{R_{\min}}$ 用来控制最终公鸡的数量比;$P_{H_{\max}}$ 用来控制母鸡的最终数量比,经实验测试 $P_{R_{\min}}$ 和 $P_{H_{\max}}$ 的取值与基本CSO算法的公鸡和母鸡占比数相同时算法的性能最优.

3.2　融合正余弦思想的公鸡位置更新优化

由于 CSO 算法中公鸡位置更新方法只是基于正态分布的变异方法,当算法的种群多样性变低时,会导致公鸡种群的个体难以跳出局部最优,从而使算法效率变慢,收敛精度变低.本文引入正余弦算法(Sine Cosine Algorithm,SCA)[16]的思想来优化公鸡位置更新公式,具体方法如下:

$$x_{i,j}^{t+1}=\begin{cases}x_{i,j}^{t}+\sigma^2r_1\sin(r_2)\,|\,r_3p_j^{t}-x_{i,j}^{t}\,|, r_4<0.5,\\ x_{i,j}^{t}+\sigma^2r_1\cos(r_2)\,|\,r_3p_j^{t}-x_{i,j}^{t}\,|, r_4\ge 0.5,\end{cases} \tag{12}$$

$$r_1 = a\left(1 - \frac{t}{T}\right), \tag{13}$$

其中 r_1 控制迭代方向,r_2 控制移动距离,r_3 控制最优解的影响力,r_4 用来选择使用正弦还是余弦更新位置;$r_2 \in [0,2\pi]$,$r_3 \in [0,2]$,$r_4 \in [0,1]$ 且都为随机数;p_j^t 为第 t 次迭代中第 j 维的最优解;a 为正常数.

3.3 改进小鸡更新思想

由于 CSO 算法中,小鸡向其母亲学习,若其母亲陷入局部最优解,小鸡也会随之学习,从而降低算法整体的效率.故本文使小鸡向其双亲学习,并增加突变的可能,学习全局最优个体.引进随机交叉变异思想,改进后的小鸡位置更新公式如下:

$$x_{i,j}^{t+1} = \begin{cases} \sqrt{FL}\,x_{i,j}^t + \omega(1-\sqrt{FL})(Q(x_{m,j}^t - x_{i,j}^t) + (1-Q)(x_{r,j}^t - x_{i,j}^t)), r < 0.5, \\ \sqrt{FL}\,x_{i,j}^t + \omega(1-\sqrt{FL})(Q(x_{m,j}^t - x_{i,j}^t) + (1-Q)(p_j^t - x_{i,j}^t)), r \geq 0.5, \end{cases} \tag{14}$$

$$\omega = \frac{t^3}{T^3}, \tag{15}$$

其中 $FL \in [0,2]$ 为随机交叉变异的调节参数;Q 是$[-1,1]$ 的随机数,该参数为学习调控因子,能够同时调控两边的学习力度;ω 是一个自适应权重系数;r 为$[0,1]$ 之间的随机数,控制小鸡是否发生突变.改进后的小鸡更新位置公式使小鸡能够更好地跳出局部最优解,提高适应度,提高算法的效率.

3.4 HCSO 节点部署优化流程

步骤 1 初始化 WSN 节点参数和 HCSO 算法参数,如:节点个数 N,感知半径 r,粒子个数 pop,最大迭代次数等.

步骤 2 初始化粒子位置,并使用式(1)~(4)计算覆盖率.

步骤 3 判断 mod(t,G)是否为 1,是则根据式(11)进行种群规划;否则进行步骤 4.

步骤 4 每个粒子根据其被划分的种群,按照式(7)~(9),式(12)~(15)进行位置更新.

步骤 5 算法若到最大迭代次数,则输出优化后的覆盖率等结果;算法若没达到最大迭代次数,跳至第 3 步进行下一次的迭代.

3.5 复杂度分析

假设种群中总粒子数量为 N,公鸡数量为 N_R,小鸡数量为 N_C,解空间维度为 D,T 为最大迭代次数.HCSO 引入了自适应种群分配策略,每 G 次更改一次种群比例,则计算种群百分比的时间复杂度为 $O(T/G)$;改进后的公鸡位置更新公式时间复杂度为 $O(N_RDT)$;修改了小鸡的学习策略,其时间复杂度为 $O(N_CDT)$,本算法的时间复杂度与原算法相同.

4 仿真实验设计与分析

4.1 基准函数测试

4.1.1 实验设计

为了验证 HCSO 算法的有效性,本文选取了 6 个具有代表性的基准测试函数,其中函数 f_1 为简单的单峰函数,能够测试算法的收敛速度;函数 f_2 为高维单峰函数,能够测试算法的收敛精度和收敛速度;函数 f_3 是不连续阶梯函数,该函数能够对算法的有效性进行测试;函数 $f_4 \sim f_6$ 为 3 个不同的多峰函数,能够很好地检验算法的全局搜索能力.通过这些函数对 HCSO 进行测试,并与 PSO[17],CSO,SCA,GWO[18],ICSO[15] 进行结果对照及分析,表 1 中给出了函数的具体表达式.每个算法在其搜索空间中共有 30 个粒子进行搜索,单独在 6 个基准函数上运行 30 次(每次迭代 1 000 次)所统计出来其优化结果进行比较.各算法的具体参数设计如表 1 所示,基准函数表达式如表 2 所示.

4.1.2 测试结果分析

图 1、图 2 为每个算法在 $f_1 \sim f_6$ 的函数收敛图和箱线图,附表 Ⅰ 为每个算法在 $f_1 \sim f_6$ 单独运行 30 次(每次迭代 1 000 次) 所统计出来的数据表. 通过 3 种不同的展示方法可以明显看出,HCSO 与原算法相比,收敛精度、收敛速度和跳出局部最优能力都有提高.

表 1　算法参数表

Tab. 1　Algorithm parameter table

算法名称	参数设置
PSO	$c_1=c_2=1.5,\omega=0.8$
CSO	$N_R=0.15N,N_H=0.7N,N_M=0.5N_H,N_C=N-N_R-N_H,G=10,F\in[0,2]$
SCA	$r_2=[0,2\pi],r_3\in[0,2],r_4\in[0,1]$
GWO	$r_1\in[0,1],r_2\in[0,1]$
ICSO	$N_R=0.15N,N_H=0.7N,N_M=0.5N_H,N_C=N-N_R-N_R,G=10,F\in[0,2]$
HCSO	$G=10,F\in[0,2],P_{r\min}=0.15,P_{h\max}=0.7,C_1=13,C_2=-8$

表 2　基准函数表

Tab. 2　Benchmark function table

基准函数	范围	维度	最优值
$f_1(x)=\sum_{i=1}^{D}x_i^2$	$x_i\in[-100,100]$	30	0
$f_2(x)=\sum_{i=1}^{D}\lvert x_i\rvert+\prod_{i=1}^{D}\lvert x_i\rvert$	$x_i\in[-10,10]$	30	0
$f_3(x)=\sum_{i=1}^{D}(\lvert x_i+0.5\rvert)^2$	$x_i\in[-1.28,1.28]$	30	0
$f_4(x)=\sum_{i=1}^{D}[x_i^2-10\cos(2\pi x_i)+10]$	$x_i\in[-5.12,5.12]$	30	0
$f_5(x)=-20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{D}x_i^2})-\exp(\frac{1}{n}\sum_{i=1}^{D}\cos(2\pi x_i))+20+e$	$x_i\in[-32,32]$	30	0
$f_6(x)=\frac{1}{4\,000}\sum_{i=1}^{D}(x_i^2)-\prod_{i=1}^{D}\cos(\frac{x_i}{\sqrt{i}})+1$	$x_i\in[-600,600]$	30	0

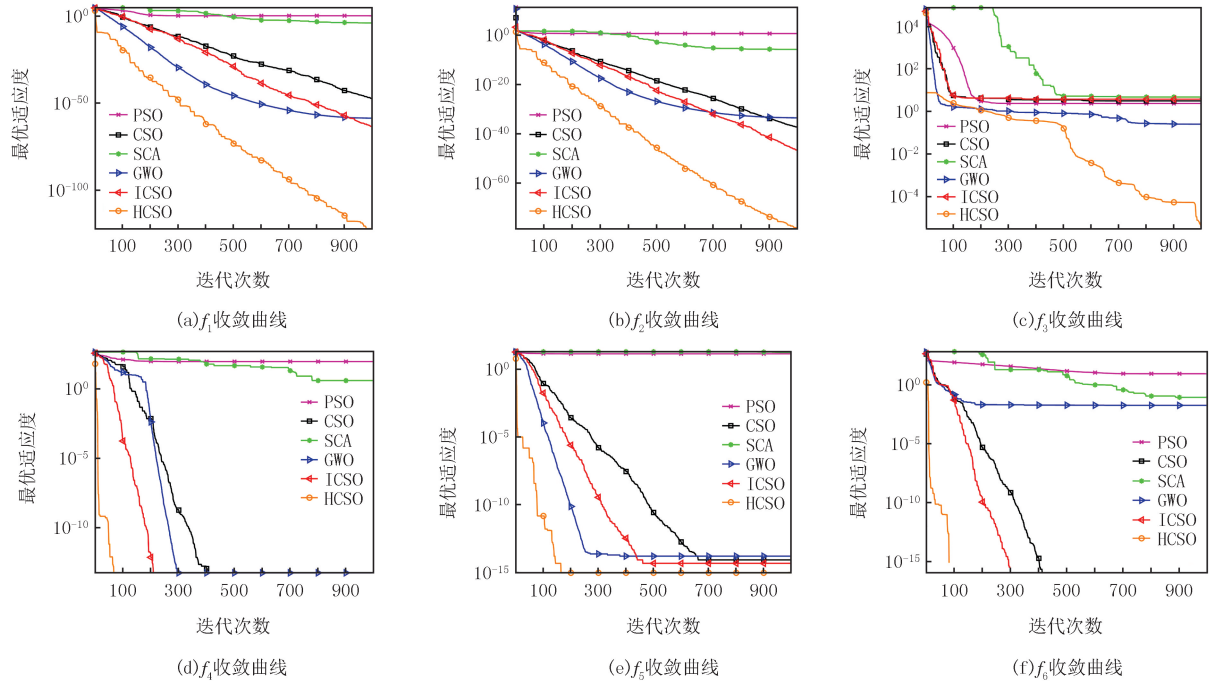


图1　基准函数优化收敛曲线

Fig. 1　Baseline function optimization convergence curve

在单峰测试函数,图 1(a-c)中可以看到虽然本算法没有优化到理想最优值,但相较于其他算法,HCSO 的精度都是最好的.GWO 虽然能快速地收敛,但都陷入了局部最优.附表Ⅰ中黑体部分为 HCSO 运算结果,可以看出 PSO 和 SCA 运行速度非常快,但他们的精度却不高,虽然 HCSO 的运行速度总体比 PSO 和 SCA

的运行速度慢,但这是由 CSO 算法本身的结构造成的,HCSO 已经比其原算法有所减少.故本算法对于单峰函数优化起到了比较好的效果.

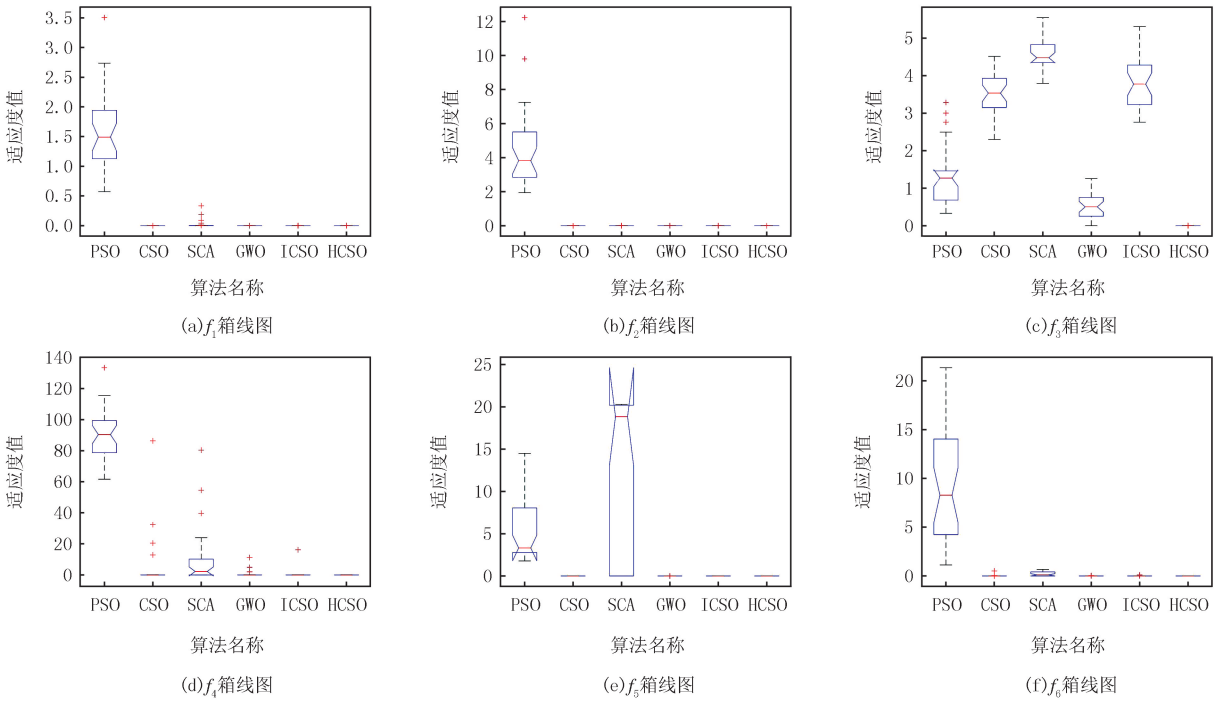


图2 基准函数优化箱线图

Fig.2 Benchmark function optimization boxplot

在多峰函数测试中,可以在箱线图中看出,HCSO 是最稳定的,且没有离群值.对于函数 f_4 和 f_5,HCSO 已经找到了理论最优值,并且十分稳定.附表Ⅰ中可以看到其平均值和标准差都是 0,说明独立运行的 30 次每次都能找到局部最优.虽然其他算法也能够找到,但是并不稳定.在图 1(d)和图 1(f)中可以看到,HCSO 几乎在不到 100 次迭代就可以找到理论最优值,故其收敛速度也很优秀.因此可以看出,HCSO 对于多峰函数的优化也很出色,其收敛速度和精度都有大幅度提高,并且稳定性较好.

由图 2 的算法箱线图可以看出,改进后的 HCSO 算法对于单峰基准函数和多峰基准函数的优化结果都很稳定,没有较大的离群值和标准差.同时在表 3 中也可以看到 HCSO 在每个算法中的标准差都是最小的,因此 HCSO 的改进也加强了算法的稳定性.

综上所述,本文提出的 HCSO 算法在测试中表现的性能均高于 PSO,CSO,SCA,GWO,ICSO.其收敛速度和收敛精度以及稳定性非常出色,所耗费的时间大部分也与原算法相仿甚至优于原算法,验证了改进的有效性.

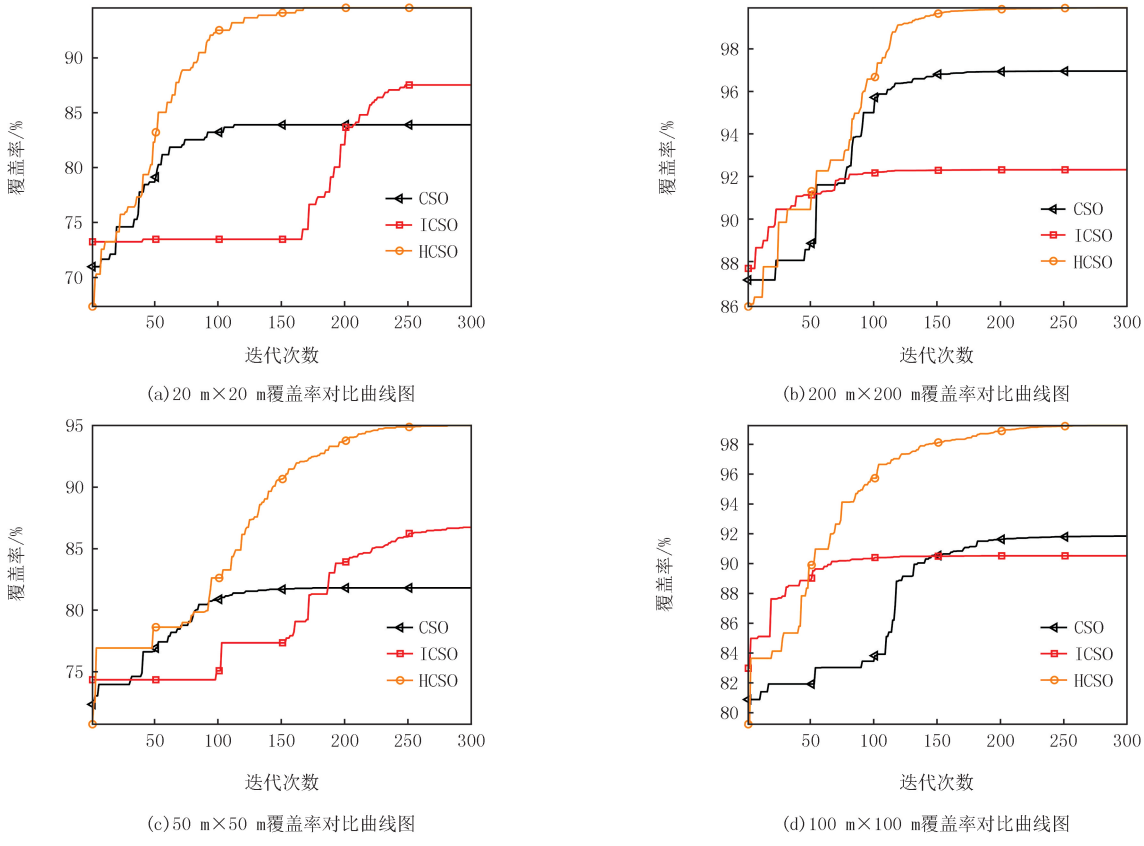
4.2 WSN 节点部署优化测试

由于函数测试中的算法应用到 WSN 节点部署优化中其表现的性能较差,无法充分验证 HCSO 算法在 WSN 节点部署优化中的性能和效果,因此在此测试中,选择 CSO,ICSO 算法和文献[19－21]中的算法和模型的覆盖实验进行对比,并且设置的实验模型环境与这些文献中的实验模型环境相同.

设需要覆盖区域是一个 $S=20\ \text{m}\times 20\ \text{m}$ 的二维平面,内有 $N=24$ 个传感器节点,其感知半径为 $r=2.5\ \text{m}$,通讯半径为 $R=5\ \text{m}$.覆盖率对比曲线如图 3(a)所示.HCSO 优化节点分布图如图 4(a)所示.

表 3 展示了在本场景中不同算法优化后得到的结果,其中随机抛撒所获得的覆盖率为 67.35%,说明这种情况下,传感器无法均匀分布在整个空间中,有许多范围没有被覆盖到.使用 CSO 算法优化后,覆盖率达到了 83.90%,使用 ICSO 算法优化后,覆盖率达到了 87.30%,使用这两种方法比随机抛撒策略覆盖率有了一定的提高,但仍有一定的优化空间.文献[19]的 IWOA 算法,将覆盖提升到了 93.56%,分布较为均匀.HCSO 算法改进了公鸡收敛公式,使其能够学习 SCA 算法的更新方法,使得算法中的优秀个体能够更好地找到最优值.同时改变小鸡学习方式,提高了算法整体粒子质量,使得算法所获得的结果精度更高.从实验中可

以看到，HCSO 算法的覆盖率达到了 94.56%，较随机抛撒提高了 27.21 个百分点，极大程度地解决了 WSN 节点分布不均匀问题.



(a)20 m×20 m覆盖率对比曲线图　(b)200 m×200 m覆盖率对比曲线图

(c)50 m×50 m覆盖率对比曲线图　(d)100 m×100 m覆盖率对比曲线图

图3　覆盖率对比曲线图

Fig.3　Coverage comparison curve

设需要覆盖区域是一个 $S=200$ m×200 m 的二维平面，内有 $N=60$ 个传感器节点，其感知半径为 $r=20$ m，通讯半径为 $R=40$ m.节点优化后的结果如表 4，覆盖率对比曲线如图 3(b)所示.HCSO 优化节点分布图如图 4(b)所示.

表 3　优化结果对比

Tab. 3　Comparison of optimization results

算法	随机抛撒	IWOA[19]	CSO	ICSO	HCSO
覆盖率/%	67.35	93.56	83.90	87.30	94.56

表 4　200×200 优化结果对比

Tab.4　200×200 Comparison of optimization results

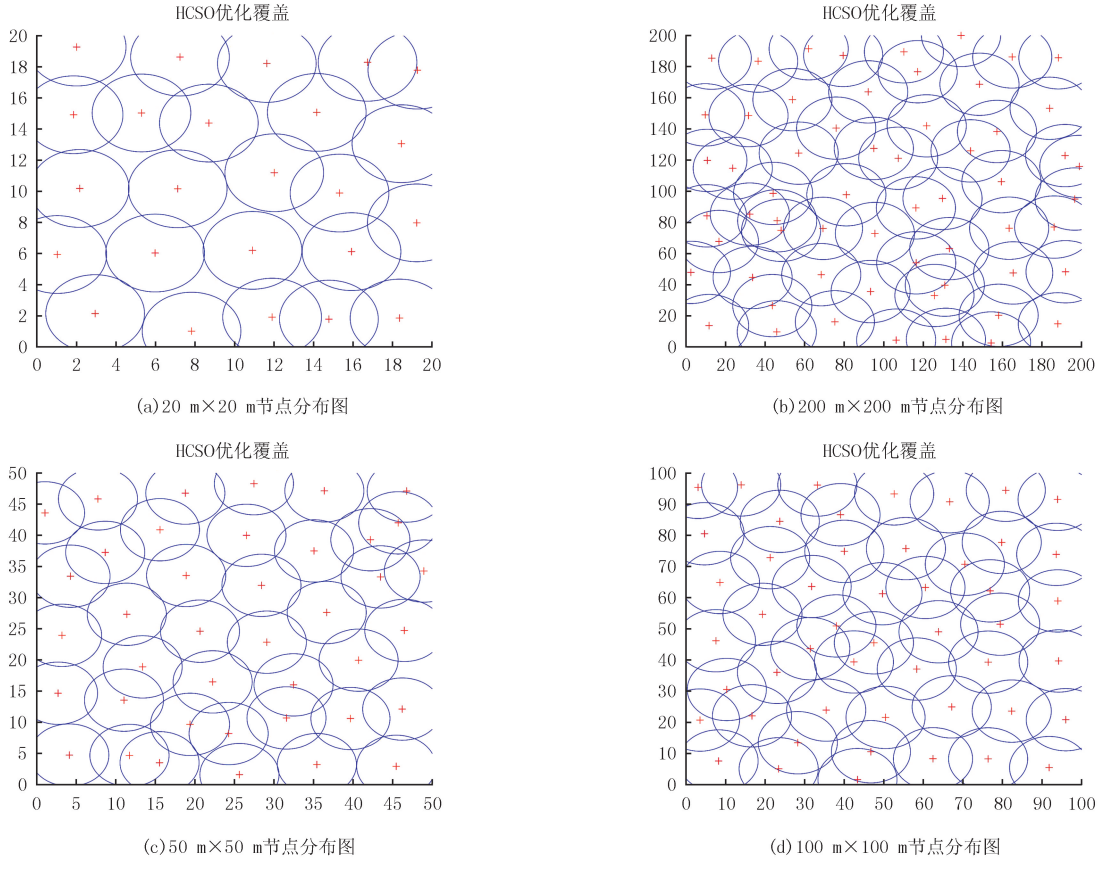
算法	随机抛撒	IWOA[19]	CSO	ICSO	HCSO
覆盖率/%	87.29	99.72	96.97	92.32	99.93

在本场景下，HCSO 较 CSO 和 ICSO 的覆盖率分别提升了 2.96 和 7.61 个百分点，与 IWOA 所获得的覆盖率相差较小，但 HCSO 的精度更高.通过图 3(b)中的 HCSO 优化覆盖图可以看出，HCSO 算法优化的 WSN 传感网络已经几乎达到了全覆盖.

设需要覆盖区域是一个 $S=50$ m×50 m 的二维平面，内有 $N=40$ 个传感器节点，其感知半径为 $r=5$ m，通讯半径为 $R=10$ m.节点优化后的结果如表 5，覆盖率对比曲线如图 3(c)所示.HCSO 优化节点分布图如图 4(c)所示.

根据表 5 的数据可以看到，HCSO 所获得的覆盖率相比随机抛撒最终提高了 22.80 个百分点.HCSO 算

法在迭代 200 次时,所用时间仅比 CSO 算法慢 0.08 s.所获得的覆盖率比 IWOA 算法提高了 7.27 个百分点,且速度要比 IWOA 算法快得多.与 HPSBA 相比,虽然覆盖率在迭代 200 次不如 HPSBA 所得到的结果好,但 HPSBA 迭代 200 次所花费的时间比 HCSO 算法迭代 300 次所花费的时间还要多,最终 HCSO 算法所得到的覆盖率比 HPSBA 提高了 0.07 个百分点.HCSO 算法使用自适应种群调整策略,较好平衡了算法局部搜索和全局搜索能力,提高了算法的运行速度.迭代次数调整到 300 次,HCSO 算法的运行速度是最快的,同时其获得的覆盖率也是精度最高的.



(a)20 m×20 m节点分布图 (b)200 m×200 m节点分布图

(c)50 m×50 m节点分布图 (d)100 m×100 m节点分布图

图4 HCSO优化节点分布图

Fig.4 HCSO optimizes node distribution

表 5 50 m×50 m 优化结果对比

Tab. 5 50 m×50 m Comparison of optimization results

算法	200		300	
	覆盖率/%	时间/s	覆盖率/%	时间/s
随机抛撒	73.20	—	74.93	—
IWOA[19]	86.62	55.31	90.81	81.43
HPSBA[21]	94.93	20.68	—	—
CSO	81.81	15.27	81.81	21.3
ICSO	83.86	15.89	87.51	21.82
HCSO	94.89	15.35	95	20.38

表 6 100 m×100 m 优化结果对比

Tab. 6 100 m×100 m Comparison of optimization results

算法	200		300	
	覆盖率/%	时间/s	覆盖率/%	时间/s
随机抛撒	79.64	—	77.20	—
ESCA[20]	98.58	—	—	—
IWOA[19]	97.92	100	99.21	158.6
CSO	91.96	75.83	92.98	103.98
ICSO	90.55	73.14	90.55	104.82
HCSO	98.90	66.75	99.26	100.29

设需要覆盖区域是一个 $S=100$ m×100 m 的二维平面,内有 $N=50$ 个传感器节点,其感知半径为 $r=10$ m,通讯半径为 $R=20$ m.节点优化后的结果如表 6,覆盖率对比曲线如图 3(d)所示.HCSO 优化节点分布图如图 4(d)所示.

在文献[20]中，没有给出 ESCA 算法的迭代次数和迭代时间的数据.根据覆盖率可以看到，最终 HCSO 算法所获得的覆盖率比 ESCA 算法提高了 0.68 个百分点.在迭代次数为 200 时，随机抛撒所得到的覆盖率仅达到了 79.64%，说明存在大部分节点冗余，节点分布不均匀.使用 CSO 算法和 ICSO 算法优化后，覆盖率分别为 91.96%和 90.55%，起到了一定的优化效果.文献[19]中的 IWOA 算法优化后，大幅度提高了节点覆盖率，达到了 97.92%，但仍比 HCSO 算法优化后的覆盖率低.HCSO 算法优化后覆盖率达到了 98.90%，同时 HCSO 算法的运行速度也是最快的，耗时为 66.75 s.迭代次数调整到 300 后，HCSO 算法仍然能够继续寻找更好的分配策略，最终使 WSN 节点覆盖率达到了 99.26%，从图 4(b)中也可以看到，节点分布比较均匀，很好地解决了节点冗余问题.

5　结　论

本文针对在传统无线传感网络中，使用随机抛撒策略对传感器节点进行部署，会出现被覆盖区域的节点分布不均匀，覆盖率低等问题.提出了一种混合鸡群优化算法.该算法加入了自适应种群分配策略，提高算法的前后期搜索速度及精度；又利用正余弦算法改进 CSO 公鸡的更新策略，提高了跳出局部最优解的能力；最后更新小鸡学习策略，使其学习其双亲以及最优个体，还能够交叉突变，提高小鸡群体的质量，从而提高算法整体质量.通过 6 个基准函数测试，比对了 HCSO 算法与其他算法所获得的优化结果.表明 HCSO 算法均取得了比较理想的适应度值，加快了运行速度和收敛速度.最后将 HCSO 算法应用到 WSN 节点部署优化中去，首先与原算法比较基本都能提高 2～10 个百分点，与其他改进算法进行比较，也能够对不同环境下的节点覆盖率进行一定的优化提高，增强了 WSN 网络的性能.后续将对多目标的 CSO 算法进行研究，并准备运用到多目标 WSN 节点部署问题中.

附　录

附表Ⅰ见电子版(DOI:10.16366/j.cnki.1000-2367.2023.05.006).

参　考　文　献

[1] SHARMA A,CHAUHAN S.Analytic evaluation of non-uniformities for coverage probability computation of randomly deployed wireless sensor network[J].International Journal of Sensor Networks,2020,34(1):1-14.

[2] WANG J,JU C W,GAO Y,et al.A PSO based energy efficient coverage control algorithm for wireless sensor networks[J].Computers, Materrals & Continua,2018(9):433-446.

[3] BAYKASOĞLU A,HAMZADAYI A,KÖSE S Y.Testing the performance of teaching-learning based optimization(TLBO)algorithm on combinatorial problems:flow shop and job shop scheduling cases[J].Information Sciences,2014,276:204-218.

[4] XUE J K,SHEN B.A novel swarm intelligence optimization approach:sparrow search algorithm[J].Systems Science & Control Engineering,2020,8(1):22-34.

[5] SHADRAVAN S,NAJI H R,BARDSIRI V K.The Sailfish Optimizer:a novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems[J].Engineering Applications of Artificial Intelligence,2019,80:20-34.

[6] MIAO Z M,YUAN X F,ZHOU F Y,et al.Grey wolf optimizer with an enhanced hierarchy and its application to the wireless sensor network coverage optimization problem[J].Applied Soft Computing,2020,96:106602.

[7] DEEPA R,VENKATARAMAN R.Enhancing Whale Optimization Algorithm with Levy Flight for coverage optimization in wireless sensor networks[J].Computers & Electrical Engineering,2021,94:107359.

[8] 宋明智，杨乐.改进 VFPSO 算法于 WSN 节点随机部署中的应用[J].计算机工程与应用，2016，52(2)：141-145.
SONG M Z,YANG L.Random deployment of sensor nodes using enhanced VFPSO algorithm[J].Computer Engineering and Applications,2016,52(2):141-145.

[9] MENG X B,LIU Y,GAO X Z,et al.A new bio-inspired algorithm:chicken swarm optimization[C]//International Conference in Swarm Intelligence.Cham:Springer,2014:86-94.

[10] LIU Z F,LI L L,TSENG M L,et al.Prediction short-term photovoltaic power using improved chicken swarm optimizer-Extreme learning machine model[J].Journal of Cleaner Production,2020,248:119272.

[11] SACHAN S,DEB S,SINGH S N,et al.Planning and operation of EV charging stations by chicken swarm optimization driven heuristics [J].Energy Conversion and Economics,2021,2(2):91-99.

[12] OTHMAN A M,EL-FERGANY A A.Adaptive virtual-inertia control and chicken swarm optimizer for frequency stability in power-grids penetrated by renewable energy sources[J].Neural Computing and Applications,2021,33(7):2905-2918.
[13] WU D H,XU S P,KONG F.Convergence analysis and improvement of the chicken swarm optimization algorithm[J].IEEE Access,2016,4:9400-9412.
[14] DEB S,GAO X Z,TAMMI K,et al.A new teaching-learning-based chicken swarm optimization algorithm[J].Soft Computing,2020,24(7):5313-5331.
[15] LIANG X M,KOU D C,WEN L.An improved chicken swarm optimization algorithm and its application in robot path planning[J].IEEE Access,2020,8:49543-49550.
[16] MIRJALILI S.SCA:a Sine Cosine Algorithm for solving optimization problems[J].Knowledge-Based Systems,2016,96:120-133.
[17] VENTER G,SOBIESZCZANSKI-SOBIESKI J.Particle swarm optimization[J].AIAA Journal,2003,41(8):1583-1589.
[18] MIRJALILI S,MIRJALILI S M,LEWIS A.Grey wolf optimizer[J].Advances in Engineering Software,2014,69:46-61.
[19] 宋婷婷,张达敏,王依柔,等.基于改进鲸鱼优化算法的 WSN 覆盖优化[J].传感技术学报,2020,33(3):415-422.
SONG T T,ZHANG D M,WANG Y R,et al.WSN coverage optimization based on improved whale optimization algorithm[J].Chinese Journal of Sensors and Actuators,2020,33(3):415-422.
[20] 何庆,徐钦帅,魏康园.基于改进正弦余弦算法的无线传感器节点部署优化[J].计算机应用,2019,39(7):2035-2043.
HE Q,XU Q S,WEI K Y.Enhanced sine cosine algorithm based node deployment optimization of wireless sensor network[J].Journal of Computer Applications,2019,39(7):2035-2043.
[21] 张孟健,汪敏,王霄,等.混合粒子群-蝴蝶算法的 WSN 节点部署研究[J].计算机工程与科学,2022,44(6):1013-1022.
ZHANG M J,WANG M,WANG X,et al.A hybrid particle swarm-butterfly algorithm for WSN node deployment[J].Computer Engineering & Science,2022,44(6):1013-1022.

Node deployment optimization of wireless sensor network based on hybrid chicken swarm optimization algorithm

Wei Xiuxi[a], Zheng Baofeng[b]

(a. College of Artificial Intelligence; b. College of Electronic Information, Guangxi Minzu University, Nanning 530006, China)

Abstract: Wireless Sensor Network(WSN)technology, with random node deployment, is prone to the problem of uneven node distribution. To improve the coverage of node deployment, this paper proposes a Hybrid Chicken Swarm Optimization Algorithm(HCSO)based WSN node deployment optimization method. Firstly, an adaptive population allocation strategy is proposed to balance the ability of global search and local search of the algorithm; secondly, by combining the idea of Sine and Cosine Algorithm, the rooster population update formula is improved to improve its convergence speed and accuracy; finally, the chick population learning method is optimized so that they not only learn from the hen but also learn from the rooster and the optimal individual to improve the quality of chick particles. The HCSO algorithm is tested on the benchmark function, and the experimental results show that the present algorithm improves the accuracy of the method convergence by 10% compared with other algorithms, and the convergence speed is all improved by 0.05-0.10 seconds compared with the original algorithm. Finally, the HCSO algorithm is applied to the optimization of WSN node deployment, and the results show that the coverage obtained by the method proposed in this paper is higher than other algorithms by 0.2-13.1 percentage points, which fully proves the superiority of the WSN node deployment optimization method based on the HCSO algorithm.

Keywords: wireless sensor network; node deployment; chicken swarm optimization algorithm; adaptive allocation; sine cosine algorithm

[责任编校 陈留院 赵晓华]

附表Ⅰ　优化结果对比

Attached tab. Ⅰ　Comparison of optimization results

函数	算法	最差值	最优值	平均值	标准差	运行时间/s
$f_1(x)$	PSO	3.35E-0	3.42E-1	1.27E-0	6.50E-1	0.091 617
	CSO	1.82E-39	9.5E-51	9.05E-41	3.51E-40	0.314 918
	SCA	5.06E-1	1.68E-7	2.78E-2	9.16E-2	0.166 028
	GWO	5.39E-58	1.91E-61	6.57E-59	1.37E-58	0.203 582
	ICSO	1.54E-38	1.99E-48	1.02E-39	3.21E-39	0.479 996
	HCSO	**4.3E-118**	**3.6E-125**	**3.3E-119**	**1.04E-118**	0.294 131
$f_2(x)$	PSO	14.60E-0	1.87E-0	4.45E-0	2.58E-0	0.098 943
	CSO	3.56E-36	1.48E-41	2.58E-37	8.17E-37	0.336 216
	SCA	6.82E-4	3.6E-8	4.36E-5	1.25E-4	0.189 376
	GWO	1.47E-33	1.30E-35	1.46E-34	2.65E-34	0.218 990
	ICSO	1.51E-31	2.1E-37	1.37E-32	3.32E-32	0.486 538
	HCSO	**1.9E-79**	**1.09E-82**	**2.85E-80**	**4.74E-80**	0.289 937
$f_3(x)$	PSO	2.77E-0	3.39E-1	1.22E-0	5.39E-1	0.117 926
	CSO	4.30E-0	2.67E-0	3.52E-0	4.47E-1	0.314 825
	SCA	6.20E-0	3.82E-0	4.56E-0	4.20E-1	0.178 802
	GWO	1.50E-0	2.46E-1	6.20E-0	3.33E-1	0.222 535
	ICSO	4.84E-0	2.73E-0	3.83E-0	5.39E-1	0.498 201
	HCSO	**2.77E-4**	**1.82E-8**	**2.52E-5**	**5.98E-5**	0.242 662
$f_4(x)$	PSO	1.25E+2	5.33E+1	8.51E+1	1.92E+1	0.152 572
	CSO	1.73E+1	0	5.77E-1	3.11E-0	0.347 105
	SCA	9.91E+1	9.24E-7	1.23E+1	2.07E+1	0.206 871
	GWO	4.49E-0	0	1.4E-1	8.07E-1	0.232 409
	ICSO	0	0	0	0	0.510 847
	HCSO	**0**	**0**	**0**	**0**	0.302 644
$f_5(x)$	PSO	1.48E+1	1.61E-0	6.58E-0	4.82E-0	0.128 758
	CSO	7.99E-15	4.44E-15	6.45E-15	1.76E-15	0.306 241
	SCA	2.03E-0	6.04E-4	1.43E+1	8.59E-0	0.195 97
	GWO	2.22E-14	1.15E-14	1.59E-14	2.54E-15	0.250 576
	ICSO	7.99E-15	4.44E-15	5.98E-15	1.76E-15	0.465 964
	HCSO	**8.88E-16**	**8.88E-16**	**8.88E-16**	**9.86E-32**	0.294 468
$f_6(x)$	PSO	3.39E+1	1.74E-0	8.46E-0	6.70E-0	0.108 225
	CSO	0	0	0	0	0.362 16
	SCA	0	0	0	0	0.227 651
	GWO	2.43E-2	0	1.09E-3	4.57E-3	0.256 250
	ICSO	0	0	0	0	0.538 848
	HCSO	**0**	**0**	**0**	**0**	0.327 673