

方波触发勘探与开发的粒子群优化算法

邓志诚^{1,2} 孙辉^{1,2,3} 赵嘉^{1,2,3} 王晖^{1,2,3} 吕莉^{1,2,3} 谢海华^{1,2}

摘要 在粒子群优化算法中,当勘探时间持续过长,将可能导致种群在解空间过度徘徊;种群在开发阶段陷入局部最优后,难以再次进行全局勘探.针对上述问题,提出方波触发勘探与开发的粒子群优化算法.依据方波的周期特性,在前半个周期内使用标准粒子群优化算法执行全局勘探,后半周期使用改进的更新公式执行局部开发.经过实验验证,在方波触发机制下,通过为粒子提供多步长,可达到周期性触发勘探与开发的目的.使用多类型测试函数,将该算法与新改进粒子群算法、改进人工蜂群算法、改进差分算法在 30、50 和 100 维下比较,实验结果表明,该算法在收敛速度和精度上更具优势.

关键词 粒子群优化算法,方波触发机制,勘探与开发,全局优化

引用格式 邓志诚,孙辉,赵嘉,王晖,吕莉,谢海华.方波触发勘探与开发的粒子群优化算法.自动化学报,2022,48(12):3042-3061

DOI 10.16383/j.aas.c190842



开放科学(资源服务)标识码(OSID):

Particle Swarm Optimization With Square Wave Triggered Exploration and Exploitation

DENG Zhi-Cheng^{1,2} SUN Hui^{1,2,3} ZHAO Jia^{1,2,3} WANG Hui^{1,2,3} LV Li^{1,2,3} XIE Hai-Hua^{1,2}

Abstract In the particle swarm optimization, when the exploration time continues to be too long, it may cause the particles to be excessively lingered in the solution space. After the population has fallen into local optimum during the exploitation stage, it is difficult to conduct global exploration again. To solve these two problems, an improved particle swarm optimization called particle swarm optimization with square wave triggered exploration and exploitation is proposed. According to the periodic characteristics of the square wave, the exploration is performed using the standard particle swarm optimization equation in the first half of the period, and the exploitation is performed using the improved update equation in the second half of the period. After experimental verification, under the square wave triggered mechanism, by providing variable step sizes to particles, the purpose of triggering exploration and exploitation periodically is achieved. The square wave triggered exploration and exploitation was compared with other state-of-the-art intelligent algorithms, numerical experiment results on several types of benchmark functions of dimensions 30, 50 and 100 showed that the square wave triggered exploration and exploitation had more advantages in terms of accuracy and convergence speed.

Key words Particle swarm optimization algorithm, square wave triggered mechanism, exploration and exploitation, global optimization

Citation Deng Zhi-Cheng, Sun Hui, Zhao Jia, Wang Hui, Lv Li, Xie Hai-Hua. Particle swarm optimization with square wave triggered exploration and exploitation. *Acta Automatica Sinica*, 2022, 48(12): 3042-3061

收稿日期 2019-12-11 录用日期 2020-05-03

Manuscript received December 11, 2019; accepted May 3, 2020

国家自然科学基金(52069014, 62066030, 61663029, 51669014, 61663028), 江西省教育厅科技项目(GJJ201915, GJJ180940), 江西省研究生创新专项资金项目(YC2018-S422)资助

Supported by National Natural Science Foundation of China (52069014, 62066030, 61663029, 51669014, 61663028), Science and Technology Research Project of Education Department of Jiangxi Province (GJJ201915, GJJ180940), and Innovation Fund Designated for Graduate Students of Jiangxi Province (YC2018-S422)

本文责任编辑 王鼎

Recommended by Associate Editor WANG Ding

1. 南昌工程学院信息工程学院 南昌 330099 2. 江西省水信息协同感知与智能处理重点实验室 南昌 330099 3. 鄱阳湖流域水工程安全与资源高效利用国家地方联合工程实验室 南昌 330099

1. School of Information Engineering, Nanchang Institute of Technology, Nanchang 330099 2. Jiangxi Province Key Laboratory of Water Information Cooperative Sensing and Intelligent

为解决传统优化算法难以解决的复杂优化问题,研究者们受启发于生物和自然物理现象,提出了粒子群优化算法、差分进化算法、人工蜂群算法、和声搜索算法和进化策略等优化技术以解决各领域的优化任务.

粒子群优化算法(Particle swarm optimization, PSO)是由 Kennedy 等^[1]在 1995 年提出的群智能优化算法.该算法具备计算快速和易实现等优点,作为群体智能算法的一个重要分支,得到了众多学者广泛深入的研究.且已应用于预测^[2]、工程设

Processing, Nanchang 330099 3. National-local Joint Engineering Laboratory of Water Engineering Safety and Effective Utilization of Resources in Poyang Lake Area, Nanchang 330099

计^[3]、特征选择^[4]和经济^[5]等众多领域。

粒子群优化算法是源于对鸟群捕食行为的模仿研究, 人们从鸟群捕食过程当中得到启示, 并将其用于解决优化问题. 该算法通过群体中粒子间合作与竞争产生的群体智能指导优化搜索. 根据算法的特性, 全局最优粒子将信息传递给其他粒子是单向的信息流动, 可使粒子快速的收敛于最优解. 但当全局最优粒子处于局部最优位置时, 易导致所有粒子收敛到局部最优. 此外, 粒子在进化过程中, 从全局勘探转为局部开发阶段后, 若粒子处于局部最优位置, 将难以获得逃离局部最优位置的步长. 针对上述问题, 提出方波触发勘探与开发的粒子群优化算法 (Particle swarm optimization with square wave triggered exploration and exploitation, SWTPSO), 在提高全局最优质量的同时, 进一步增强粒子逃离局部最优的能力.

SWTPSO 中提出方波触发机制. 首先, 将标准 PSO 的更新公式进行改进, 去除速度项和加速因子的影响, 使新改进的公式局部搜索能力增强, 并存储当前粒子的位置, 作为下次使用标准 PSO 公式更新种群时粒子的新步长. 然后, 根据种群设定的总迭代次数, 将种群进化的过程划分为多个方波周期. 最后, 按照种群当前运行的迭代次数属于方波周期内的“高”或“低”值, 分别执行标准 PSO 公式和新改进公式更新种群, 使粒子的搜索状态随方波周期性变化. 将 SWTPSO 与多种其他优化算法进行比较, 其在数值优化问题上展现较好的优化性能.

本文结构如下: 第 1 节介绍标准粒子群算法原理及相关研究; 第 2 节阐述方波触发机制; 第 3 节对本文提出的策略进行理论分析和实验验证; 第 4 节将 SWTPSO 与新改进的 PSO 和其他智能算法进行仿真实验, 并分析算法的优化性能; 第 5 节为总结与展望.

1 粒子群算法及其简要综述

1.1 粒子群算法基本原理

设粒子数为 N 的种群, 在维度为 D 的空间中搜索, 粒子在解空间所处的位置可表示为 $x_i = (x_{i1}, x_{i2}, \dots, x_{iD})$, $i = 1, 2, \dots, N$ 代表第 i 个粒子的位置, 而每个 x_i 都可由目标函数求得函数值 $f(x_i)$. $v_i = (v_{i1}, v_{i2}, \dots, v_{iD})$, $i = 1, 2, \dots, N$ 代表第 i 个粒子的速度, 表示在每一次迭代中粒子移动的步长, $pbest_i = (pbest_{i1}, pbest_{i2}, \dots, pbest_{iD})$, $i = 1, 2, \dots, N$ 代表第 i 个粒子所经历的历史最优位置, $gbest = (gbest_1, gbest_2, \dots, gbest_D)$ 代表整个种群的历史最优位置. 速度和位置可按照以下公式更新:

$$v_{ij}^{t+1} = wv_{ij}^t + c_1r_1(pbest_{ij}^t - x_{ij}^t) + c_2r_2(gbest_j^t - x_{ij}^t) \quad (1)$$

$$x_{ij}^{t+1} = x_{ij}^t + v_{ij}^{t+1} \quad (2)$$

式中, 惯性权重 w 控制粒子速度 v 使粒子的局部和全局搜索能力相平衡, 当 w 较大时粒子的全局搜索能力较强, 反之粒子的局部搜索能力较强. 式 (1) 中 c_1 和 c_2 为学习因子 (也叫加速因子), 可使粒子具有自主学习和向其他优秀粒子学习的能力, 使其以较快速度向最优解靠近. r_1 和 r_2 为 $(0, 1)$ 内的随机数, $pbest_{ij}^t$ 为粒子 i 在第 t 次迭代时历史最优位置的第 j 维, $gbest_j^t$ 为第 t 次迭代时整个种群历史最优位置的第 j 维.

1.2 粒子群算法简要综述

粒子群优化算法具有高效和实现简单等特点, 已成功应用于许多现实优化任务^[6-7], 但存在多样性差、易早熟收敛等问题制约了粒子群优化算法的发展, 为克服上述问题, 众多学者做了大量研究, 所做工作包含以下 4 类:

1) 参数选择

惯性权重和学习因子等参数, 对算法的优化性能有重要影响. 文献 [8] 首次引入时变惯性权重, 改善粒子的收敛速度. 文献 [9] 根据种群的信息调整惯性权重. 文献 [10] 引入贝叶斯技术调整惯性权重. 文献 [11] 提出时变的加速因子策略. 文献 [12] 引入正余弦函数控制加速因子的变化. 文献 [13] 将惯性权重和加速因子作为三个维度信息处理. 文献 [14] 提出一种反馈系统, 该系统可使参数适应粒子所处的特定环境, 并使用惩罚函数来处理约束.

2) 粒子更新方程

为增强粒子逃离局部最优能力, 探索更多新区域, 文献 [15] 使用高斯分布更新粒子位置. 文献 [16] 引入反向学习生成粒子的反向解参与种群进化. 文献 [17] 结合随机学习和列维飞行策略, 在标准速度更新公式中增加向随机粒子学习项. 文献 [18] 在速度更新公式中增加子社会学习项, 使粒子向邻域最优粒子学习. 文献 [19] 中, 为消除对 $gbest$ 的过度依赖, 将其他粒子的维度信息作为新的信息源, 并提出了 3 个更新粒子的公式.

3) 拓扑结构

种群的拓扑结构与种群的多样性有关, 吸引了众多学者关注. 文献 [20] 分析了不同类型的静态邻域结构及其对 PSO 性能的影响, 并认为星型、环型和冯诺依曼拓扑结构的适应性最好. 文献 [21] 使用整个邻域的信息引导粒子搜索最优解. 文献 [22] 提出自适应时变拓扑连接模型, 根据种群搜索性能

改变粒子的拓扑连接,平衡算法的勘探与开发搜索.文献[23]采用环形拓扑结构增强种群的多样性和开发能力.文献[24]在迭代过程中保留局部最优解,并根据位置对维度进行排序,生成局部最优拓扑.

4) 算法混合

各优化技术在处理不同优化问题时展现不同的优势.文献[25]通过差分算法和混沌映射控制算法运行期间的参数.文献[26]将遗传算法的变异和交叉操作引入粒子群算法中.文献[27]将萤火虫算法的局部搜索能力与粒子群的全局搜索能力结合.文献[28]将多种经典的改进粒子群算法优势集合,通过学习上一代粒子经验,采用自适应方法为粒子分配最佳策略.文献[29]将标准 PSO 和社交学习粒子群优化相结合,以提高算法的探索和开发能力,并在 30、50 和 100 维上验证了算法的性能.

综上所述可知,通过在种群进化过程中调节参数,动态调节勘探与开发;通过改变粒子的更新方程,进一步改变粒子的运动方式或运动轨迹;通过改变粒子的拓扑结构,根据粒子间不同的信息分享机制调节勘探与开发;通过将其他智能算法的策略引入粒子群算法中,调节勘探与开发.

上述算法的种群在进化的全过程,始终保持初始设置的进化方式不变,或只对进化方式进行简单调整.例如,在进行参数选择的改进中,其使用的惯性权重线性变化策略,可等价于以整个进化过程为一个周期,通过惯性权重的变化,调节勘探与开发.因周期设置不合理,容易出现因勘探持续时间过长而耗费评估资源,或因陷入局部最优后难以再次进行全局勘探等问题.同理,在粒子更新方程、拓扑结构和算法混合等改进中,都未考虑多阶段调整种群的进化方式,进而调节勘探与开发等问题.

针对上述问题,本文提出方波触发勘探与开发的粒子群优化算法,其将种群进化过程划分为多个方波周期,周期性地增强种群勘探与开发能力,可避免因过度勘探造成评估资源浪费,又同时增强粒子逃离局部最优的能力,进一步提升算法的收敛速度和精度.

2 方波触发勘探与开发的粒子群优化算法

2.1 方波触发机制

方波 (Square wave, SW) 是一种非正弦曲线波形,通常在逻辑电路和信号处理时出现.理想方波只有“高”和“低”两个值.“高”值在一个波形周期内占有的时间比值称为占空比,占空比为 50% 的矩形波称为方波.若使用 1 和 0 分别表示方波振幅的高

和低值,则方波的前半个周期内振幅为 1,后半个周期振幅为 0.方波表达式见下:

$$f_{sw}(t) = \begin{cases} 1, & (n-1)T < t \leq (n-1)T + \frac{T}{2} \\ 0, & (n-1)T + \frac{T}{2} < t \leq nT \end{cases} \quad (3)$$

式中, t 为种群当前迭代次数, T 为方波函数的周期, n 为整个迭代过程可划定的周期个数, $n = 1, 2, \dots, Max_iteration/T$, $Max_iteration$ 为最大迭代次数.

标准粒子群算法在优化过程中,整个粒子群体根据全体粒子和自身的搜索经验向着最优解的方向“飞行”,在较大动量项系数(加速因子等)的作用下,粒子可能错过最优解,使算法收敛到局部最优.

为消除动量项的影响,在新速度更新式(4)中,原等式(1)右边的速度项替换成粒子的当前位置,加速因子设置为 1, r_4 和 r_5 为 (0,1) 间的随机数.联合式(4)~(5)可知,其可使粒子获得比前半周期更小的步长,粒子直接向个体最优 (pbest) 和全局最优 (gbest) 学习,探测两者外围解空间区域,相对增强种群局部搜索能力.

$$V_j^{t+1} = x_j^t + r_4(pbest_j^t - x_j^t) + r_5(gbest_j^t - x_j^t) \quad (4)$$

$$x_j^{t+1} = V_j^{t+1} \quad (5)$$

在标准粒子群算法进化过程中,搜索状态由全局勘探转为局部开发后,若种群陷入局部最优,则粒子多样性丧失较快,难以逃离局部最优.此外,种群执行局部开发搜索后,难以再次开启全局勘探搜索.

针对上述问题,SWTPSO 利用方波的周期特性,在种群的进化过程中,使粒子周期性增强全局勘探与局部开发搜索.具体实现过程如下:

$$\begin{cases} \text{运行式(1)~(2)}, & (n-1)T < t \leq \\ & (n-1)T + \frac{T}{2} \\ \text{运行式(4)~(5)}, & (n-1)T + \frac{T}{2} < t \leq nT \end{cases} \quad (6)$$

由式(6)可知,在前半个方波周期内,使用标准 PSO 公式增强全局勘探能力,提高寻得最优解的概率;在后半个方波周期内,使用去除动量项的改进公式更新种群,增强其局部搜索能力,对所探得区域进行精细搜索.在整个方波周期内,种群在前半周期相对于后半周期,增强全局勘探能力,后半周期相对前半周期增强局部开发能力.当种群粒子执行完后半个周期的局部搜索后,在下一个周期将自动获得与粒子当前位置有关的新速度,使粒子获得的运动步长多变,种群多样性增强.

2.2 算法实现步骤

本文采用方波触发机制, 可有效平衡算法全局勘探与局部开发的能力, 算法实现步骤如下:

算法 1. SWTPSO 程序

- 1) 初始化. $N, D, c_1, c_2, r_1, r_2, r_3, r_4, Max_iteration, n, w, iteration$.
- 2) 在问题空间的初始化范围内, 用随机位置初始化粒子.
- 3) 评估适应度函数值, 并将 X 指定为 $pbest$. 在种群中找到 $gbest$.
- 4) $iteration = 1$;
- 5) while (终止条件不满足) do;
- 6) $w = 0.55 \times (-0.5 \times iteration / Max_iteration)$
- 7) if $\{iteration < (n - 1)T + T/2\}$ then;
- 8) 分别通过式 (1) ~ (2) 改变粒子的速度和位置;
- 9) else if $\{(n - 1)T + T/2 \leq iteration \leq nT\}$ then;
- 10) 分别通过式 (4) ~ (5) 改变粒子的速度和位置;
- 11) End if;
- 12) End if;
- 13) 当粒子位置值超出边界值时, 则其被赋为搜索空间的边界;
- 14) 评估新粒子 X_i 的适应度函数;
- 15) if $f(X_i) < f(pbest_i)$ then;
- 16) 将 $f(X_i)$ 赋值给 $f(pbest_i)$, $pbest_i$ 为在 N 维空间中当前位置 X_i 的值;
- 17) End if;
- 18) 更新 $gbest$ 的值;
- 19) $iteration = iteration + 1$;
- 20) End while;
- 21) 输出全局最优值 ($gbest$).

3 算法策略分析

本节主要对算法采用的策略进行理论分析或实验验证, 通过实验测定方波触发机制中的最佳方波周期 T , 并验证方波触发机制可有效提高种群多样性.

方波触发机制从整个种群搜索状态入手, 在式 (1) 中去除速度项, 使算法拥有更强的局部搜索能力, 而保留速度项的标准 PSO 更新公式, 则可探索更多新搜索空间^[8]. 方波触发机制将种群进化的全过程划分为若干个方波周期, 在一个周期内的前半周期采用式 (1) ~ (2) 更新种群, 进行大范围的全局搜索, 在后半个周期内采用式 (4) ~ (5) 更新种群, 进行小范围的局部搜索, 以此平衡算法的勘探与开发能力. 因此, 为提高算法的优化性能, 划分合适的方波周期是本文的关键.

3.1 方波周期分析

为便于实验分析设定最佳方波周期 T , 改为求 $f = 1/T = n/Max_iteration$ 的值, n 为周期的个数, $Max_iteration$ 为最大迭代次数. 在算法运行时, 周期 T 为初始化设定的常数, 当最大迭代次数 $Max_iteration$ 变化, 则只有周期执行的个数 n 变化. 本文通过取不同的 f 值, 测试算法在优化不同函数时展现的综合性能, 选取最佳方波周期 T . 因此, 本文设定的最佳方波周期 T 只与算法的优化性能有关, 与本节设定的最大迭代次数无关.

实验选取多模函数 Penalized 2、Levy 作为测试函数, 粒子数 N 为 10, 维度 D 为 30, 最大迭代次数为 20 000, 各函数运行 30 次, 当所得全局最优适应值达到 10^{-18} , 则认为算法求得函数最优解, 记录下所需迭代次数. 当算法达到终止条件, 全局最优适应值未满足 10^{-18} , 则判定此次运行搜索失败, 用“×”标示, 且不计入平均值的计算. f 分为 0.01 ~ 0.002 共 9 个值, 实验结果见表 1 ~ 2.

在 Penalized 2 函数上, 当 f 为 0.01 时, 算法有 28 次求得最优解, 其平均迭代次数为 4 798. 当 f 为 0.009、0.008、0.007、0.006、0.005、0.004 时, 算法在 30 次都求得最优解. 其中, 当 f 为 0.009 时, 平均迭代次数为 5 148; 当 f 为 0.008 时, 平均迭代次数为 6 028; 当 f 为 0.007 时, 平均迭代次数为 5 277; 当 f 为 0.006 时, 平均迭代次数为 5 021; 当 f 为 0.005 时, 平均迭代次数为 4 488; 当 f 为 0.004 时, 平均迭代次数为 4 510. 当 f 为 0.003 时, 算法有 29 次求得最优解, 平均迭代次数 4 907. 当 f 为 0.002 时, 算法有 25 次求得最优解, 平均迭代次数为 3 219.

在 Levy 函数上, 当 f 为 0.01 时, 算法有 22 次求得最优解, 平均迭代次数 5 887. 当 f 为 0.009、0.008、0.007 和 0.005 时, 算法在 29 次都求得最优解, 其中 0.009 时平均迭代次数为 5 954; 0.008 时平均迭代次数为 5 259; 0.007 时平均迭代次数为 4 923; 0.005 时平均迭代次数为 4 321. 当 f 为 0.006 时, 算法有 27 次求得最优解, 平均迭代次数 4 027. 当 f 为 0.004 时, 算法有 30 次求得最优解, 平均迭代次数 4 494. 当 f 为 0.003 时, 算法有 28 次求得最优解, 平均迭代次数 4 252. 当 f 为 0.002 时, 算法有 27 次求得最优解, 平均迭代次数 5 146.

为更直观反映 f 值对算法优化性能的影响, 以不同 f 值为横坐标, 算法在运行 30 次中未寻得最优解的次数为纵坐标, 绘制折线图 1. 在 Penalized 2 函数上, f 为 0.004、0.005、0.006、0.007、0.008、0.009 时, 算法都能求得最优解. 在 Levy 函数上, f 为 0.004 时, 算法能求得最优解.

综合分析实验结果可知, 不同的周期 T 对算法

表 1 Penalized 2 函数上使用不同的 f 值寻找全局最优值的迭代次数Table 1 Numbers of iterations for finding the global optimum using different values of f on the Penalized 2 function

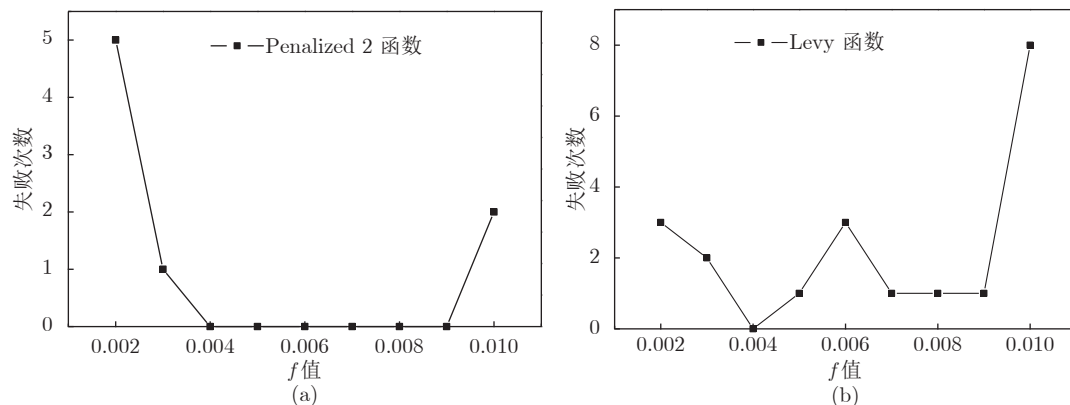
运行次数	0.01	0.009	0.008	0.007	0.006	0.005	0.004	0.003	0.002
1	11704	12273	11981	11625	10146	9985	11252	11198	×
2	2	2	2	2	1780	2	2	2	3056
3	13065	11979	12499	11119	2	9847	10795	9500	2
4	1405	2	3	2	10421	801	2	2	8887
5	2	13336	2	11896	1109	2	11033	×	2
6	14060	2	10895	1364	2	10132	3	9472	×
7	2758	14999	1106	1107	12761	2	1333	2	1222
8	865	2	2	2	2476	2	2	9418	426
9	2	12276	13873	11136	2	2	10078	2	2
10	14962	1407	2	2	10834	2	2	10544	×
11	1436	2109	13018	12386	1034	12742	11571	2	9060
12	2	2	2	2	2	2	2	10181	488
13	2	14577	13927	10019	2	11213	10855	1071	2
14	13346	2	2	1457	13428	2	1397	401	8615
15	1633	12661	11396	2	2	10691	2	2	2
16	2	2	2	12466	13939	2	9906	9438	8822
17	14325	13727	15113	2	2	10756	413	424	1319
18	2	1470	2	12205	12848	930	2	2	2
19	13871	2018	12345	2	2	2	11565	9610	×
20	3	505	2	11701	10586	11815	829	2	×
21	3	3	14717	2	2	978	2	11406	5039
22	2	2	2	2	12163	2	10889	601	529
23	×	14773	13978	11976	3	10870	2	2	517
24	14313	270	2	1368	2	2	10413	11753	639
25	927	2	10368	2	11707	11587	857	2	2
26	830	12428	2	12196	2	2	2	14083	9400
27	2	2	13887	2	11969	10566	12339	2	2
28	×	2	953	11848	2	2	2	9706	8173
29	14813	12621	2	2	12454	11169	9447	2	2
30	2	975	10766	12405	960	533	297	13482	14269
平均	4798	5148	6028	5277	5021	4488	4510	4907	3219

表 2 Levy 函数上使用不同的 f 值寻找全局最优值的迭代次数Table 2 Numbers of iterations for finding the global optimum using different values of f on the Levy function

运行次数	0.01	0.009	0.008	0.007	0.006	0.005	0.004	0.003	0.002
1	13874	14277	13198	10780	10421	13557	11056	9483	13787
2	2826	1183	2	1449	2	2	2	2	2
3	2	2	13689	2	11314	10167	13358	2	12695
4	×	13732	2042	11042	2	2	2	10550	2
5	×	2	2686	2	11160	11234	13494	1239	×
6	14837	13975	2	9906	2	2	1578	552	3200
7	5154	2	13005	2	11868	9903	2	2	2
8	2	14063	1244	10620	2	2	10440	9624	13760

表 2 Levy 函数上使用不同的 f 值寻找全局最优值的迭代次数 (续表)Table 2 Numbers of iterations for finding the global optimum using different values of f on the Levy function (continued table)

运行次数	0.01	0.009	0.008	0.007	0.006	0.005	0.004	0.003	0.002
9	12041	2	2	1362	×	10850	1376	2	2
10	2	×	12501	610	4469	603	2	2	12647
11	×	12550	2	2	2	2	10836	9370	620
12	12924	2	11695	11196	×	9632	2	2	2
13	2	13450	431	1730	4908	1492	9457	13545	13183
14	2	2	3306	2	783	2	2	2	805
15	×	12873	1861	10321	1413	9453	10773	9836	625
16	13729	3	2	2	2	2	701	2	2
17	2	2	×	12235	12042	10359	2	×	×
18	13234	14143	12546	2	2	2	8828	12012	10905
19	2780	2	2	2	10954	11268	621	375	2
20	2	14885	12626	12806	2	776	1180	2	8737
21	13953	2	2	2	2	2	2	×	2
22	4594	13092	13093	14701	11422	×	10400	9022	12969
23	2	4313	2	2	2	1381	2	2	446
24	13667	2	11992	11462	×	1442	9952	11454	2
25	2	14440	2	2	2434	2	2	3	×
26	×	754	13095	2	1446	10102	10023	2	13288
27	×	992	956	×	2	959	2	2	2
28	×	1009	2	11072	14639	2	10721	11631	8908
29	×	2	2	2	636	12115	2	2	2
30	14436	12920	12514	11453	2	2	2	10330	12355
平均	5887	5954	5259	4923	4072	4321	4494	4252	5146

图 1 不同 f 值的失败次数Fig.1 Number of failures for different f

优化性能有重要影响, 当 T 设置过大, 种群长时间使用标准 PSO 更新公式搜索解空间, 其具有的较大动量项系数将使种群错过最优解, 且周期过大将占用大量的迭代次数, 使执行的周期次数变少, 种群粒子自动获得较大步长的次数减少, 削弱了种群逃离局部最优的能力; 当 T 设置过小, 种群粒子或未探得最优解所在区域, 就进行局部搜索, 使种群在局部最优区域耗费评估次数。

对于 f 的取值, 通过增加两组不同类型 (单模、多模、旋转、偏移、组合等) 测试函数进行实验, 验证了 f 的最佳取值范围为 $[0.004, 0.009]$. 当 $f = 0.004$ 时, 所得秩均值最小, 算法拥有更好优化性能。

3.2 粒子步长和种群多样性分析

分析标准 PSO 算法中粒子的运动方式, 将式 (1) ~ (2) 经整理变化, 可得:

$$x_{ij}^{t+1} = x_{ij}^t + wv_{ij}^t + c_1r_1(pb_{ij}^t - x_{ij}^t) + c_2r_2(gb_{ij}^t - x_{ij}^t) \quad (7)$$

粒子的实际运动步长为:

$$step_{PSO}^t = wv_{ij}^t + c_1r_1(pb_{ij}^t - x_{ij}^t) + c_2r_2(gb_{ij}^t - x_{ij}^t) \quad (8)$$

本文提出的方波触发机制, 在一个方波周期内的前半方波周期采用式 (1) ~ (2) 更新种群, 搜索到一定区域后, 使用式 (4) ~ (5) 在后半方波周期对该区域进行精细搜索. 再到执行下一个方波周期时, 式 (6) 中所得粒子的速度 V , 将自动替换成为式 (1) 等号右边第 1 项 V , 此时再次使用式 (1) ~ (2) 更新, 粒子的实际运动步长为:

$$step_{SW}^t = wx_{ij}^t + (c_1r_1 + wr_4)(pb_{ij}^t - x_{ij}^t) + (c_2r_2 + wr_5)(gb_{ij}^t - x_{ij}^t) \quad (9)$$

对比式 (8) 和式 (9) 可知, 在不越界的情况下, $step_{PSO}^t$ 随迭代次数的增加逐渐减少, $step_{SW}^t$ 的大小则与粒子的当前位置有关, 其提供给粒子的步长多变, 能大幅提升种群多样性.

为验证标准 PSO 公式下 ($step_{ps}$), 相比方波触发机制下 ($step_{sw}$) 的优势, 使用 Rastrigin 函数, 测试粒子在进化过程中步长的变化. 实验中, 种群个数为 $N = 10$, 维度 $D = 10$, 迭代次数为 500 次. 随机选取种群中一个粒子, 记录粒子 10 个维度中步长的变化情况. 实验结果见图 2.

第 1 ~ 250 迭代次数为前半周期, 第 251 ~ 500 迭代次数为后半周期. 由图 2 中可以看出, 在粒子的 10 个维度中, $step_{sw}$ 步长的变化次数在进化全过程多于 $step_{ps}$, 验证在 $step_{sw}$ 策略下, 通过提供多变的步长, 可达到按周期触发勘探与开发的目的.

为验证引入方波触发机制可有效提高种群多样性, 使用多样性度量式 (10)^[30], 对比标准 PSO 和加入方波触发机制的 PSO, 其种群进化过程中多样性的变化.

$$Diversity = \frac{1}{N \cdot 2 \cdot a} \sum_{i=1}^N \sqrt{\frac{1}{D} \sum_{j=1}^D (x_{i,j} - \bar{x}_j)^2} \quad (10)$$

式中, N 表示种群的个数, $2 \cdot a$ 为搜索空间的范围, D 表示维度, \bar{x} 表示种群的中心位置. 方波触发机制记为 SW, 选取 Rastrigin 旋转偏移函数和 Shifted 旋转偏移函数进行测试, 粒子数 N 为 10, 维度 D 为 30, 评估次数为 10000 次, 所得两者多样性实验结果见图 3. 引入方波触发机制的 PSO (PSO + SW) 比标准 PSO, 在种群进化的全过程中能保持更高的多样性, 种群勘探与开发能力可保持更好平衡. 同时对标准 PSO 和 PSO + SW 进行收敛性分

析, 见图 4. 实验结果表明, PSO + SW 比标准 PSO 可收敛到更高精度解.

4 仿真实验

本节结构如下: 第 4.1 节设置 SWTPSO 涉及的参数, 给出测试函数的基本信息; 第 4.2 节选取新近改进的 PSO 与 SWTPSO 在维度为 30、50、100 的条件下比较; 第 4.3 节对算法收敛性能进行分析. 第 4.4 节将 SWTPSO 与新改进的蜂群和差分算法在 26 个函数上进行 50 维比较, 再与改进和声搜索算法、粒子群算法、人工蜂群算法、差分算法进行 100 维比较, 最后在 CEC2015 测试集上进一步验证 SWTPSO 的优化性能.

4.1 测试函数和参数设置

引入 26 个基准函数进行仿真实验, 表 3 给出了 26 个测试函数的基本信息. SWTPSO 的参数设置如下: 种群规模 N 设置为 10, c_1 和 c_2 设置为 2.0, 惯性权重 $w = 0.55 \times \exp(-0.5 \times iterNum / Max_iterNum)$, 用 $Max_iterNum$ 表示最大迭代次数, $iterNum$ 表示当前迭代次数, 方波触发机制中, f 值设为 0.004.

4.2 仿真实验及结果分析

将 SWTPSO 与 7 种 PSO 改进算法在维度 $D = 30, 50, 100$ 的条件下对比, 7 种 PSO 改进算法分别为 Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients (HPSO-TVAC)^[11]、Comprehensive learning particle swarm optimizer for global optimization of multi-modal functions (CLPSO)^[31]、A novel particle swarm optimization algorithm with levy flight (LFPSO)^[32]、Dynamic multi-swarm particle swarm optimizer (DMS-PSO)^[33]、An enhanced particle swarm optimization with levy flight for global optimization (PSOLF)^[34]、A particle swarm optimization algorithm with random learning mechanism and levy flight for optimization of atomic clusters (RPSOLF)^[17]、A hybrid particle swarm optimizer with sine cosine acceleration coefficients (H-PSO-SCAC)^[12]. 各算法的参数都按照原文献设置, 当 $D = 30$ 时, 评估次数设为 20 万次; $D = 50$ 时, 评估次数设为 25 万次; $D = 100$ 时, 评估次数设为 50 万次. 所有算法独立运行 50 次, 最终结果取 50 次的平均值, 具体实验数据见表 4 ~ 6. 表中, MBF 表示平均最优适应值, SD 表示标准差, 粗体表示最好值.

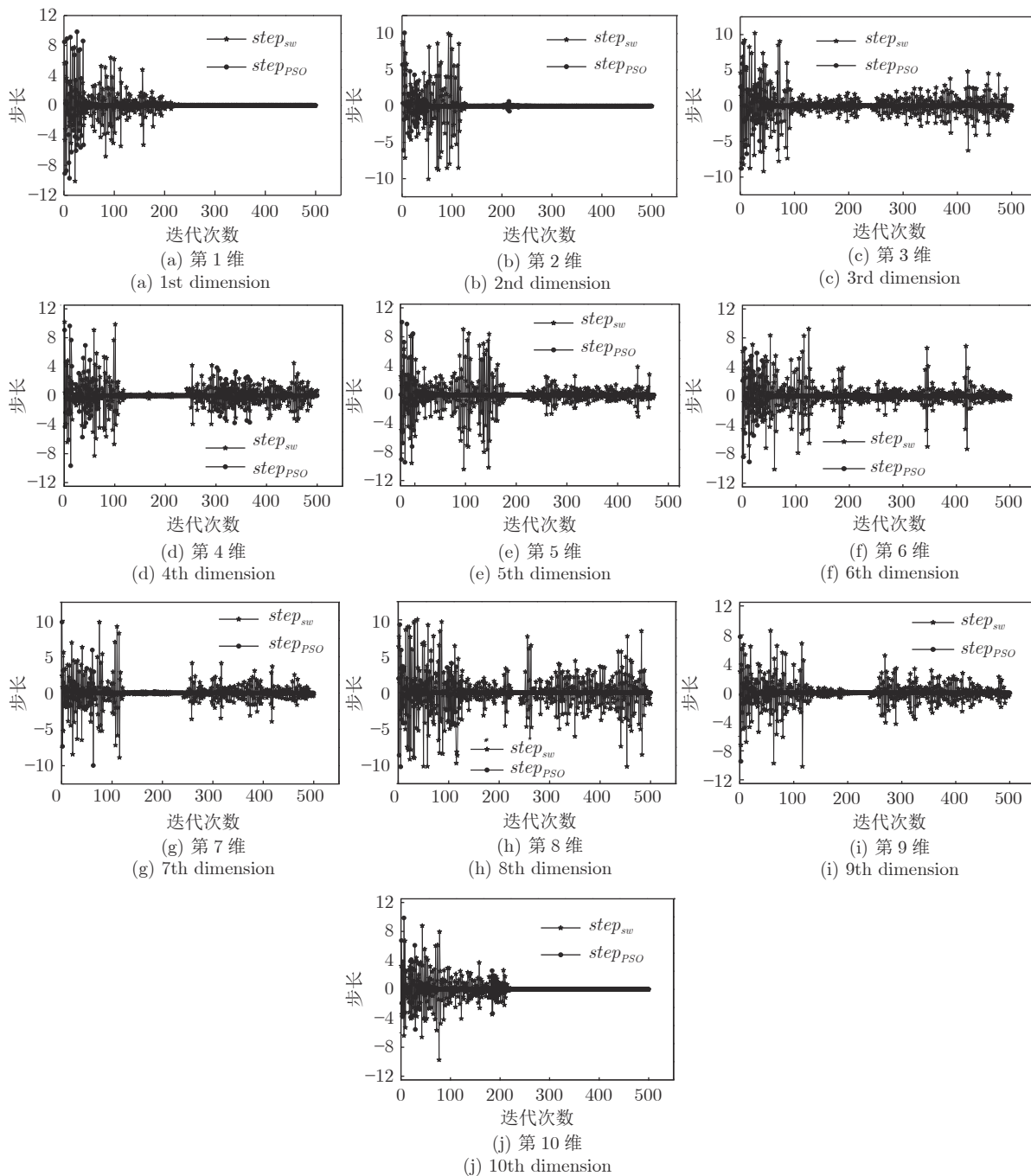


图 2 粒子在 10 维中的步长变化

Fig.2 Step size changes of 10 dimensions of particles

由表 4~6 可以看出,在相同评估次数下,SWTPSO 在 $D = 30, 50, 100$ 维时,所测函数 $f_1、f_2、f_6、f_8、f_{12}、f_{13}、f_{14}$ 全部求得最优解,所测函数 $f_9、f_{10}$ 同样求得比其他算法更高质量的解,测试 $f_3、f_4、f_5$ 函数所得最优解相比其他算法优势明显. CLPSO 在测试函数 f_5 时优势明显,其余函数在 30 维时能取得较好质量的解,但在 50、100 维时算法性能有

待加强. CLPSO 主要借助不同的个体最优 ($pbest$) 优势信息产生最优解,但因其搜索过程中种群常易更新停滞,早熟收敛仍是 CLPSO 需解决的问题. HPSO-TVAC 和 DMS-PSO 在测试函数 $f_9、f_{10}$ 时优势明显,测试其余函数时,上述算法在低维时能求得较好质量的解,在高维搜索时算法易陷入局部最优. LPSO、RPSOLF 都采用列维飞行策略,使

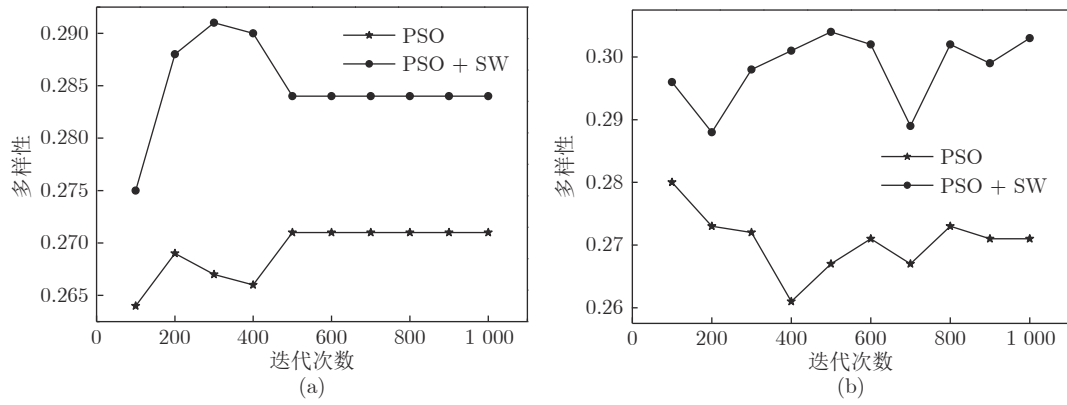


图 3 PSO 和 PSO+SW 的种群多样性变化

Fig.3 The population's diversity of PSO and PSO + SW

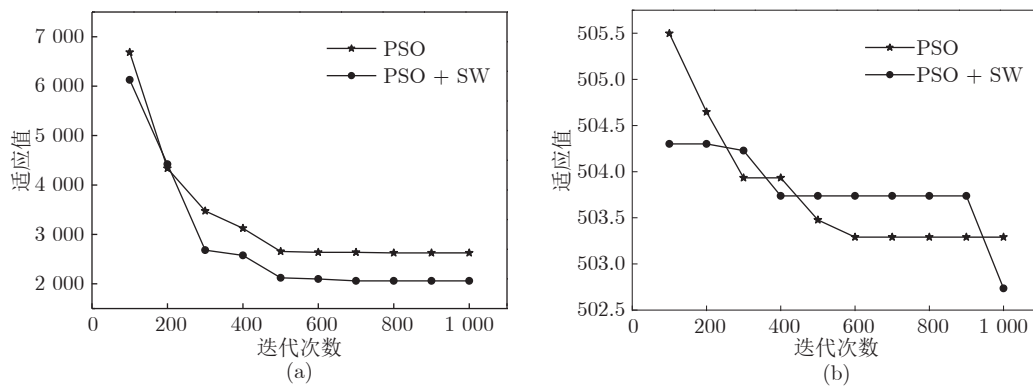


图 4 PSO 和 PSO + SW 的收敛性

Fig.4 The convergence performance of PSO and PSO + SW

表 3 26 个基准测试函数

Table 3 Information of 26 benchmark functions

函数名	函数式	范围	最优解
Sphere	$f_1(x) = \sum_{i=1}^D x_i^2$	[-100, 100]	0
Schwefel2.22	$f_2(x) = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i $	[-10, 10]	0
Rosebrock	$f_3(x) = \sum_{i=1}^D [100(x_{i+1} - x_i^2)^2 + (1 - x_i^2)^2]$	[-5, 10]	0
Quartic	$f_4(x) = \sum_{i=1}^D ix_i^4 + random [0, 1)$	[-1.28, 1.28]	0
Schwefel2.26	$f_5(x) = 418.98288727243380 \cdot D - \sum_{i=1}^D x_i \sin \sqrt{ x_i }$ $\bar{f}_5(x) = \sum_{i=1}^D -x_i \sin \sqrt{ x_i }, (D = 100)$	[-500, 500]	$0 \sim 418.9826 \times D$
Rastrigin	$f_6(x) = \sum_{i=1}^D [x_i^2 - 10 \cos 2\pi x_i + 10]$	[-5.12, 5.12]	0
Ackley	$f_7(x) = -20 \exp \left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2} \right) - \exp \left(\frac{1}{D} \sum_{i=1}^D \cos 2\pi x_i \right) + 20 + e$	[-50, 50]	0

表 3 26 个基准测试函数 (续表)
Table 3 Information of 26 benchmark functions (continued table)

函数名	函数式	范围	最优解
Griewank	$f_8(x) = \frac{1}{4000} \sum_{i=1}^D (x_i)^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	[-600, 600]	0
Penalized 1	$f_9(x) = \frac{\pi}{D} \left\{ \sum_{i=1}^{D-1} (y_i - 1)^2 [1 + \sin(\pi y_{i+1})] + (y_D - 1)^2 + 10 \sin^2(\pi y_1) \right\} + \sum_{i=1}^D u(x_i, 10, 100, 4), y_i = 1 + \frac{x_i + 1}{4} u(x_i, a, k, m) = \begin{cases} u(x_i, a, k, m), & x_i > a \\ 0, & -a \leq x_i \leq a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$	[-100, 100]	0
Penalized 2	$f_{10}(x) = 0.1 \left\{ \sin^2(\pi x_1) + \sum_{i=1}^{D-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + (x_D - 1)^2 [1 + \sin^2(2\pi x_{i+1})] \right\} + \sum_{i=1}^D u(x_i, 5, 100, 4)$	[-100, 100]	0
Rotated Schwefel2.26	$f_{11}(x) = 4.18.928 \cdot D - \sum_{i=1}^D z_i, \text{ where } z_i = \begin{cases} -y_i \sin(\sqrt{ y_i }) i y_i \leq 500 \\ 0, & \text{否则} \end{cases}$ $y_i = y_i + 420.96, y_i = M \cdot (x - 420.96), M \text{ is an orthogonal matrix}$	[-500, 500]	0
Rotated Rastrigin	$f_{12}(x) = \sum_{i=1}^D [y_i^2 - 10 \cos(2\pi y_i) + 10], \text{ where } y = M \cdot x, M \text{ is an orthogonal matrix}$	[-5.12, 5.12]	0
Rotated Ackly	$f_{13}(x) = -20 \exp \left[-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D y_i^2} \right] - \exp \left[\frac{1}{D} \sum_{i=1}^D \cos(2\pi y_i) \right] + (20 + e)$ $\text{where } y = M \cdot x, M \text{ is an orthogonal matrix}$	[-32, 32]	0
Rotated Griewank	$f_{14}(x) = \frac{1}{4000} \sum_{i=1}^D y_i^2 - \prod_{i=1}^D \cos\left(\frac{y_i}{\sqrt{i}}\right) + 1, \text{ where } y = M \cdot x, M \text{ is an orthogonal matrix}$	[-600, 600]	0
Elliptic	$f_{15}(x) = \sum_{i=1}^D (10^6)^{\frac{i-1}{D-1}} x_i^2$	[-100, 100]	0
SumSquare	$f_{16}(x) = \sum_{i=1}^D i x_i^2$	[-10, 10]	0
SumPower	$f_{17}(x) = \sum_{i=1}^D x_i ^{(i+1)}$	[-1, 1]	0
Schwefel2.21	$f_{18}(x) = \max\{ x_i , 1 \leq i \leq D\}$	[-100, 100]	0
Step	$f_{19}(x) = \sum_{i=1}^D [x_i + 0.5]$	[-100, 100]	0
Exponential	$f_{20}(x) = \exp\left(0.5 \sum_{i=1}^D x_i\right)$	[-10, 10]	0
NCRastrigin	$f_{21}(x) = \sum_{i=1}^D [y_i^2 - 10 \cos(2\pi y_i) + 10]$	[-5.12, 5.12]	0
Alpine	$f_{22}(x) = \sum_{i=1}^{D-1} x_i \sin(x_i) + 0.1 x_i $	[-10, 10]	0
Levy	$f_{23}(x) = \sum_{i=1}^{D-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + \sin^2(3\pi x_1) + x_D - 1 [1 + \sin^2(3\pi x_D)]$	[-10, 10]	0
Weierstrass	$f_{24}(x) = \sum_{i=1}^D \left(\sum_{k=0}^{k_{max}} [a^k \cos(2\pi b^k (x_i + 0.5))] \right) - D \sum_{k=0}^{k_{max}} [a^k \cos(2\pi b^k 0.5)]$ $a = 0.5; b = 3; k_{max} = 20$	[-1, 1]	0

表 3 26 个基准测试函数 (续表)

Table 3 Information of 26 benchmark functions (continued table)

函数名	函数式	范围	最优解
Himmelballa	$f_{25}(x) = \frac{1}{D} \sum_1^D (x_i^4 - 16x_i^2 + 5x_i)$	$[-5, 5]$	-78.332
Michalewicz	$f_{26}(x) = - \sum_{i=1}^D \sin(x_i) \sin^{20}\left(\frac{ix_i^2}{\pi}\right)$	$[0, \pi]$	-50, $D = 50$

表 4 30 维实验结果对比

Table 4 Comparison of 30-dimensional test results

函数	指标	CLPSO	HPSO-TVAC	DMS-PSO	LFPSO	PSOLF	RPSOLF	H-PSO-SCAC	SWTPSO
f_1	MBF	8.06×10^{-96}	2.83×10^{-33}	1.53×10^{-113}	4.69×10^{-31}	0	0	0	0
	SD	3.53×10^{-96}	3.19×10^{-33}	5.14×10^{-113}	2.50×10^{-30}	0	0	0	0
f_2	MBF	5.50×10^{-57}	9.03×10^{-20}	2.18×10^2	2.64×10^{-17}	0	0	4.09×10^{-217}	0
	SD	1.92×10^{-56}	9.58×10^{-20}	1.83×10^2	6.92×10^{-17}	0	0	0	0
f_3	MBF	4.18×10^1	2.39×10^1	3.49×10^1	2.38×10^1	2.68×10^1	1.01×10^1	2.36×10^1	3.99
	SD	3.35×10^1	2.65×10^1	2.76×10^1	3.17×10^{-1}	1.03	9.69×10^{-1}	1.51×10^{-1}	2.22×10^1
f_4	MBF	1.74×10^{-3}	9.82×10^{-2}	6.09×10^{-4}	2.41×10^{-3}	2.60×10^{-5}	6.50×10^{-3}	3.94×10^{-237}	8.89×10^{-4}
	SD	7.83×10^{-4}	3.26×10^{-2}	4.18×10^{-4}	8.07×10^{-4}	2.10×10^{-5}	5.50×10^{-3}	0	1.58×10^{-3}
f_5	MBF	3.82×10^{-4}	1.59×10^3	3.21×10^3	1.37×10^3	2.00×10^3	2.85×10^3	6.64×10^3	3.51×10^2
	SD	1.28×10^{-7}	3.26×10^2	6.51×10^2	6.36×10^2	6.08×10^2	3.79×10^2	2.39×10^3	1.14×10^3
f_6	MBF	1.27×10^1	9.43	1.49×10^1	4.54	0	0	0	0
	SD	4.22	3.48	3.62	1.03×10^1	0	0	0	0
f_7	MBF	1.42×10^{-14}	7.29×10^{-14}	6.06×10^{-12}	1.68×10^{-14}	8.88×10^{-16}	8.88×10^{-16}	8.88×10^{-16}	5.88×10^{-16}
	SD	7.46×10^{-15}	3.00×10^{-14}	3.90×10^{-13}	4.84×10^{-15}	0	0	0	0
f_8	MBF	3.20×10^{-3}	9.75×10^{-3}	1.85×10^{-3}	8.14×10^{-17}	0	0	0	0
	SD	4.93×10^{-3}	8.33×10^{-3}	4.07×10^{-3}	4.46×10^{-16}	0	0	0	0
f_9	MBF	1.36×10^{-33}	2.71×10^{-29}	0	4.67×10^{-31}	1.66×10^{-2}	1.98×10^{-32}	2.10×10^{-1}	1.58×10^{-32}
	SD	2.82×10^{-33}	1.88×10^{-29}	0	9.01×10^{-31}	1.27×10^{-2}	9.23×10^{-33}	9.67×10^{-1}	5.07×10^{-33}
f_{10}	MBF	1.65×10^{-33}	2.79×10^{-28}	6.16×10^{-35}	1.51×10^{-28}	9.01×10^{-9}	1.65×10^{-32}	1.09	1.36×10^{-32}
	SD	4.03×10^{-33}	2.18×10^{-28}	2.76×10^{-34}	8.00×10^{-28}	1.11×10^{-8}	4.36×10^{-33}	1.42	4.84×10^{-33}
f_{11}	MBF	4.39×10^3	5.32×10^3	4.04×10^3	5.51×10^3	1.54×10^3	9.98×10^3	4.48×10^3	6.63×10^3
	SD	3.51×10^2	7.00×10^2	5.68×10^2	5.64×10^2	4.28×10^2	7.58×10^{-12}	1.73×10^2	1.43×10^3
f_{12}	MBF	8.17×10^1	5.29×10^3	4.20×10^1	1.79	0	0	0	0
	SD	1.08×10^1	1.25×10^1	9.74	9.81	0	0	0	0
f_{13}	MBF	5.91×10^{-5}	9.29	2.42×10^{-14}	1.65×10^{-14}	0	0	0	0
	SD	6.46×10^{-5}	2.07	1.52×10^{-14}	5.40×10^{-15}	0	0	0	0
f_{14}	MBF	7.69×10^{-5}	9.26×10^{-3}	1.02×10^{-2}	1.48×10^{-3}	0	0	0	0
	SD	7.66×10^{-5}	8.80×10^{-3}	1.24×10^{-2}	6.17×10^{-3}	0	0	0	0

种群增加了随机步长, 其中 LFPSO 使种群粒子绕着最优解做列维飞行, 在算法后期种群易更新停滞, 故 LFPSO 在低维时, 测试单模和多模函数时, 搜索到最优解的质量较高, 但在高维仍易陷入局部最优。LFPSO 和 RPSOLF 则另增加了随机选择策略, 在一定概率范围内使用列维飞行策略, 在测试函数 f_1 、 f_2 、 f_6 、 f_8 、 f_{12} 、 f_{13} 、 f_{14} 求得了函数最优解, 但在高维条件下测试 f_9 、 f_{10} 时算法仍陷入局部最优。H-PSO-

SCAC 采用 \sin 、 \cos 函数控制加速因子的变化, 并在反向学习初始化种群后, 采用改进的速度更新公式, 其在优化函数 f_4 时, 能够取得较高质量的解, 在函数 f_1 、 f_2 、 f_6 、 f_8 、 f_{12} 、 f_{13} 、 f_{14} 同样能求得函数最优解, 但在测试函数 f_9 和 f_{10} 时, 算法仍陷入局部最优。

为进一步综合评价 SWTPSO 的性能, 引入 Friedman 检验分析所得测试数据, 秩均值作为 Fri-

表 5 50 维实验结果对比
Table 5 Comparison of 50-dimensional test results

函数	指标	CLPSO	HPSO-TVAC	DMS-PSO	LFPSO	PSOLF	RPSOLF	H-PSO-SCAC	SWTPSO
f_1	MBF	2.28×10^{-85}	5.50×10^{-11}	8.62×10^{-103}	9.17×10^{-17}	0	6.57×10^{-201}	0	0
	SD	3.87×10^{-85}	1.42×10^{-10}	2.82×10^{-102}	3.20×10^{-16}	0	0	0	0
f_2	MBF	4.47×10^{-52}	1.12×10^{-7}	3.09×10^2	8.00×10^{-1}	0	8.01×10^{-124}	4.69×10^{-245}	0
	SD	1.22×10^{-51}	1.76×10^{-7}	2.87×10^1	4.44	0	2.36×10^{-122}	0	0
f_3	MBF	6.97×10^1	1.36×10^2	7.03×10^1	4.41×10^1	4.74×10^1	4.41×10^1	4.71×10^1	7.24
	SD	4.45×10^1	5.90×10^2	4.49×10^1	2.82×10^{-1}	9.84×10^{-1}	4.74×10^1	6.75	5.33×10^1
f_4	MBF	3.31×10^{-3}	3.78×10^{-23}	1.15×10^{-3}	2.37×10^{-3}	1.60×10^{-5}	3.26×10^{-3}	8.85×10^{-176}	9.08×10^{-4}
	SD	9.05×10^{-4}	1.39×10^{-22}	3.92×10^{-4}	1.01×10^{-3}	1.17×10^{-5}	5.36×10^{-3}	0	2.68×10^{-3}
f_5	MBF	9.53	9.83×10^2	9.76×10^3	4.36×10^3	5.07×10^3	5.96×10^2	1.20×10^4	5.38×10^2
	SD	4.76×10^1	1.76×10^3	5.62×10^2	1.27×10^3	9.74×10^2	1.43×10^3	2.50×10^3	2.01×10^3
f_6	MBF	2.42×10^1	2.31×10^1	2.99×10^1	1.26×10^1	0	0	0	0
	SD	6.40	3.06×10^1	6.30	1.61×10^1	0	0	0	0
f_7	MBF	1.01×10^{-14}	1.53×10^{-4}	7.75×10^{-8}	6.53×10^{-10}	8.88×10^{-16}	5.89×10^{-16}	8.88×10^{-16}	5.89×10^{-16}
	SD	3.32×10^{-15}	1.91×10^{-3}	2.79×10^{-7}	2.46×10^{-9}	0	0	0	0
f_8	MBF	4.93×10^{-4}	4.50×10^{-4}	7.40×10^{-4}	4.86×10^{-3}	0	0	0	0
	SD	2.20×10^{-3}	2.80×10^{-2}	2.28×10^{-3}	1.36×10^{-2}	0	0	0	0
f_9	MBF	3.11×10^{-3}	2.88×10^{-7}	3.11×10^{-3}	1.94×10^{-17}	3.15×10^{-2}	2.26×10^{-14}	2.96×10^{-1}	1.15×10^{-32}
	SD	1.39×10^{-2}	7.95×10^{-6}	1.39×10^{-2}	7.21×10^{-17}	1.66×10^{-2}	3.46×10^{-13}	5.03×10^{-1}	1.79×10^{-32}
f_{10}	MBF	5.49×10^{-4}	3.49×10^{-8}	8.63×10^{-34}	2.39×10^{-4}	4.23×10^{-8}	9.67×10^{-13}	2.21	1.56×10^{-32}
	SD	2.46×10^{-3}	4.85×10^{-7}	2.30×10^{-33}	1.69×10^3	5.84×10^{-8}	1.12×10^{-11}	3.38	2.39×10^{-32}
f_{11}	MBF	5.96×10^3	1.07×10^4	1.78×10^3	5.72×10^3	1.79×10^3	1.32×10^4	8.71×10^3	1.14×10^4
	SD	6.22×10^2	2.85×10^3	6.03×10^2	1.64×10^3	4.44×10^2	2.93×10^3	5.30×10^2	3.27×10^3
f_{12}	MBF	1.41×10^3	2.61×10^2	4.91×10^2	2.56×10^2	0	0	0	0
	SD	2.35×10^2	6.73×10^1	1.20×10^2	3.44×10^1	0	0	0	0
f_{13}	MBF	2.07×10^{-4}	7.31×10^{-9}	1.81×10^{-8}	7.97×10^{-9}	0	0	0	0
	SD	5.52×10^{-3}	9.51×10^{-9}	4.09×10^{-7}	2.99×10^{-8}	0	0	0	0
f_{14}	MBF	5.22×10^{-3}	4.58×10^{-3}	4.11×10^{-1}	3.74×10^{-3}	0	0	0	0
	SD	1.19×10^{-2}	6.37×10^{-3}	3.72×10^{-1}	7.97×10^{-3}	0	0	0	0

表 6 100 维实验结果对比
Table 6 Comparison of 100-dimensional test results

函数	指标	CLPSO	HPSO-TVAC	DMS-PSO	LFPSO	PSOLF	RPSOLF	H-PSO-SCAC	SWTPSO
f_1	MBF	4.16×10^{-75}	5.48×10^{-26}	4.89×10^{-77}	2.65×10^{-10}	0	0	0	0
	SD	1.80×10^{-74}	2.59×10^{-25}	1.40×10^{-76}	3.89×10^{-8}	0	0	0	0
f_2	MBF	3.45×10^{-45}	3.77×10^{-15}	5.83×10^2	1.02	0	1.22×10^{-252}	0	0
	SD	1.52×10^{-44}	1.20×10^{-14}	3.54×10^1	3.21	0	0	0	0
f_3	MBF	1.46×10^2	1.34×10^2	1.20×10^2	2.56×10^1	9.04×10^1	9.15×10^1	9.27×10^1	1.03×10^1
	SD	4.78×10^1	2.28×10^2	3.62×10^1	5.23×10^{-1}	2.57×10^1	1.50	4.31	2.55×10^1
f_4	MBF	7.00×10^{-3}	1.17×10^{-48}	6.40×10^{-3}	3.21×10^{-3}	2.35×10^{-5}	8.68×10^{-3}	3.25×10^{-125}	6.28×10^{-1}
	SD	1.53×10^{-3}	1.27×10^{-47}	2.56×10^{-3}	3.24×10^{-3}	1.92×10^{-5}	5.47×10^{-3}	0	1.60×10^{-3}
f_5	MBF	5.36	2.30×10^3	6.58×10^4	3.56×10^4	2.46×10^4	1.29×10^3	2.66×10^4	6.16×10^2
	SD	2.35×10^1	2.88×10^3	5.21×10^1	3.56×10^3	5.75×10^3	2.05×10^3	4.11×10^3	2.08×10^3
f_6	MBF	7.02	4.90×10^1	1.95×10^1	3.25×10^1	0	0	0	0
	SD	1.00×10^2	4.21×10^1	2.59×10^1	5.68×10^1	0	0	0	0
f_7	MBF	2.74×10^{-14}	3.21×10^{-12}	9.91×10^{-1}	3.87×10^{-9}	5.89×10^{-16}	5.89×10^{-16}	5.89×10^{-16}	5.88×10^{-16}

表 6 100 维实验结果对比 (续表)

Table 6 Comparison of 100-dimensional test results (continued table)

函数	指标	CLPSO	HPSO-TVAC	DMS-PSO	LFPSO	PSOLF	RPSOLF	H-PSO-SCAC	SWTPSO
f_8	SD	5.17×10^{-15}	4.29×10^{-11}	4.43	5.89×10^{-9}	0	0	0	0
	MBF	3.33×10^{-17}	2.71×10^{-3}	7.40×10^{-4}	3.89×10^{-2}	0	0	0	0
f_9	SD	7.29×10^{-17}	2.32×10^{-2}	2.28×10^{-3}	5.78×10^{-2}	0	0	0	0
	MBF	9.33×10^{-3}	2.39×10^{-24}	1.56×10^{-2}	9.56×10^{-2}	2.15×10^{-2}	1.27×10^{-12}	5.47×10^{-1}	1.90×10^{-31}
f_{10}	SD	2.87×10^{-2}	7.19×10^{-24}	4.23×10^{-2}	3.46×10^{-2}	3.26×10^{-2}	1.24×10^{-11}	7.08×10^{-1}	5.09×10^{-30}
	MBF	1.10×10^{-3}	1.74×10^{-21}	5.49×10^{-3}	8.97×10^{-2}	6.32×10^{-3}	3.25×10^{-3}	4.50	2.80×10^{-30}
f_{11}	SD	3.38×10^{-3}	7.05×10^{-21}	1.26×10^{-2}	2.65×10^{-2}	2.53×10^{-2}	9.57×10^{-2}	6.71	3.77×10^{-29}
	MBF	6.32×10^3	2.85×10^4	1.12×10^4	3.56×10^4	4.18×10^4	3.43×10^4	2.07×10^4	2.99×10^4
f_{12}	SD	2.32×10^2	5.14×10^3	3.65×10^3	3.25×10^3	1.67×10^2	3.57×10^3	8.63×10^2	7.92×10^3
	MBF	2.56×10^3	5.47×10^2	9.32×10^2	1.25×10^2	0	0	0	0
f_{13}	SD	5.32×10^2	1.03×10^3	1.35×10^2	6.32×10^1	0	0	0	0
	MBF	6.87×10^{-3}	1.05×10^{-6}	3.56×10^{-6}	6.89×10^{-6}	0	0	0	0
f_{14}	SD	3.98×10^{-3}	2.54×10^{-6}	3.89×10^{-5}	1.58×10^{-5}	0	0	0	0
	MBF	2.32×10^{-2}	3.89×10^{-3}	8.25×10^{-1}	9.58×10^{-2}	0	0	0	0
	SD	1.25×10^{-2}	5.89×10^{-3}	9.23×10^{-1}	2.56×10^{-2}	0	0	0	0

edman 检验的评价参数, 该值越小, 表明算法的性能更优. 表 7 给出各算法数据的 Friedman 检验结果, 其中 SWTPSO 在维度 $D = 30, 50, 100$ 维时, 所得秩均值最小, 故该算法性能更优. H-PSO-SCAC 在不同维度下的优化性能较为稳定, RPSOLF、PSOLF、LFPSO 在 30 和 50 维时优化性能较强, 在 100 维时优化性能减弱, CLPSO 和 DMS-PSO 的优化性能随维度的增加逐渐减弱, HPSO-TVAC 在 30 维的优化性能弱于 50 和 100 维.

4.3 收敛性分析

图 5 在维度为 30, 评估次数 20 万次的条件下, 对 SWTPSO 与 CLPSO、HPSO-TVAC、DMS-PSO、LFPSO、PSOLF、RPSOLF 和 H-PSO-SCAC, 在函数 $f_1 \sim f_{14}$ 优化过程中的收敛性能进行分析, 图 5 中 Fitness 表示算法所得适应值. 在单模函数 Sphere、Schwefel2.22 上, 所有算法在 Sphere 函数上都能够求得较高精度的解, 但 SWTPSO 求得最优解所用评估次数更少, 收敛速度最快. 在 Schwefel2.2 上, DMS-PSO 未能求得最优解, 其余算法能求得较高精度的解, 但消耗的评估次数比 SWTPSO 多.

在单模函数 Rosebrock、Quartic 上, H-PSO-SCAC 的优化性能突出, 能求得比其他算法更高精度的解. SWTPSO 相比其他算法, 收敛到较有优势的解且收敛速度较快.

在多模函数 Schwefel2.26 上, CLPSO 的优化性能相比其他算法优势明显, 所求解的精度最高, 收敛速度最快. SWTPSO 对此函数的优化能力相

表 7 各 PSO 算法的 Friedman 检验结果
Table 7 Friedman test results of each POS algorithm

算法	$D = 30$ (排名)	$D = 50$ (排名)	$D = 100$ (排名)
SWTPSO	2.86 (1)	2.46 (1)	2.61 (1)
H-PSO-SCAC	3.86 (4)	3.93 (4)	3.71 (4)
RPSOLF	3.79 (3)	3.46 (3)	3.46 (2)
PSOLF	3.57 (2)	3.39 (2)	3.68 (3)
LFPSO	5.43 (7)	5.25 (5)	6.50 (8)
DMS-PSO	5.32 (6)	5.96 (7)	6.04 (7)
HPSO-TVAC	6.50 (8)	5.57 (6)	5.04 (6)
CLPSO	4.68 (5)	5.96 (7)	4.96 (5)

比其他函数较有优势.

在多模函数 Rastrigin、Ackley、Griewank 上, SWTPSO、PSOLF、RPSOLF 和 H-PSO-SCAC 都能求得最优解, SWTPSO 相比其他算法在收敛速度上, 有较大优势. 其余算法收敛速度慢, 且收敛精度低.

在多模函数 Penalized 1、Penalized 2 上, H-PSO-SCAC 和 PSOLF 收敛精度较低, 其他算法能够收敛到较高精度的解, SWTPSO 的收敛速度更快、精度更高.

在旋转函数 Rotated Schwefel2.26 上, PSOLF 的收敛精度最高, SWTPSO 相比其他算法收敛速度和精度有较大优势.

在旋转函数 Rotated Rastrigin、Rotated Ackley、Rotated Griewank 上, SWTPSO、PSOLF、RPSOLF 和 H-PSO-SCAC 都能求得最优解, SWTPSO 在收敛速度上优势明显, 其余算法未能求得最优解.

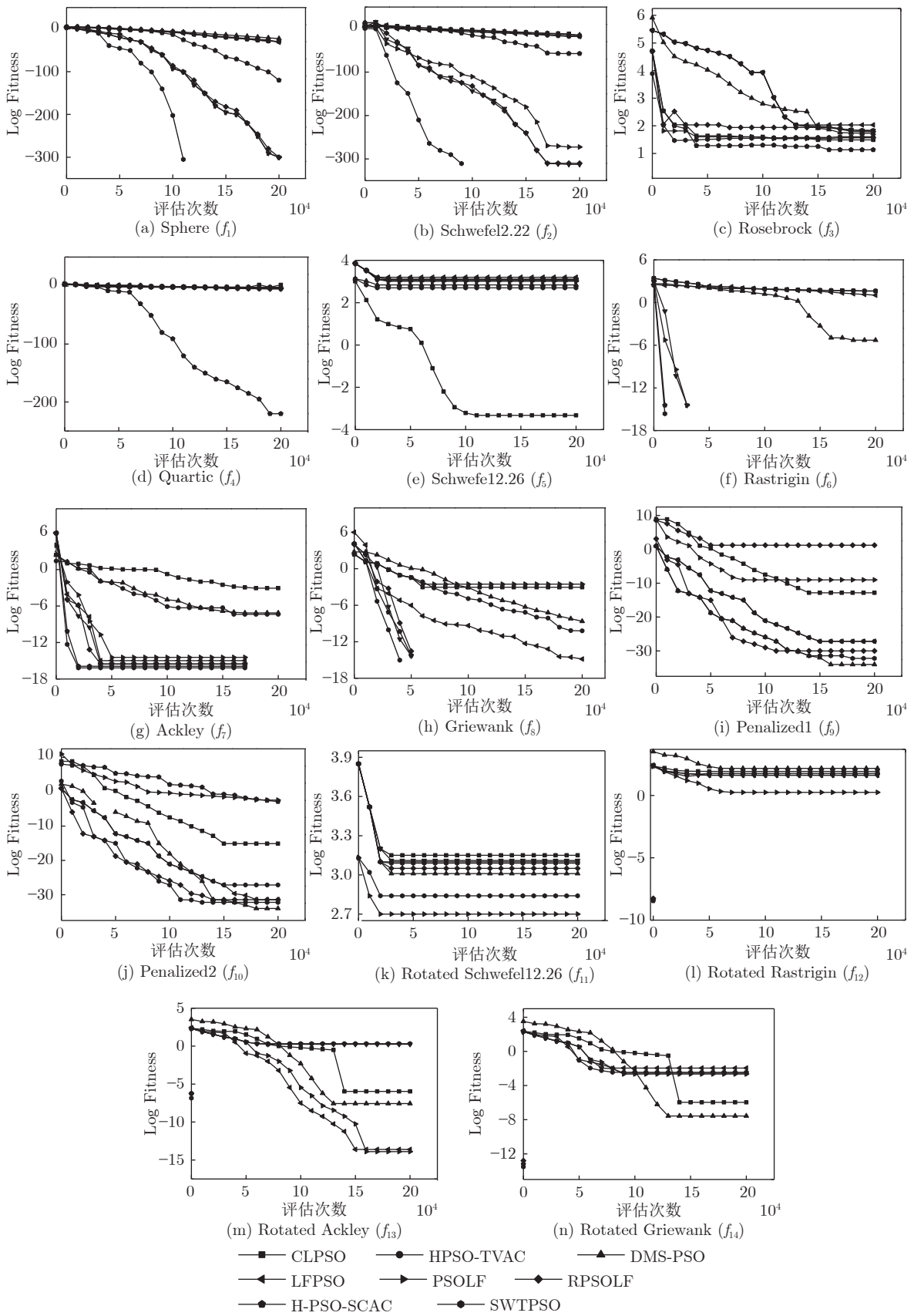


图 5 14 个函数进化曲线图

Fig.5 Evolution curves of 14 functions

4.4 与相关新算法比较

4.4.1 在 50 维的常规基准函数上的实验结果

为进一步体现 SWTPSO 的性能, 将其与改进人工蜂群算法 Bare bones artificial bee colony algorithm with parameter adaptation and fitness-based neighborhood (BABC)^[35]、Accelerating artificial bee colony algorithm with adaptive local search (AABCLS)^[36]、Artificial bee colony algorithm with variable search strategy for continuous optimization (ABCVSS)^[37]、Modified Gbest-guided artificial bee colony algorithm with new probability model (MPGABC)^[38] 改进差分算法 Differential evolution with composite trial vector generation strategies and control parameters (CoDE)^[39]、Adaptive differential evolution with optional external archive (JADE)^[40]、Self-adaptive differential evolution algorithm using population

size reduction and three strategies (jDEscop)^[41]、Differential evolution algorithm with strategy adaptation for global numerical optimization (SaDE)^[42]、以及高效的进化策略 Completely randomized self-adaptation in evolution strategies (CMAES)^[43] 比较. 实验选取第 4.1 节的 26 个基准函数, 设置维度 $D = 50$, 评估次数为 25 万次, 各函数参数设置参照原文献, 所得实验结果见表 8, 表 8 中 MPGABC 的数据来源文献 [38], 其余各算法的数据参照文献 [30]. 由表 8 数据可知, 在测试的 22 个函数中, SWTPSO 在 $f_1 \sim f_7$ 、 $f_{11} \sim f_{13}$ 、 f_{18} 、 f_{20} 共 12 个函数上寻得最优解, 且稳定性强, 相比其他 8 种相关新算法, 优势突出. 在 f_9 和 f_{15} 函数上, SWTPSO 所得最优解, 算法的稳定性都优于其他算法. 在剩余的函数上, SWTPSO 总体优势明显. 将 8 种新相关算法与 SWTPSO 进行 Friedman 检验, 所得结果见表 9. 表 9 中 SWTPSO 所得秩均值最小, 表明其更具优势.

表 8 SWTPSO 与 9 种相关算法实验结果对比

Table 8 Comparison of experimental results of SWTPSO and nine related algorithms

函数	指标	BABC	CoDE	JADE	jDEscop	SaDE	AABCLS	CMAES	ABCVSS	MPGABC	SWTPSO
f_1	MBF	1.01×10^{-14}	1.16×10^{-37}	2.50×10^{-87}	1.15×10^{-36}	1.48×10^{-57}	4.60×10^{-35}	1.21×10^{-28}	6.68×10^{-23}	2.47×10^{-51}	0
	SD	5.07×10^{-14}	2.69×10^{-37}	8.62×10^{-87}	5.75×10^{-36}	6.02×10^{-57}	1.67×10^{-35}	2.09×10^{-29}	3.16×10^{-32}	9.52×10^{-51}	0
f_2	MBF	3.14×10^{-29}	1.25×10^{-34}	4.43×10^{-78}	4.77×10^{-36}	8.97×10^{-55}	4.48×10^{-33}	4.88×10^{-23}	1.11×10^{-23}	4.54×10^{-49}	0
	SD	4.95×10^{-29}	1.50×10^{-34}	2.21×10^{-77}	1.85×10^{-35}	2.38×10^{-54}	2.16×10^{-33}	9.20×10^{-24}	4.17×10^{-23}	1.32×10^{-48}	0
f_3	MBF	1.23×10^{-13}	1.94×10^{-38}	2.27×10^{-85}	1.80×10^{-41}	5.22×10^{-58}	2.54×10^{-35}	4.22×10^{-27}	1.77×10^{-33}	6.05×10^{-53}	0
	SD	6.16×10^{-13}	4.08×10^{-38}	1.14×10^{-82}	1.47×10^{-16}	1.49×10^{-57}	1.24×10^{-35}	9.12×10^{-28}	7.88×10^{-33}	1.39×10^{-52}	0
f_4	MBF	2.96×10^{-92}	6.20×10^{-143}	3.01×10^{-100}	3.58×10^{-124}	3.38×10^{-80}	1.52×10^{-50}	6.82×10^{-13}	7.21×10^{-41}	8.51×10^{-103}	0
	SD	0	3.15×10^{-142}	1.00×10^{-99}	1.79×10^{-123}	1.59×10^{-79}	4.88×10^{-50}	5.39×10^{-13}	3.56×10^{-40}	3.61×10^{-102}	0
f_5	MBF	2.18×10^{-6}	1.21×10^{-20}	1.89×10^{-41}	1.71×10^{-22}	1.71×10^{-38}	4.98×10^{-18}	3.69×10^{-4}	2.89×10^{-18}	2.95×10^{-28}	1.13×10^{-300}
	SD	1.06×10^{-5}	8.94×10^{-21}	8.32×10^{-41}	8.46×10^{-22}	7.47×10^{-38}	1.34×10^{-18}	1.84×10^{-3}	6.52×10^{-18}	4.04×10^{-28}	0
f_6	MBF	7.20	6.00×10^{-5}	8.20×10^{-10}	2.52	3.42×10^{-1}	1.16×10^{-1}	6.00×10^{-15}	2.06	2.12×10^1	0
	SD	2.78	9.51×10^{-5}	9.51×10^{-10}	5.61×10^{-1}	4.67×10^{-1}	1.24×10^{-1}	7.45×10^{-16}	3.77×10^{-1}	4.04	0
f_7	MBF	0	0	0	0	0	0	1.60×10^{-1}	0	0	0
	SD	0	0	0	0	0	0	4.73×10^{-1}	0	0	0
f_8	MBF	2.67×10^{-109}	2.67×10^{-109}	2.67×10^{-109}	2.67×10^{-109}	2.67×10^{-109}	2.67×10^{-109}	5.40×10^{-66}	2.67×10^{-109}	2.67×10^{-109}	2.96×10^{-109}
	SD	2.62×10^{-116}	9.67×10^{-125}	9.60×10^{-125}	9.65×10^{-125}	3.71×10^{-122}	2.77×10^{-119}	1.99×10^{-65}	3.08×10^{-120}	3.06×10^{-122}	1.14×10^{-111}
f_9	MBF	5.74×10^{-2}	8.17×10^{-3}	2.50×10^{-3}	3.91×10^{-3}	1.33×10^{-2}	1.63×10^{-2}	2.80×10^{-1}	6.14×10^{-2}	5.02×10^{-2}	6.61×10^{-4}
	SD	1.11×10^{-2}	2.79×10^{-3}	1.54×10^{-3}	1.47×10^{-3}	3.33×10^{-3}	4.68×10^{-3}	6.70×10^{-2}	1.38×10^{-2}	7.98×10^{-3}	1.37×10^{-2}
f_{10}	MBF	6.29×10^{-2}	3.32×10^1	3.19×10^{-1}	2.58×10^1	9.45	3.09	1.75×10^{-25}	1.09×10^{-1}	4.32×10^{-1}	8.19
	SD	1.24×10^{-1}	2.26×10^1	1.10	2.68×10^1	2.08×10^1	1.36×10^1	4.31×10^{-26}	2.52×10^{-1}	7.44×10^{-1}	5.68×10^1
f_{15}	MBF	0	7.34×10^{-1}	1.78×10^{-11}	1.03×10^{-14}	6.77×10^{-1}	0	3.89×10^2	0	0	0
	SD	0	8.82×10^{-1}	1.74×10^{-11}	1.31×10^{-14}	6.87×10^{-1}	0	7.11×10^1	0	0	0
f_{16}	MBF	0	2.28×10^1	2.74×10^{-8}	8.02×10^{-2}	4.40×10^{-1}	8.00×10^{-2}	3.78×10^2	0	0	0
	SD	0	4.68	2.07×10^{-8}	2.77×10^{-1}	6.51×10^{-1}	2.77×10^{-1}	4.90×10^1	0	0	0
f_{17}	MBF	0	2.96×10^{-4}	7.88×10^{-4}	1.47×10^{-16}	6.88×10^{-3}	4.44×10^{-18}	1.38×10^{-3}	3.67×10^{-14}	0	0
	SD	0	1.48×10^{-3}	2.82×10^{-3}	2.60×10^{-16}	1.22×10^{-2}	2.22×10^{-17}	3.34×10^{-3}	1.84×10^{-13}	0	0

表 8 SWTPSO 与 9 种相关算法实验结果对比 (续表)

Table 8 Comparison of experimental results of SWTPSO and nine related algorithms (continued table)

函数	指标	BABC	CoDE	JADE	jDEscop	SaDE	AABCLS	CMAES	ABCVSS	MPGABC	SWTPSO
f_{18}	MBF	3.78×10^{-12}	4.74	4.75	6.18×10^1	3.64×10^{-12}	1.41×10^{-11}	9.01×10^3	4.66×10^{-12}	1.05×10^{-11}	5.38×10^2
	SD	7.28×10^{-13}	2.37×10^1	2.32×10^1	1.69×10^2	0	3.53×10^{-12}	1.13×10^3	2.47×10^{-12}	3.03×10^{-12}	2.01×10^3
f_{19}	MBF	7.50×10^{-15}	2.81×10^{-15}	6.22×10^{-15}	1.43×10^1	1.32	2.65×10^{-14}	1.99×10^1	1.70×10^{-14}	2.51×10^{-14}	5.89×10^{-16}
	SD	2.49×10^{-15}	7.11×10^{-16}	0	6.50	4.90×10^{-1}	3.48×10^{-15}	2.63×10^{-2}	5.75×10^{-33}	4.55×10^{-15}	0
f_{20}	MBF	1.22×10^{-13}	9.42×10^{-33}	2.49×10^{-3}	1.91×10^{-31}	3.24×10^{-2}	9.42×10^{-33}	4.98×10^{-3}	1.07×10^{-32}	9.42×10^{-33}	1.15×10^{-32}
	SD	6.09×10^{-13}	1.40×10^{-48}	1.24×10^{-2}	5.28×10^{-31}	9.01×10^{-2}	1.40×10^{-48}	1.72×10^{-2}	5.74×10^{-33}	1.40×10^{-48}	1.79×10^{-32}
f_{21}	MBF	2.50×10^{-15}	1.55×10^{-33}	1.01×10^{-4}	5.02×10^{-32}	1.09×10^{-2}	1.50×10^{-33}	8.01×10^3	1.80×10^{-33}	1.50×10^{-33}	1.56×10^{-32}
	SD	1.25×10^{-14}	2.47×10^{-34}	8.05×10^{-5}	5.95×10^{-32}	3.00×10^{-2}	0	6.25×10^3	1.08×10^{-33}	0	2.39×10^{-32}
f_{22}	MBF	2.07×10^{-16}	4.39×10^{-3}	1.01×10^{-4}	2.95×10^{-5}	1.75×10^{-16}	5.32×10^{-11}	8.59×10^{-1}	2.14×10^{-16}	3.21×10^{-7}	3.10×10^{-290}
	SD	7.79×10^{-16}	6.44×10^{-3}	8.05×10^{-5}	3.09×10^{-5}	2.63×10^{-16}	9.68×10^{-11}	8.49×10^{-1}	7.43×10^{-16}	4.58×10^{-7}	0
f_{23}	MBF	1.35×10^{-31}	1.35×10^{-31}	1.35×10^{-31}	1.30×10^{-30}	7.39×10^{-2}	1.35×10^{-31}	3.77×10^{-1}	1.35×10^{-31}	1.35×10^{-31}	1.78×10^{-31}
	SD	2.23×10^{-47}	2.47×10^{-33}	2.23×10^{-47}	3.69×10^{-30}	1.07×10^{-1}	2.23×10^{-47}	1.22	2.23×10^{-47}	2.23×10^{-47}	2.83×10^{-31}
f_{24}	MBF	0	3.45	3.33×10^{-1}	0	4.17×10^{-4}	2.56×10^{-6}	9.41	0	1.56×10^{-2}	0
	SD	0	2.94×10^{-1}	4.45×10^{-2}	0	5.94×10^{-4}	7.39×10^{-6}	4.19	0	2.04×10^{-2}	0
f_{25}	MBF	-7.83×10^1	-7.83×10^1	-7.83×10^1	-7.83×10^1	-7.83×10^1	-7.83×10^1	-6.43×10^1	-7.83×10^1	-7.83×10^1	-6.26×10^1
	SD	1.16×10^{-14}	4.10×10^{-15}	2.26×10^{-1}	1.12×10^{-13}	1.13×10^{-1}	4.10×10^{-15}	2.63	1.00×10^{-14}	1.12×10^{-14}	3.31
f_{26}	MBF	-5.00×10^1	-4.86×10^1	-4.98×10^1	-5.00×10^1	-4.83×10^1	-5.00×10^1	-4.10×10^1	-5.00×10^1	-5.00×10^1	-3.99×10^1
	SD	6.79×10^{-6}	4.45×10^{-1}	3.86×10^{-2}	9.23×10^{-3}	2.70×10^{-1}	9.13×10^{-4}	2.65	3.82×10^{-7}	4.26×10^{-1}	2.13×10^1

4.4.2 在 100 维的常规基准函数上的实验结果

将改进的和声搜索算法 Global dynamic harmony search algorithm (GDHS)^[44]、An improved global-best harmony search algorithm (IGHS)^[45]、差分算法 Opposition-based differential evolution (ODE)^[46]、人工蜂群算法 Gbest-guided artificial bee colony algorithm for numerical function opti-

mization (GABC)^[47]、粒子群 Improved global-best-guided particle swarm optimization with learning operation for global optimization problems (IG-PSO)^[48] 和本文算法比较, 维度为 100, 评估次数为 50 万次, 各算法独立运行 50 次, 算法所测函数参考文献 [48] 的表 2~3, 数据来源参考文献 [48] 中表 9~10. 实验所得结果见表 10, 本文算法在 $f_1 \sim f_4$ 、

表 9 8 种新相关算法和 SWTPSO 的 Friedman 测试结果

Table 9 Friedman test results of 8 new correlation algorithms and SWTPSO

算法	SWTPSO	MPGABC	AABCLS	JADE	ABCVSS	BABC	CoDE	jDEscop	SaDE	CMAES
平均值 (排名)	3.86 (1)	4.41 (2)	5.11 (3)	5.05 (4)	5.07 (5)	5.11 (6)	5.59 (7)	5.91 (8)	6.07 (9)	8.82 (10)

表 10 6 种算法在 15 个函数上的比较结果

Table 10 Results of the fifteen functions for six algorithms

函数	指标	GDHS	ODE	IGHS	GABC	IGPSO	SWTPSO
f_1	MBF	8.07×10^{-3}	5.67×10^{-81}	2.14×10^{-10}	8.58×10^2	4.45×10^{-156}	0
	SD	8.87×10^{-4}	9.55×10^{-81}	1.22×10^{-11}	1.52×10^3	9.29×10^{-156}	0
f_2	MBF	7.20×10^{-1}	3.60×10^{-6}	3.96×10^{-1}	8.47×10^1	0	0
	SD	4.16×10^{-2}	1.13×10^{-5}	3.07×10^{-1}	3.06×10^1	0	0
f_3	MBF	6.17	1.81×10^{-5}	2.72×10^{-6}	8.49×10^1	0	0
	SD	7.70×10^{-1}	4.69×10^{-5}	9.49×10^{-7}	3.51×10^1	0	0
f_4	MBF	6.71×10^{-2}	9.27×10^{-2}	3.53	5.34×10^1	0	0
	SD	1.05×10^{-2}	2.33×10^{-2}	2.67×10^{-1}	1.93×10^1	0	0
f_5	MBF	1.07×10^2	9.69×10^{-12}	1.32×10^2	6.07×10^5	4.18×10^{-2}	1.03×10^1
	SD	2.39×10^1	3.06×10^{-11}	4.49×10^1	8.23×10^5	2.86×10^{-2}	2.55×10^1

表 10 6 种算法在 15 个函数上的比较结果 (续表)

Table 10 Results of the fifteen functions for six algorithms (continued table)

函数	指标	GDHS	ODE	IGHS	GABC	IGPSO	SWTPSO
f_6	MBF	0	0	0	2.35×10^3	0	0
	SD	0	0	0	2.90×10^3	0	0
f_7	MBF	2.87×10^{-3}	1.83×10^{-3}	1.80×10^{-2}	8.57×10^{-1}	1.40×10^{-4}	6.28×10^{-4}
	SD	5.74×10^{-4}	2.02×10^{-3}	2.17×10^{-3}	6.27×10^{-1}	1.45×10^{-4}	1.60×10^{-3}
f_8	MBF	-4.19×10^4	-4.17×10^4	-3.99×10^4	-4.20×10^4	-4.19×10^4	-3.34×10^4
	SD	4.57×10^{-2}	4.35×10^{-2}	9.76×10^{-1}	4.57×10^{-2}	1.33×10^{-2}	1.28×10^4
f_9	MBF	1.16×10^{-2}	4.52×10^{-8}	4.41×10^{-8}	4.64×10^1	0	0
	SD	1.14×10^{-3}	1.41×10^{-7}	3.47×10^{-9}	1.63×10^1	0	0
f_{10}	MBF	1.44×10^{-2}	1.72×10^{-4}	5.91×10^{-6}	7.03	3.55×10^{-15}	5.88×10^{-16}
	SD	6.71×10^{-4}	5.44×10^{-4}	1.63×10^{-7}	2.47	0	0
f_{11}	MBF	4.50×10^{-3}	7.40×10^{-4}	2.28×10^{-3}	2.14×10^1	0	0
	SD	5.18×10^{-4}	2.28×10^{-3}	4.27×10^{-3}	3.00×10^1	0	0
f_{12}	MBF	6.83×10^{-6}	3.49×10^{-25}	5.26×10^{-13}	2.93×10^{-1}	1.50×10^{-7}	1.90×10^{-31}
	SD	6.46×10^{-7}	8.66×10^{-25}	4.25×10^{-14}	9.27×10^{-1}	3.70×10^{-8}	5.09×10^{-30}
f_{13}	MBF	3.33×10^{-4}	1.07×10^{-15}	1.86×10^{-6}	2.04×10^2	1.75×10^{-5}	2.80×10^{-30}
	SD	4.66×10^{-5}	5.60×10^{-6}	5.89×10^{-6}	4.44×10^2	5.60×10^{-6}	3.77×10^{-29}
f_{14}	MBF	1.94×10^1	1.23×10^1	4.06×10^1	3.17×10^1	0	0
	SD	1.20	7.70	7.70	5.15	0	0
f_{15}	MBF	1.33×10^{-1}	4.29×10^2	3.15×10^1	2.18×10^2	0	0
	SD	1.61×10^{-2}	8.19×10^2	6.83×10^1	2.42×10^4	0	0

f_6 、 f_9 、 f_{11} 、 $f_{14} \sim f_{15}$ 共 9 个函数中求得最优解, 在 f_5 、 f_7 、 f_8 、 f_{10} 、 $f_{12} \sim f_{13}$ 共 6 个函数中求得最优解优势明显, 将 6 种算法与 SWTPSO 进行 Friedman 检验, 所得结果见表 11, 表 11 中 SWTPSO 所得秩均值最小, 表明其更具优势。

4.4.3 在 30 维的 CEC2015 测试集上的实验结果

群智能算法经过近几年的大量研究, 绝大多数改进算法对较早版本的标准测试函数有较好优化效果. 因此, 本文采用新近提出的测试函数 CEC2015^[49] 进一步验证算法的优化性能, 表 12 为 CEC2015 函

表 11 6 种算法的 Friedman 检验结果
Table 11 Friedman test results of 6 algorithms

算法	平均值 (排名)
SWTPSO	1.83 (1)
IGPSO	1.97 (2)
ODE	3.13 (3)
IGHS	3.93 (4)
GDHS	4.20 (5)
GABC	5.93 (6)

表 12 CEC2015 函数集
Table 12 CEC2015 test suite

函数序号	函数类型	描述	最优值
1	单峰函数	Bent Cigar 旋转函数	100
2		Discus 旋转函数	200
3	简单多峰函数	Schwefel 偏移旋转函数	300
4		Schwefel 偏移旋转函数	400
5		Katsuura 偏移旋转函数	500
6		HappyCat 偏移旋转函数	600
7		HGBat 偏移旋转函数	700
8		Griewank + Rosenbrock 扩展偏移旋转函数	800
9		Schaffer's F6 扩展偏移旋转函数	900

表 12 CEC2015 函数集 (续表)

Table 12 CEC2015 test suite (continued table)

函数序号	函数类型	描述	最优值
10	混合函数	混合函数 1 ($N = 3$)	1000
11		混合函数 2 ($N = 4$)	1100
12		混合函数 3 ($N = 5$)	1200
13	组合函数	组合函数 1 ($N = 5$)	1300
14		组合函数 2 ($N = 3$)	1400
15		组合函数 3 ($N = 5$)	1500

数集.

SWTPSO 分别与新近提出的萤火虫与粒子群混合优化算法 (Fire fly particle swarm optimization, FFPSO)^[50], Hybrid particle swarm optimization and fire fly (HPSOFF)^[51]、A hybrid algorithm combining firefly and particle swarm optimization (HFPSO)^[27]比较, 维度 $D = 30$, 评估次数为 1 500, 实验数据和其他参数设置参见文献 [27].

表 13 为各算法测函数 CEC 2015 的实验结果, SWTPSO 在所测 15 个函数中, 有 14 个函数占较大优势. 表 14 对各算法进行 Friedman 检验, SWTPSO 所得秩均值最小, 再次验证了本文算法更具优势.

5 结束语

为提高粒子群算法全局最优的质量, 增强粒子逃离局部最优的能力, 本文提出方波触发机制, 在

种群初始化后, 在前半个方波周期内进行全局勘探, 探索更多新解空间区域, 在后半个方波周期内转换为局部开发, 对该区域进行精细搜索. 在下个周期将自动获得与粒子当前位置有关的新速度, 使粒子获得的运动步长多变, 种群多样性增强. 在第 3.1 节中通过实验测定出方波触发机制的最佳周期, 并在第 3.2 节中证明, 方波触发机制通过为粒子提供多变步长, 提升种群的多样性. 在多类型测试函数上与新改进粒子群算法进行 30、50、100 维的测试, 且与其他新种类智能算法比较, 结果都表明 SWTPSO 优化性能更具优势.

但是, SWTPSO 在高维情况下测试复杂函数时, 仍易陷入局部最优. 下一步的研究工作可从以下方面展开: 在方波触发机制中, 使用标准 PSO 公式进行全局勘探, 未考虑进化后期标准 PSO 中粒子的速度将变小, 全局勘探能力减弱的问题, 下一步的研究中, 可提出改进的更新公式, 承担全局勘探任务.

表 13 CEC2015 实验结果对比

Table 13 Test results comparison of CEC2015 test suite

函数符号	PSO	FA	FFPSO	HPSOFF	HFPSO	SWTPSO
1 MBF	3.9049×10^9	2.8899×10^{10}	9.2383×10^{10}	4.7539×10^9	1.1795×10^9	3.0501×10^8
2 MBF	9.9760×10^4	1.3418×10^5	6.9430×10^6	9.7376×10^4	8.5653×10^4	5.6540×10^4
3 MBF	3.3113×10^2	3.3850×10^2	3.4771×10^2	3.3059×10^2	3.2638×10^2	3.2631×10^2
4 MBF	7.7928×10^3	8.0330×10^3	9.6696×10^3	6.8199×10^2	5.1202×10^3	5.8445×10^3
5 MBF	5.0418×10^2	5.0425×10^2	5.0586×10^2	5.0422×10^2	5.0410×10^2	5.0403×10^2
6 MBF	6.0096×10^2	6.0410×10^2	6.0755×10^2	6.0090×10^2	6.0076×10^2	6.0053×10^2
7 MBF	7.0405×10^2	7.6997×10^2	8.9401×10^2	7.0750×10^2	7.0074×10^2	7.0046×10^2
8 MBF	4.0013×10^3	2.3491×10^6	1.6032×10^8	1.7191×10^4	2.6354×10^3	1.5070×10^3
9 MBF	9.1358×10^2	9.1381×10^2	9.1427×10^2	9.1365×10^2	9.1337×10^2	9.1328×10^2
10 MBF	7.5602×10^6	2.9938×10^7	3.9391×10^8	1.1337×10^7	5.4690×10^6	1.1662×10^7
11 MBF	1.1614×10^3	1.2889×10^3	2.1094×10^3	1.1551×10^3	1.1336×10^3	1.1267×10^3
12 MBF	2.2417×10^3	2.9655×10^3	4.9876×10^5	2.0617×10^3	1.7752×10^3	1.7819×10^3
13 MBF	1.7719×10^3	1.9596×10^3	3.6329×10^3	1.7390×10^3	1.6866×10^3	1.6552×10^3
14 MBF	1.6644×10^3	1.7522×10^3	2.1683×10^3	1.6711×10^3	1.6469×10^3	1.6510×10^3
15 MBF	2.5522×10^3	2.8256×10^3	3.9013×10^3	2.6529×10^3	2.4467×10^3	2.4639×10^3

表 14 Friedman 检验结果
Table 14 Results of Friedman test

算法	秩均值 (排名)
SWTPSO	1.53 (1)
HFPSO	1.73 (2)
HPSOFF	3.33 (3)
PSO	3.40 (4)
FA	5.00 (5)
FFPSO	6.00 (6)

References

- Kennedy J, Eberhart R C. Particle swarm optimization. In: Proceedings of the 1995 International Conference on Neural Networks. Perth, Australia: 1995. 1942–1948
- Jia Cheng-Feng, Han Hua, Lü Ya-Nan, Zhang Lu. Link prediction algorithm based on word2vec and particle swarm. *Acta Automatica Sinica*, 2020, **46**(8): 1703–1713
(贾承丰, 韩华, 吕亚楠, 张路. 基于word2vec和粒子群的链路预测算法. *自动化学报*, 2020, **46**(8): 1703–1713)
- Chou J S, Pham A D. Nature-inspired metaheuristic optimization in least squares support vector regression for obtaining bridge scour information. *Information Sciences*, 2017, **399**: 64–80
- Tran B, Xue B, Zhang M. Variable-length particle swarm optimization for feature selection on high-dimensional classification. *IEEE Transactions on Evolutionary Computation*, 2018, **23**(3): 473–487
- Mousavi-Avval S H, Rafiee S, Sharifi M, Hosseinpour S, Notaricola B, Tassielli G, et al. Application of multi-objective genetic algorithms for optimization of energy, economics and environmental life cycle assessment in oilseed production. *Journal of Cleaner Production*, 2017, **140**: 804–815
- Chen S M, Huang Z C. Multi-attribute decision making based on interval-valued intuitionistic fuzzy values and particle swarm optimization techniques. *Information Sciences*, 2017, **397**: 206–218
- Wu Qing, Zhao Tao, Dian Song-Yi, Guo Rui, Li Sheng-Chuang, Fang Hong-Wei, et al. General type-2 fuzzy logic control for a power-line inspection robot based on FPSO. *Acta Automatica Sinica*, 2022, **48**(6): 1482–1492
(吴庆, 赵涛, 佃松宜, 郭锐, 李胜川, 方红伟, 等. 基于FPSO的电力巡检机器人的广义二型模糊逻辑控制. *自动化学报*, 2022, **48**(6): 1482–1492)
- Shi Y, Eberhart R. A modified particle swarm optimizer. In: Proceedings of the 1998 IEEE International Conference on Evolutionary Computation. Anchorage, USA: IEEE, 1998. 69–73
- Zhan Z H, Zhang J, Li Y, Chung H S H. Adaptive particle swarm optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 2009, **39**(6): 1362–1381
- Zhang L, Tang Y, Hua C, Guan X. A new particle swarm optimization algorithm with adaptive inertia weight based on Bayesian techniques. *Applied Soft Computing*, 2015, **28**: 138–149
- Ratnaweera A, Halgamuge S K, Watson H C. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *IEEE Transactions on Evolutionary Computation*, 2004, **8**(3): 240–255
- Chen K, Zhou F, Yin L, Wang S, Wang Y, Wan F. A hybrid particle swarm optimizer with sine cosine acceleration coefficients. *Information Sciences*, 2018, **422**: 218–241
- Li S F, Cheng C Y. Particle swarm optimization with fitness adjustment parameters. *Computers & Industrial Engineering*, 2017, **113**: 831–841
- Isiet M, Gadala M. Self-adapting control parameters in particle swarm optimization. *Applied Soft Computing*, 2019, **83**: 105653
- Kennedy J. Bare bones particle swarms. In: Proceedings of the 2003 IEEE Swarm Intelligence Symposium. Indianapolis, USA: 2003. 80–87
- Wang H, Wu Z, Rahnamayan S, Liu Y, Ventresca M. Enhancing particle swarm optimization using generalized opposition-based learning. *Information Sciences*, 2011, **181**(20): 4699–4714
- Yan B, Zhao Z, Zhou Y, Yuan W, Li J, Wu J, et al. A particle swarm optimization algorithm with random learning mechanism and Levy flight for optimization of atomic clusters. *Computer Physics Communications*, 2017, **219**: 79–86
- Xia X, Xing Y, Wei B, Zhang Y, Li X, Deng X, et al. A fitness-based multi-role particle swarm optimization. *Swarm and Evolutionary Computation*, 2019, **44**: 349–364
- Liu H R, Cui J C, Lu Z D, Liu D Y, Deng Y J. A hierarchical simple particle swarm optimization with mean dimensional information. *Applied Soft Computing*, 2019, **76**: 712–725
- Kennedy J, Mendes R. Neighborhood topologies in fully informed and best-of-neighborhood particle swarms. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 2006, **36**(4): 515–519
- Mendes R, Kennedy J, Neves J. The fully informed particle swarm: Simpler, maybe better. *IEEE Transactions on Evolutionary Computation*, 2004, **8**(3): 204–210
- Lim W H, Isa N A M. Particle swarm optimization with adaptive time-varying topology connectivity. *Applied Soft Computing*, 2014, **24**: 623–642
- Lin A, Sun W, Yu H, Wu G, Tang H. Global genetic learning particle swarm optimization with diversity enhancement by ring topology. *Swarm and Evolutionary Computation*, 2019, **44**: 571–583
- Zhang K, Huang Q, Zhang Y. Enhancing comprehensive learning particle swarm optimization with local optima topology. *Information Sciences*, 2019, **471**: 1–18
- Alatas B, Akin E, Ozer A B. Chaos embedded particle swarm optimization algorithms. *Chaos, Solitons & Fractals*, 2009, **40**(4): 1715–1734
- Kao Y T, Zahara E. A hybrid genetic algorithm and particle swarm optimization for multimodal functions. *Applied Soft Computing*, 2008, **8**(2): 849–857
- Aydilek İ B. A hybrid firefly and particle swarm optimization algorithm for computationally expensive numerical problems. *Applied Soft Computing*, 2018, **66**: 232–249
- Lynn N, Suganthan P N. Ensemble particle swarm optimizer. *Applied Soft Computing*, 2017, **55**: 533–548
- Wang F, Zhang H, Li K, Lin Z, Yang J, Shen X. A hybrid particle swarm optimization algorithm using adaptive learning strategy. *Information Sciences*, 2018, **436**: 162–177
- Cui L, Li G, Lin Q, Du Z, Gao W, Chen J, et al. A novel artificial bee colony algorithm with depth-first search framework and elite-guided search equation. *Information Sciences*, 2016, **367**: 1012–1044
- Liang J J, Qin A K, Suganthan P N, Baskar S. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Transactions on Evolutionary Computation*, 2006, **10**(3): 281–295
- Hakhi H, Uğuz H. A novel particle swarm optimization algorithm with levy flight. *Applied Soft Computing*, 2014, **23**: 333–345
- Liang J J, Suganthan P N. Dynamic multi-swarm particle swarm optimizer. In: Proceedings of the 2005 IEEE Swarm In-

telligence Symposium. Pasadena, USA: 2005. 124–129

- 34 Jensi R, Jiji G W. An enhanced particle swarm optimization with levy flight for global optimization. *Applied Soft Computing*, 2016, **43**: 248–261
- 35 Gao W, Chan F T S, Huang L, Liu S. Bare bones artificial bee colony algorithm with parameter adaptation and fitness-based neighborhood. *Information Sciences*, 2015, **316**: 180–200
- 36 Jadon S S, Bansal J C, Tiwari R, Sharma H. Accelerating artificial bee colony algorithm with adaptive local search. *Memetic Computing*, 2015, **7**(3): 215–230
- 37 Kiran M S, Hakli H, Gunduz M, Uguz H. Artificial bee colony algorithm with variable search strategy for continuous optimization. *Information Sciences*, 2015, **300**: 140–157
- 38 Cui L, Zhang K, Li G, Fu X, Wen Z, Lu N, et al. Modified Gbest-guided artificial bee colony algorithm with new probability model. *Soft Computing*, 2018, **22**(7): 2217–2243
- 39 Wang Y, Cai Z, Zhang Q. Differential evolution with composite trial vector generation strategies and control parameters. *IEEE Transactions on Evolutionary Computation*, 2011, **15**(1): 55–66
- 40 Zhang J, Sanderson A C. JADE: Adaptive differential evolution with optional external archive. *IEEE Transactions on Evolutionary Computation*, 2009, **13**(5): 945–958
- 41 Brest J, Maucec, M S. Self-adaptive differential evolution algorithm using population size reduction and three strategies. *Soft Computing*, 2011, **15**(11): 2157–2174
- 42 Qin A K, Huang V L, Suganthan P N. Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Transactions on Evolutionary Computation*, 2009, **13**(2): 398–417
- 43 Hansen N, Ostermeier A. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 2001, **9**(2): 159–195
- 44 Khalili M, Kharrat R, Salahshoor K, Sefat M H. Global dynamic harmony search algorithm: GDHS. *Applied Mathematics and Computation*, 2014, **228**: 195–219
- 45 El-Abd M. An improved global-best harmony search algorithm. *Applied Mathematics and Computation*, 2013, **222**: 94–106
- 46 Rahnamayan S, Tizhoosh H R, Salama M M A. Opposition-based differential evolution. *IEEE Transactions on Evolutionary Computation*, 2008, **12**(1): 64–79
- 47 Zhu G, Kwong S. Gbest-guided artificial bee colony algorithm for numerical function optimization. *Applied Mathematics and Computation*, 2010, **217**(7): 3166–3173
- 48 Ouyang H, Gao L, Li S, Kong X Y. Improved global-best-guided particle swarm optimization with learning operation for global optimization problems. *Applied Soft Computing*, 2017, **52**: 987–1008
- 49 Liang J J, Qu B Y, Suganthan P N, Chen Q. Problem Definitions and Evaluation criteria for the CEC 2015 Competition on Learning-based Real-parameter Single Objective Optimization. Technical Report 201411A, Computational Intelligence Laboratory, Zhengzhou University, China, Nanyang Technological University, Singapore, 2014. 29: 625–640
- 50 Kora P, Krishna K S R. Hybrid firefly and particle swarm optimization algorithm for the detection of bundle branch block. *International Journal of the Cardiovascular Academy*, 2016, **2**(1): 44–48
- 51 Arunachalam S, AgnesBhomila T, Babu M R. Hybrid particle swarm optimization algorithm and firefly algorithm based combined economic and emission dispatch including valve point effect. In: Proceedings of the 2014 International Conference on Swarm, Evolutionary, and Memetic Computing. Odisha, India: 2014. 647–660



邓志诚 南昌工程学院硕士研究生。主要研究方向为群智能算法。

E-mail: deng_zc215@163.com

(DENG Zhi-Cheng Master student at Nanchang Institute of Technology. His main research interest is swarm intelligence algorithm.)



孙辉 南昌工程学院教授。1988年获清华大学硕士学位, 2002年获南昌大学博士学位。主要研究方向为群智能算法, 粗糙集和变分不等原理。本文通信作者。

E-mail: sun_hui2006@163.com

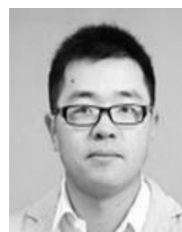
(SUN Hui Professor at Nanchang Institute of Technology. He received his master degree from Tsinghua University in 1988 and received his Ph.D. degree from Nanchang University in 2002, respectively. His research interest covers swarm intelligent algorithm, rough sets and variational inequality principle. Corresponding author of this paper.)



赵嘉 南昌工程学院教授。主要研究方向为群智能算法与数据挖掘。

E-mail: zhaojia925@163.com

(ZHAO Jia Professor at Nanchang Institute of Technology. His research interest covers swarm intelligent algorithm and data mining.)



王晖 南昌工程学院教授。主要研究方向为群智能算法与水资源优化。

E-mail: huiwang@nit.edu.cn

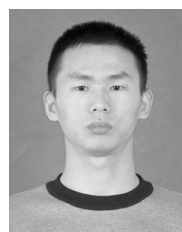
(WANG Hui Professor at Nanchang Institute of Technology. His research interest covers swarm intelligent algorithm and water resources optimization.)



吕莉 南昌工程学院教授。主要研究方向为群智能算法与目标跟踪。

E-mail: lvli623@163.com

(LV Li Professor at Nanchang Institute of Technology. Her research interest covers swarm intelligent algorithm and single target track.)



谢海华 南昌工程学院硕士研究生。主要研究方向为群智能算法。

E-mail: pxlh_xhh@163.com

(XIE Hai-Hua Master student at Nanchang Institute of Technology. His main research interest is swarm intelligence algorithm.)