一种混合梯度搜索的改进粒子群优化算法

摘 要：粒子群优化算法（PSO）是一种基于群体智能的随机搜索算法，具有较强的全局搜索能力且易于安装实现。因此，PSO 算法现已受到生物信息学、图像处理、信号处理等众多研究领域的广泛关注。然而，传统 PSO 算法存在重复搜索、精度不高等问题。针对上述问题，提出一种混合梯度搜索的改进粒子群优化算法，简记为 GIPSO（Gradient Information Based PSO）。此算法主要是利用目标函数的梯度信息指挥粒子的搜索行为，有效降低粒子的重复搜索，以提升算法的收敛速度与精度。首先，在线性衰减惯性权重的粒子群算法（LDWPSO）的速度公式中加入梯度下降法，有效避免粒子的重复搜索行为，提升算法的收敛精度；其次，随机选中半数粒子按照新速度公式进行迭代，其余粒子按照原速度公式进行迭代，大幅降低算法的执行时间。仿真实验结果证明，GIPSO 算法比传统算法有更少的迭代计算次数以及更高的收敛精度。

关键词：机器学习；粒子群；梯度搜索；群智能搜索

An Hybrid Particle Swarm Optimization Method Based on Gradient Search

Abstract: Particle Swarm Optimization (PSO) is a stochastic search method based on population intelligence, which has strong global search capabilities and is easy to implement. Therefore, the PSO method has received extensive attention in many research fields, such as bioinformatics, image processing, signal processing, and so on. However, the traditional PSO method suffers from the problems of repetitive search and low accuracy. An Hybrid Particle Swarm Optimization Method Based on Gradient Search, abbreviated as GIPSO (Gradient Information Based PSO), is proposed in this article. The GIPSO method uses the gradient information of the objective function to regulate the search behavior of particles, and improve the convergence speed and accuracy. On one side, added to the gradient descent method the velocity formula of the Linear Decay-inertial Weight Particle Swarn Optimization，(LDWPSO), which effectively avoids the repetitive searching behavior of the particles and improves the convergence accuracy of the method; On the other side, one half of the particles are randomly selected to iterate by the new velocity formula, and the rest of the particles iterate by the original velocity formula, which significantly reduces the execution time of the method. The results of simulation experiments prove that the GIPSO method has fewer iteration calculations and higher convergence accuracy than the traditional method.

Keywords: Machine Learning; Particle Swarm; Gradient Search; Intelligent Search

1 绪论

1.1 研究背景与现状

PSO 算法是 1995 年由 James Kennedy[1]受鸟群觅食行为规律性启发后，提出的一个随机优化搜索的优化算法。由于该算法具有基本思想简单明了、参数设置简单、计算复杂度低等优点，所以经常被广大学者改进并运用在神经网络训练、生物信息学、图像、信号处理、路径规划、函数优化、机器学习等领域。然而，粒子搜索具有一定的盲目性，使得该算法仍然存在算法执行时间过长、收敛精度低，重复搜索等问题。

针对 PSO 算法存在的这些关键性问题，许多学者从不同侧面对 PSO 算法进行改进和优化。其中，最主要的改进方案是对 PSO 算法速度衰减权值进行优化，分为基于线性衰减权重的改进方法与基于非线性衰减权重的改进方法。在基于线性权重的优化方向上，最具有代表性的是一种线性衰减惯性权重的粒子群算法（LDWPSO），由 shi[2]提出。该算法通过惯性系数的逐步缩减的方法，使粒子可以维持较长时间的优越搜索能力，进而使得算法具有良好的算法执行时间与收敛精度。但是，粒子也容易出现过早收敛的问题。在基于非线性权重的优化方向上，虽然部分学者提出的算法可以优化一定的算法执行时间与收敛精度，但缺忽视了粒子的搜索行为具有盲目性、重复性，这使得粒子搜索行为较为混乱，不确定因素较大。因此这类算法在收敛精度的方向还具有很大的优化空间。

为有效提升 PSO 算法的收敛精度，基于梯度加速的粒子群优化算法[19]与基于共轭梯度的粒子群优化算法[20]应运而生。这两种算法的核心是让粒子进行分步计算，先按照线性递减权重的速度进化公式，再按照融合梯度信息的速度进化公式进行迭代计算。虽然这两种算法有效的解决了由粒子搜索的盲目性、不确定性所带来的问题。但是，梯度信息算法执行时间本身就长，这使得算法的执行时间明显被拉长。

区别于上述两种改进策略，本文则在 LDWPSO 算法的基础上直接在速度进化公式中加入梯度下降法，提出一种混合梯度搜索的改进粒子群优化算法，简记为 GIPSO（Gradient Information Based PSO）。GIPSO 算法是基于 LDWPSO 算法，对其飞行速度进化公式进行改良，增添梯度下降法，降低粒子搜索行为的盲目性、不确定性，减少粒子的重复搜索。同时，为了减少 GIPSO 算法的执行时长，在算法的循环迭代计算过程中随机选择半数粒子使用新速度进化公式，另半数粒子使用原有速度进化公式进行迭代计算。通过一系列的测试函数与仿真实验后，证明了本文提出的这种改进 GIPSO 算法具有良好的收敛速度和较高的收敛精度。

1.2 研究目的与意义

PSO算法是一种基于群体智能的随机搜索算法，是根据鸟类群体觅食的行为提出的，常用于寻找最优位置、动态网络路径规划和寻找复杂函数最值点等问题。通常解决这类问题的算法几乎都是由社会性动物群体行为的研究而提出的。著名的有蚁群算法，蜜蜂算法。在PSO算法中，粒子群代表整个鸟群，一个粒子代表一只鸟，粒子在搜索空间内相互影响，按照一定的运动规律，在搜索过程中找到问题的最优解，实现对优化问题的快速求解，为实际问题提供高效的解决方案。

首先，优化PSO算法往往从提高算法的收敛速度与全局搜索能力入手。在算法设计中，如何有效地调整参数，使得算法能够快速收敛到全局最优解，并且能够在大规模问题中具备较好的搜索能力，是算法改进的核心。其次，PSO算法还具备一定的鲁棒性与稳定性。在实际应用中，算法往往会受到问题本身特性、初始位置的选择以及参数设置等因素的影响。那么如何使得算法对这些因素具有一定的适应性，保证算法的可靠性与稳定性，是优化算法的另一核心任务。此外，PSO算法的改进还可以从算法的扩展方面入手。优化的PSO算法虽然已经在很多问题中得到成功应用，但仍然存在一些限制和不足之处。如何在特定问题上进行改进，使得算法能够更好地适应复杂问题的求解，是研究者们持续探索的方向。因此通过调整参数设置、完善策略设计和引入新的策略是有效提升PSO算法的收敛精度与算法执行时间的方法。

在实际应用中粒子群优化算法不仅能够针对多维、非线性、非凸优化问题提供有效解决方案。还可以通过优秀的搜索能力，找到全局或局部最优解，从而为解决复杂问题提供支持。同时，算法对初始条件要求不严格，且具备较强的抗干扰能力。即使在粒子具有很强的不确定性，仍能够保持较好的性能，并且能够灵活地适应问题的变化。PSO算法还具备一定的平衡能力，可以权衡好全局与局部搜索能力，能够在搜索过程中充分探索空间，又能够局部搜索以找到更好的最优解。这使得PSO算法在各种优化问题中都能够取得较好的效果。

因此 PSO 算法的研究目的在于提高算法的收敛速度与全局搜索能力、改进算法的抗干扰能力与稳定性，而其意义则在于解决复杂优化问题、具备高效的抗外干扰能力以及全局搜索和局部搜索的平衡能力。随着科技的进步，以及对算法的研究不断深入，相信PSO算法在优化问题、求解等领域将有更加广泛而深入的应用。

2 经典 PSO 算法

2.1 粒子群优化算法(PSO)

PSO 算法的基本思想是将每个粒子代表成社会空间内的一只鸟，其在搜索空间内的位置代表成目标函数的一个解，在公式中对于该解的目标函数值即为粒子的适应值。假定某粒子群中的粒子总个数为 N（N 为正整数），粒子群所在的搜索空间维度为 D（D 为正整数）。粒子的循环迭代计算公式如下：

$$V_i = V_i + c_1 r_1 (P_i - X_i) + c_2 r_2 (P_G - X_i) \tag{1}$$

$$X_i = X_i + V_i \ . \tag{2}$$

在公式（1）、（2）中，V_i 是粒子 i 在空间内的速度信息，X_i 粒子i 在空间内的位置信息；粒子的速度 V_i 与位置 X_i 参照公式 (1)(2) 进行更新；P_i 是粒子 i 在空间内的基期最优解；P_G 是已被标记的全局最优解；c_1 与 c_2 是自我因子与全局因子的权重系数；r_1 和r_2 是 [0,1] 区间的随机值（伪随机数）；i 为第 N 个粒子（N 为非 0 自然数）。传统粒子群优化算法（PSO）在循环迭代计算过程中能够使粒子维持较高水平的全局搜索能力，然而局部搜索能力往往不被重视，因此，粒子往往很难准确的寻找到全局最优解。

2.2 线性衰减权重的粒子群优化算法(LDWPSO)

由 Shi 提出的一种基于线性递减权重的 PSO 算法，简记为 LDWPSO，粒子飞行速度更新公式如下：

$$V_i = wV_i + c_1 r_1 (P_i - X_i) + c_2 r_2 (P_G - X_i) \tag{3}$$

$$w = w_{max} - \frac{it * (w_{max} - w_{min})}{it_{max}} \ . \tag{4}$$

在公式 (3) 、(4) 中，w 是粒子的飞行速度惯性系数，且$w > 0$。w_{max} 是 w 的最大值；w_{min} 即为 w 的最小值；it 为粒子的迭代步数即当前循环次数；it_{max} 是粒子群体总循环计算次数。在 LDWPSO 算法的循环迭代计算过程中，粒子在前期能够保持优秀的搜索速度，也因此具有优秀的全局搜索能力。然而随着循环计算次数的累计增加，惯性权重 w 则会逐渐减小，影响到粒子的搜索行为，使粒子过早的聚拢在某个较小的区域。

由于 LDWPSO 算法是通过调整粒子的惯性权重来使目标函数指挥粒子进行搜索行为进而提高算法的收敛精度的，这也使得该粒子在空间内的搜索行为存在较多的重复搜索。

2.3 控制种群多样性的粒子群算法(ARPSO)

粒子群优化算法本质上是模拟群体性生物在自然界中的觅食、学习等行为。在自然界中寻找食物的行为就等于粒子在搜索空间内寻找最优解的行为。然而在寻找最优解的过程中，往往会出现局部最优解来迷惑粒子的搜索行为。这就要求粒子群优化算法具有良好的平衡能力可以平衡好全局搜索能力与局部搜索能力。这两种能力本身是互相矛盾的。在自然界中群体性生物总是在不断的分开与重聚。2002 年，J. Riget, J.S. Vesterstrom 根据群体性生物的分散与聚集性，提出了一种控制种群多样性的粒子群算法（ARPSO），速度进化公式如下：

$$V_i(t+1) = V_i(t) + dir \cdot [c_1 \cdot rand \cdot \left(P_i(t) - X_i(t)\right) + c_2 \cdot rand \cdot (P_g(t) - X_i(t))] \quad (5)$$

$$dir = \begin{cases} -1, \ Diversity \le d_{low} \\ \ \ 1, \ Diversity \ge d_{high} \end{cases} \quad (6)$$

$$Diversity = \frac{1}{N \cdot |L|} \sum_{i=1}^{N} \sqrt{\sum_{j=1}^{D} (P_{ij} - \overline{P_j})^2} \ . \quad (7)$$

根据定义可知 $|L|$ 为搜索空间的半径，P_{ij} 是第 i 个粒子的第 j 个分量。$\overline{P_j}$ 是所有粒子平均值得第 j 个分量。在粒子进行搜索最优解的过程中，通过 dir 来控制种群的多样性。当 $Diversity \le d_{low}$，赋 dir 为负号，使粒子分散；当 $Diversity \ge d_{high}$；赋 dir 为正号，使粒子聚拢。此种改进策略，使得粒子的局部搜索能力得到了加强；同时，也使粒子可以继续保持着搜索剩余空间的能力。

2.4 基于正态分布衰减权重的粒子群算法(NDPSO)

NDPSO 算法的核心是通过加入正态分布的方法来解决落后的线性衰减机，解决收敛速度慢、收敛精度浅，等一系列问题。该改进策略即保证了算法的执行效率也避免了粒子容易陷入局部干扰项中，设 x 为伪随机数，μ 为正态分布的集中趋势位置，θ 为正态分布的离散程度。[18]正态分布公式如下(8)：

$$f(x) = \frac{1}{\sqrt{2\pi}\theta} e^{-\frac{(x-\mu)^2}{2\theta^2}} \ . \quad (8)$$

NDPSO 算法的粒子位置信息更新公式如下(9):

$$x_{id}(t+1) = x_{id}(t) + \frac{1}{1+e^{-\frac{f(x_{id})}{f(x)}}} v_{id}(t+1) \ . \quad (9)$$

$f(x_{id})$ 是目标函数的适应值，$\overline{f(x)}$ 为现有粒子群适应度的平均值，$x_{id}(t)$ 为粒子的位置信息，$v_{id}(t)$ 为粒子的飞行速度。NDPSO 算法在迭代计算的前期维持着较高的全局搜索能力，当全局搜索能力提升至最大值时通过正态分布函数对其进行衰减，并提升局部搜索能力，直至平衡。该算法保证了算法的执行时间也避开了多个局部最优解的干扰项。但，该算法在单峰函数里算法执行时间过长，精度不高，在干扰项较少的测试函数里该算法还有待提升。

3 混合梯度搜索的优化 PSO 算法(GIPSO)

PSO 算法与 LDWPSO 算法是群体智能的随机搜索算法，粒子在搜索空间中查找最优位置时会不可避免的出现重复搜索行为。如下图 1 所示， X_{opt}是目标函数中的某个全局最优解；P_G 是粒子集群已经搜索到的全局最优解。

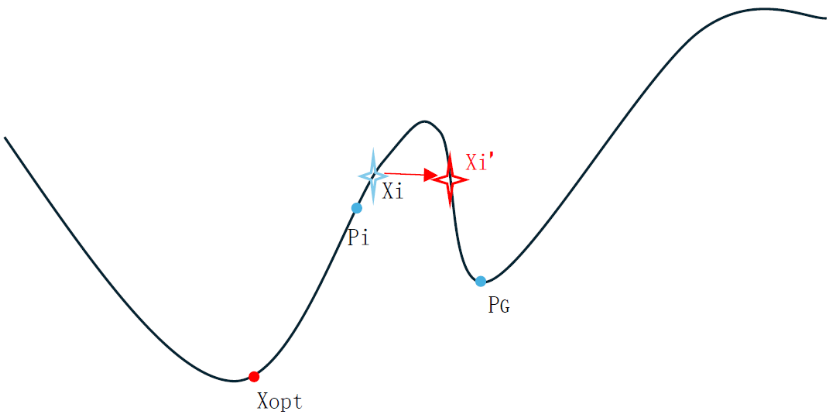


图 1 LDWPSO 算法的重复搜索示例

粒子 X_i 会因为受到已经搜索到的全局最优位置 P_G 的影响而逐渐被 P_G 所吸引，进而错过真正的全局最优位置 X_{opt} 。由上述图 1 可知，为了避免粒子出现错过，只需要让粒子 X_i 沿着目标函数的梯度下降方向，即可到达全局最优位置 X_{opt} 。然而，如果在复杂的目标函数中让粒子完全按照梯度下降方向进行搜索，不仅无法使粒子找到全局最优解，反而会适得其反，使粒子陷入局部最优解。因此，在目标函数中加入梯度信息时，必须要对梯度信息进行控制，以避免粒子因过分追求全局最优而陷入局部最优。

综合上述问题，参考文献 [19] 、文献 [20] ，本文所提出的改进算法的研究思路是直接让目标函数指挥粒子 X_i 沿着梯度的负方向进行搜索行为，达到目标结果。基于此提出一种混合梯度搜索的改进粒子群优化算法，记为 GIPSO。此算法是在 LDWPSO 算

法的基础上，在其速度进化公式中增添梯度信息，并使用权重系数约束梯度信息，防止粒子过分死板的按照梯度负方向进行移动，进而错过真正的全局最优解。GIPSO 算法的研究改进思路分为如下两个步骤：

一、基于 LDWPSO 算法的粒子飞行速度进化公式 (10) ，增添梯度信息：

$$V_i = wV_i + c_1r_1(P_i - X_i) + c_2r_2(P_G - X_i) - c_3r_3G(X_i)\ . \tag{10}$$

在公式 (10) 中，c_3（ c_3 为非 0 自然数）是梯度信息的权重系数。当 $c_3 \gg c_1 + c_2$ 时，公式 (10) 的速度进化公式等同于梯度下降搜索；当 $c_3 \ll c_1 + c_2$ 时，梯度信息对粒子在搜索空间内的运动影响逐渐趋于零，可以忽略不计。由此可知，c_3 也是梯度信息对粒子搜索能力的影响程度的权重系数。r_3 是值域 [0,1] 内的随机因子（这里为伪随机数）。目标函数 $f(x)$ 在 X_i 处的梯度信息为 $G(X_i)$，具体定义为公式 (11) ：

$$G(X_i) = \left(\frac{\eta f}{\eta X_{i,1}}, \frac{\eta f}{\eta X_{i,2}}, \ldots, \frac{\eta f}{\eta X_{i,j}}, \ldots, \frac{\eta f}{\eta X_{i,D}}\right)\ . \tag{11}$$

由于目标函数在 X_i 处不一定可以微分，取 X_i 的差商近似值代替目标函数的梯度值。

二、随机选择半数粒子使用加入梯度信息的公式 (10) 进行迭代计算，剩余其他半数粒子参照原有速度进化公式 (3) 进行迭代计算，同时指挥粒子在空间内搜索全局最优位置。粒子群中的粒子总量大，且需要经过成百上千次的循环计算才能完成搜索，所以采用此方法，不仅不会降低算法的收敛精度，反而会大部分提升算法的执行速度，降低执行时间。

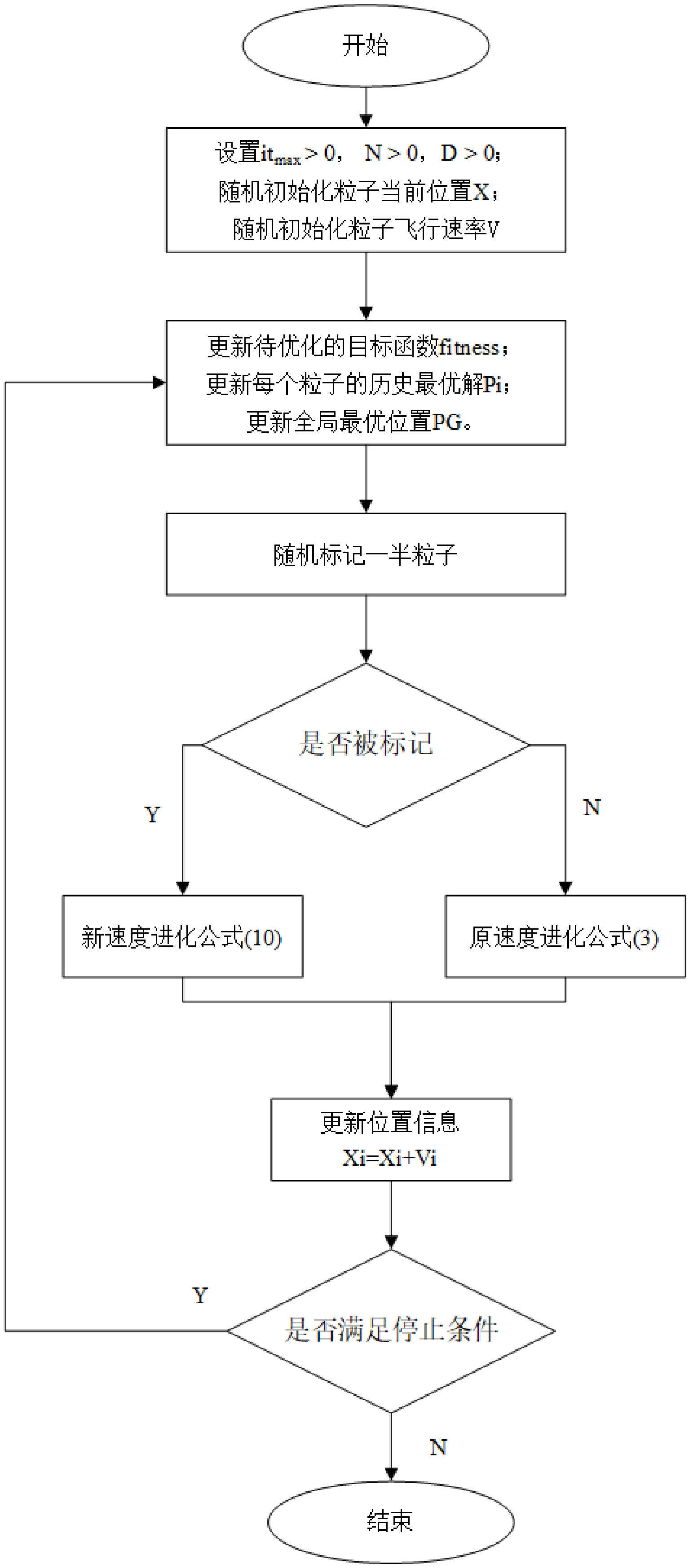
算法核心步骤如下，流程图如图 2。

1）初始化粒子集群。设置最大迭代次数$it_{max} > 0$，种群规模N > 0，目标函数维度 D > 0；随机初始化粒子当前位置与初始运行速度。

2）更新待优化的目标函数 $fitness$，更新每个粒子的历史最优解 P_i，更新全局最优位置 P_G。

3）随机选择半数粒子参照公式 (10) 迭代计算，更新速度;剩余粒子选择公式 (3) 进行计算更迭。所有粒子按照公式 (2) 更新位置。

4）检查是否满足计算停止条件，若满足，则算法停止更新，结束计算;否则转至第 2）步，持续更新计算。


开始
设置it_{max} > 0， N > 0，D > 0；
随机初始化粒子当前位置X；
随机初始化粒子飞行速率V
更新待优化的目标函数fitness；
更新每个粒子的历史最优解Pi；
更新全局最优位置PG。
随机标记一半粒子
是否被标记
Y
N
新速度进化公式(10)
原速度进化公式(3)
更新位置信息
Xi=Xi+Vi
Y
是否满足停止条件
N
结束
图 2 算法执行流程图

4 实验结果与分析

4.1 测试函数介绍

本次实验将分别采用两个单峰函数（Sphere；Rosenbrock），两个多峰函数（Rastrigrin；Griewank）对改进的 PSO 算法进行性能测试。为了更加高效，便捷，直观的了解这四种测试函数的区别，本文将对四种测试算法做出相关介绍，并画出其三维成像图。

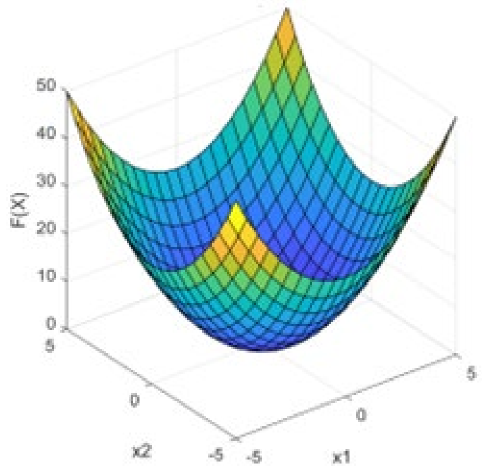


图 3 Sphere(F1)函数三维成像图

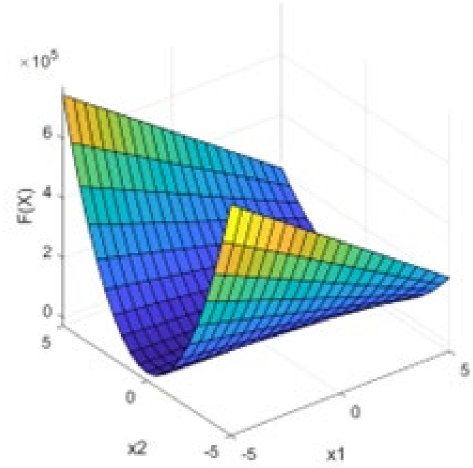


图 4 Rosenbrock (F2) 函数三维成像图

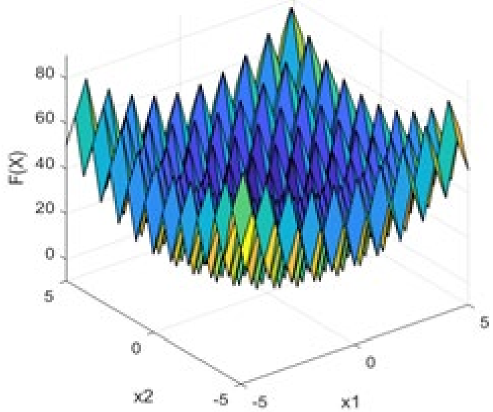


图 5 Rastrigrin (F3) 函数三维成像图

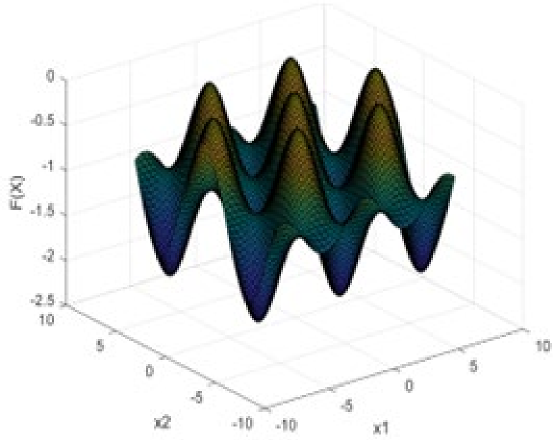


图 6 Griewank (F4) 函数三维成像图

Sphere 函数是单峰函数，是一种最为常见的最小化函数，由于其具有计算速度快，计算步骤简洁，应用广泛等特点，常常被广大学者运用在机器学习算法、对象追踪、人脸识别等领域内，不仅如此，Sphere 函数还因具有较高的维度，也被运用在高纬度的数据集上。

Rosenbrock 函数是单峰函数，是一种用来测试最优算法的性能单峰函数，由 Howard Harry Rosenbrock 在 1960 年提出，观察其图像信息可知，Rosenbrock 函数的每个等高线呈现为圆锥曲线型（抛物线），其全局最小解也在圆锥曲线的最低点。

Rastrigrin 函数是多峰函数，此算法是根据 DeJong 函数改进得来，通过增添余弦调制传递函数来产生大量的局部干扰项（局部最优解），此测试函数，可以广泛的被运用在随机搜索算法中。测试算法的收敛精度。

Griewank 函数也是多峰函数，因其具有很多的极值点，故而存在数个局部最优解与一个全局最优解，因此在 Griewank 函数内是否能够准确的寻找全局最优解，对于改进的粒子群优化算法来说是一项性能测试挑战。

综上， F1 函数的曲面较为陡峭，易于粒子群求解其最优值。F2 函数虽然也是单峰函数，但是其曲面构成一个较为平坦的下降面，寻优难度稍大。F3 和 F4 函数都是多峰函数，在其最优解附近包含大量局部极值点，寻优难度很大。

表 1 测试函数名称及其数学公式

函数名称	表达式	维度	定义域
Sphere(F1)	$\sum_{i=1}^{D} x_i^2$	30	[-100,100]
Rosenbrock(F2)	$\sum_{i=1}^{D-1}[100(x_{i+1}-x_i^2)^2+(1-x_i^2)^2]$	30	[-100,100]
Rastrigrin(F3)	$\sum_{i=1}^{D}[x_i^2-10\cos(2\pi x_i)+10]$	30	[-100,100]
Griewank(F4)	$\frac{1}{4000}\sum_{i=1}^{D} x_i^2-\prod_{i=1}^{D}\cos(\frac{x_i}{\sqrt{i}})+1$	30	[-100,100]

如表 1 所示，D 是测试函数的维度，$x_i \in [-100,100]$，$i = 1,2,\cdots,\mathrm{D}$ 。F1 与 F2 是单峰函数，全局最优解分别在向量 $x = (0，0，\cdots，0)$ 和向量 $x = (1，1，\cdots，1)$ 处取得，最小函数值为 0；F3 和 F4 是多峰函数，在最优解向量 $x = (0，0，\cdots，0)$ 的周围参杂较多干扰项（局部最优解），且干扰项的数量随着维度的增加而增加。

4.2 参数调优设置

在仿真实验中，分别选用一种混合梯度搜索的改进粒子群优化算法 (GIPSO)、一种线性衰减惯性权重的粒子群算法 (LDWPSO)、基于吸引排斥的粒子群算法 (ARPSO) 以及基于正态分布衰减权重的粒子群算法 (NDPSO) 进行循环计算，并客观分析 GIPSO 算法的优缺点及不足。算法的参数设置情况分别如下表 2。

表 2 四种改进算法的参数设置

算法名称	参数设置		
LDWPSO 算法	$c_1 = c_2 = 2.0$	$w_{max} = 0.9$, $w_{min} = 0.4$	
ARPSO 算法	$c_1 = c_2 = 2.0$	$w_{max} = 0.9$, $w_{min} = 0.4$	$D_h = 0.25$, $D_l = 5 \times 10^{-5}$
NDPSO 算法	$\mu = 0$, $\theta = 0.443$		
GIPSO 算法	$c_1 = 0.5$, $c_2 = 2.0$	$c_3 = 0.05(F1, F3, F4)$	$c_3 = 0.05(F2)$

对于测试函数 Sphere(F1)与 Griewank(F4)，算法的迭代计算上限为$it_{max} = 1000$；对于测试函数 Rosenbrock(F2)与 Rastrigrin(F3)，算法迭代计算的上限为$it_{max} = 2000$ 。*D*=30，*N*=30。为了避免实验的不确定性、以及实验结果的偶然性所导致的系统性误差和随机性误差；保证仿真实验结果的严谨性。每个算法将独立运行 20 次，取粒子最优位置的平均数作为每个算法的最终实验结果。

4.3 实验结果&&对比分析

在仿真实验中，通过统计、分析粒子的误差精度，能够更好的反映粒子的收敛精度算法的误差精度数学公式如下公式 (12) ，P_G 为算法的全局最优解。X_{opt} 为全局最优解。

$$error = \left(\left|f(P_G) - f\left(X_{opt}\right)\right|\right) \quad . \tag{12}$$

为了证实 GIPSO 算法的优越性能，实验还分别统计了四个算法的误差精度和算法收敛所需要的迭代次数，图 7 是四种算法分别在四个测试函数中的收敛迭代次数——误差精度的曲线图。

由下图 7 可以得出，LDWPSO 算法在所有测试函数中，算法的执行时间非常短，但是精度并不高，有着过高的误差精度，这也表明 LDWPSO 算法的目标函数在指挥粒子运动时，过早的使粒子聚拢至局部最优位置，从而错过了全局最优解。由于一般的目标函数成千上万次的更迭计算才能得到较好的收敛结果，所以 ARPSO 算法的误差精度不高但是算法的执行时间长。这对于资源的使用和算法的执行时间都是很大的浪费。NDPSO 算法的执行时间和收敛精度都非常的优越，其改进策略是使用正态函数对惯性

权值进行实时调整，使得该算法在 500-600 次的迭代计算之后才开始收敛。但是，NDPSO 算法最终的误差精度仍然存在一定的提高的空间。

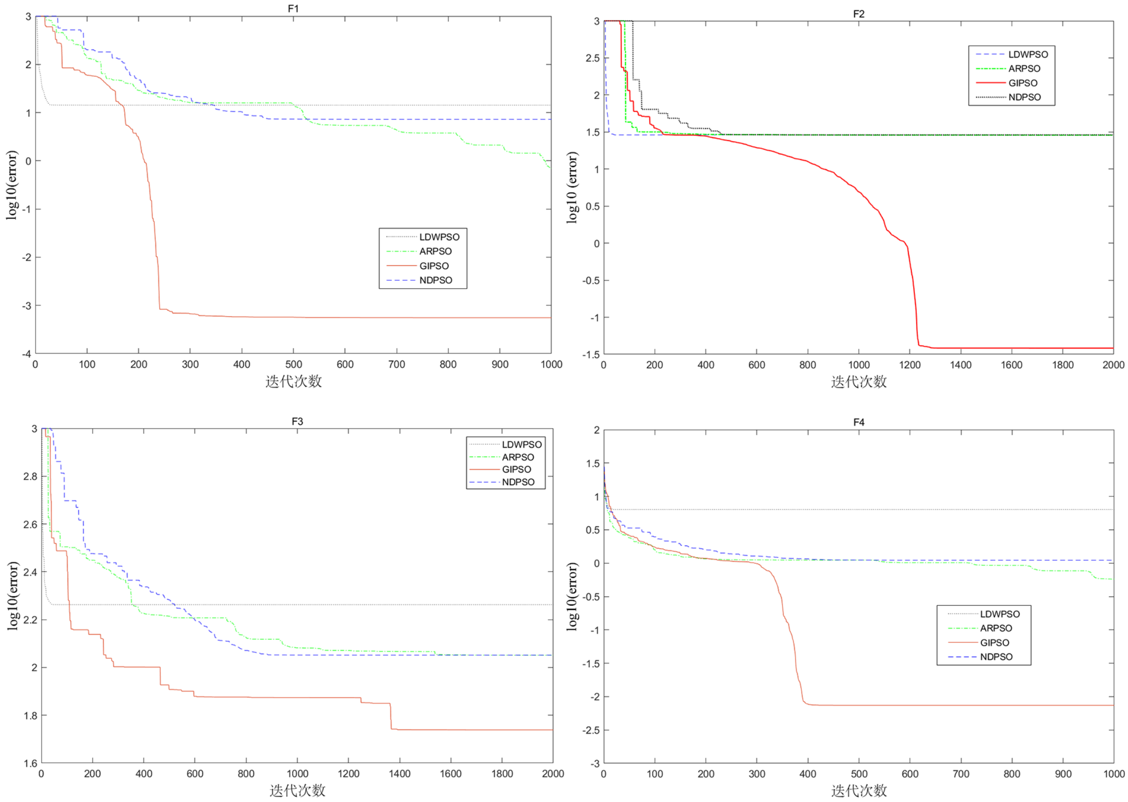


图 7 迭代次数——误差精度曲线图

在 F1 和 F2 函数中进行测试时，GIPSO 算法的展现出了非常优越的性能，仅仅只需要 300-400 次的循环迭代计算即可得到更好的误差精度，误差精度直逼 10^{-3} 至 10^{-2} 是四个改进算法中误差精度最小的。在干扰项较多的 F3 和 F4 测试函数上，GIPSO 算法需要循环计算 1500 次左右才能收敛到最终结果。

综上，GIPSO 算法在指挥粒子进行搜索行为时，需要根据搜索难度的大小，适当调整最大迭代次数 it_{max}，以获得更好的收敛精度以及算法执行时间。

仿真实验所得出算法的误差精度如表 3 所示。

表 3 四种改进算法的误差精度

算法	F1	F2	F3	F4
LDWPSO	8.3×10^{1}	2.8×10^{2}	1.7×10^{2}	1.8×10^{0}
ARPSO	7.6×10^{-2}	2.6×10^{2}	2.2×10^{2}	3.7×10^{0}
NDPSO	4.9×10^{1}	1.8×10^{2}	7.4×10^{1}	5.1×10^{0}
GIPSO	3.2×10^{-4}	9.1×10^{-2}	5.7×10^{1}	2.6×10^{-2}

测试函数 F1 和 F2 是单峰函数，粒子群搜索最优解的过程中无干扰项的影响，较为简单。从表 3 的误差精度数值可以分析得出，LDWPSO 算法的误差精度最大。这主要是因为最早采用的惯性权重递减的策略[2]使得粒子出现早熟现象，导致无法继续搜索最优位置，错过全局最优解。NDPSO 通过采用动态调整惯性系数的方法，避免了粒子群的早熟现象。这使得该算法在 F1 和 F2 上的误差精度低于 LDWPSO 算法。ARPSO 算法通过采用“吸引——排斥”方法，保持了粒子的全局搜索能力，使得在 F1 和 F2 测试函数上的误差精度仅次于 GIPSO。

GIPSO 算法在 F1 和 F2 上的误差精度是这四种改进算法中最小的，达到了 10^{-4} 量级。这表明在使用梯度下降法的目标函数指挥下粒子的搜索行为可以有效提升粒子的局部搜索能力，使粒子的不确定性所带来的弊端得到了很好的解决，不仅避免了粒子过早收敛的现象，还降低了误差精度，减少了算法循环计算次数。

测试函数 F3 是多峰函数，全局最优解参杂在很多局部最优解中，干扰项较多，这便要求粒子具有更强的搜索能力。从表 3 中可以得出，在测试函数 F3 上，LDWPSO 和 ARPSO 算法的误差精度都为 10^2 量级，而 NDWPSO 和 GIPSO 算法的误差精度在 10^1，且系数最小，显然 GIPSO 算法的误差精度是四个改进算法中最小的，但是 GIPSO 算法所带来寻优结果仍然不是最优的。测试函数 F4 是也是多峰函数，但是局部最优解的干扰项相对于 F3 比较少，降低了整体寻优难度。在 F4 上所呈现的结果可以看出 GIPSO 算法的误差精度是最小的，且数量级为 10^{-2}，明显优于其他改进算法。

综上，本文提出的混合梯度搜索的改进粒子群优化算法 GIPSO，在收敛精度和算法执行时间上均由于 LDWPSO，ARPSO，NDPSO 这三种改进算法。

5 总结与展望

本文主要针对传统 PSO 算法的收敛精度不高，算法执行时间长，重复搜索等问题进行了总结分析。虽然很多学者在不同侧面对粒子群优化算法均做出了优化改进，以提高收敛精度和降低算法执行时间，但是由于粒子的运动具有不确定性，大多数算法并没有完美的平衡好两者之间的取值。

为了平衡好粒子的不确定性运动所导致算法的重复盲目搜索，收敛精度低，执行时间长等问题。本文提出一种混合梯度搜索的粒子群优化算法 (GIPSO)。GIPSO 算法主要从两个侧入点对传统的 PSO 优化算法进行改进。第一，直接采用梯度下降法控制梯度信息，降低重复搜索和局部最优解的影响，以提升算法的收敛精度。第二，随机选择半数

粒子使用新速度进化公式循环迭代计算，另半数选择原速度进化公式进行循环迭代计算，以降低算法的执行时间。

最后通过大量的仿真实验对提出的算法的性能进行了验证。实验结果表明，无论是处于只有一个最优解的单峰函数，还是拥有多个干扰项的多峰函数，GIPSO 算法均展现出了其在线性梯度优化范畴内指挥粒子搜索最优解的优越性能。优于 LDWPSO，ARPSO 以及 NDPSO 算法。

另外，实验所选得测试函数较少，无法更进一步寻找 GIPSO 算法的最优参数设置。因此在实验的后续我会接着寻找其他合适的测试函数对改进的算法进行参数优化，展现出改进算法更优秀的性能。其次，本文所采用的梯度下降法在求解机器学习无约束化问题时，往往存在过拟合问题，为了更好的解决过拟合问题带来的弊端，我将继续深入学习，拓展研究机器学习领域内其他算法，以平衡过拟合问题。PSO 算法经常被用于优化神经网络的连接阈值，用以提升神经网络的性能与分类精度，提高分类器的精准度，目前由于我的学术能力与科研能力有限，在我日后的学术生涯中，我将以此为基点，去深入学习神经网络与 PSO 算法之间的应用。

参考文献：

[1] KENNEDY J, EBERHART R. Particle swarm optimization[C]. In proceedings of the 4th IEEE International Conference on Neural Networks. Perth, Australia: IEEE,1995: 1942-1948.

[2] SHI Y, EBERHART R C. A modified particle swarm optimizer [C]//Proceedings of IEEE Congress on Evolutionary Computation, Piscataway, USA: IEEE, 1998: 69-73.

[3] KORDESTANI J, REZVANIAN A, MEYBODI M. An efficient oscillating inertia weight of particle swarm optimization for tracking optima in dynamic environments[J]. Journal of Experimental and Theoretical Artificial, 2016, 28(1-2)137-149.

[4] KONSTANTINOS E, Michael N. On the Computation of All Global Minimizers Through Particle Swarm Optimization[J]. IEEE Trans, on Evolutionary Computation, 2004, 8(3): 211-224.

[5] BERGH F. A Cooperative approach to particle swarm optimization[J]. IEEE Trans, on Evolutionary Computation, 2004, 8(3): 22529.

[6] 李超. 粒子群优化算法改进策略及其应用研究[D]. 江南大学, 2021.

[7] 朱淑芳. 粒子群优化算法的改进及其应用分析[D]. 重庆大学, 2020.

[8] 张庆科. 粒子群优化算法及差分进行算法研究[D]. 山东大学, 2017.

[9] 崔云浩. 基于 CNN-PSO-LSTM 组合模型的生鲜蔬菜销量预测研究[D]. 安徽农业大学, 2022.

[10] 杨佳, 刘海员. 基于 PSO-BP 神经网络的固有无序蛋白质预测[J]. 南开大学学报(自然科学版), 2023, 56(01): 1-7.

[11] 王舜尧. 基于 PSO-GA 混合优化的传送带工件图像识别及其机械臂夹取仿真研究[D]. 辽宁工程技术大学, 2023.

[12] 党博宇, 李海燕. 基于改进 PSO 算法的移动机器人最优路径规划[J]. 组合机床与自动化加工技术, 2024, (02): 71-74.

[13] 谷贵志, 赵咪, 魏子涵, 等. 基于 PSO- 模糊的 PHEV 转矩分配策略研究[J]. 现代电子技术, 2022, 45(11): 143-148.

[14] 陈晓斌, 郝哲睿, 谢康, 等. 基于 PSO-ML-AdaBoost 模型的级配碎石最优压实参数智能预测研究[J/OL]. 铁道科学与工程学报, 1-14[2024-04-15].

[15] 许志良, 曾德炉, 张运生. 一种结合次梯度的粒子群全局优化算法[J]. 计算机应用研究, 2015, 32(04): 1007-1010.

[16] 刘清, 许汪俊彤, 刘正余. 一种融合梯度信息的粒子群优化算法[J]. 佳木斯大学学报(自然科学版), 2023, 41(01): 16-20.
[17] 黄洋, 鲁海燕, 许凯波, 等. 基于S型函数的自适应粒子群优化算法[J]. 计算机科学, 2019, 46(1): 245-250.
[18] 徐浩天,季伟东,孙小晴,等. 基于正态分布衰减惯性权重的粒子群优化算法[J]. 深圳大学学报理工版, 2020, 37(2): 208-213.
[19] 王俊伟, 汪定伟. 一种带有梯度加速的粒子群算法[J]. 控制与决策, 2004, 19(11): 1298-1304.
[20] 梁昔明, 李德生. 嵌入共轭梯度法的混合粒子群优化算法[J]. 小型微型计算机系统. 2014, 35(4): 835-839.

致谢

都说相逢有期，伯乐难遇。知性有余，知音难觅。大学四年是人生中很重要的阶段，我在此结交到了亲人、恩师。他们对我都有着知遇之恩，是他们在一直不断的激励我前进。

古之学者必有师。师者，所以传道、受业、解惑也。老师在我们的求学生涯中扮演着及其重要的角色。首先，我要感谢我的论文指导老师——刘正余老师，是他在我论文研究方面给予了细致入微的指导和建议，才让我顺利完成这篇论文。第二，我要感谢我的 42 位授课老师，是他们弥补了我知识的匮乏，我也从中学到了更多专业基础知识。其次，我还要感谢我的辅导员，在四年的时光里，是他一直给予我生活上的关心与帮助。最后，我要特别感谢刘清老师，是他让我对学科竞赛有了更深入的认知，也是他一直鼓励我尝试参加比赛。我也在此学到了课本之外的新知识，还拓宽了我的见识。

哀哀父母，生我劬劳。父兮生我，母兮鞠我。拊我畜我，长我育我，顾我复我，出入腹我。人生中最需要感谢的是家人，是父母。虽然他们不懂我在学业上的压力与困难，但是他们永远会给予我最好的条件，尊重我的决定。当然我也不负众望，成为了爷爷奶奶、父母的骄傲，成为了邻居聊天的对象。

相逢一醉是前缘，风雨散、飘然何处。在致谢的最后，我要特别感谢席雨缘学姐和左超学长。大学四年，一路走来，他们不仅是。一路走来，不知不觉已相知相识四年。对我而言早已不仅仅是我的学长学姐，而是哥哥姐姐，是挚友，也是亲人。是他们影响了我的整个大学生活，是他们在我需要帮助的时候给予援助，是他们的影子在激励着我不断的向前奔跑。虽然他们已经毕业，都在正在为了各自为了的目标忙忙碌碌，但是我们依旧会心系彼此。不知来岁牡丹时，再相逢何处。人生海海，祝君有帆有岸！

皖西四载与君同，马入尘埃鹤入笼。行文至此，临文涕零，不知所言。