

带随机变异及感知因子的粒子群优化算法

黄懿¹, 梁放驰¹, 范成礼², 宋占福²

(1.空军工程大学 基础部, 陕西 西安 710051; 2.空军工程大学 防空反导学院, 陕西 西安 710051)

摘要:针对传统粒子群算法(PSO)在求解高维空间中复杂函数时容易发生“早熟”现象,根据粒子在空间中的运动规律和散布特点,提出带随机变异因子和动态感知因子的粒子群优化算法。算法通过引入对邻域具有质疑策略的随机变异因子,促使个体粒子对自身邻域进行探索,降低粒子因过于信赖个体最优和全局最优而发生的“早熟”现象,从而改进速度更新策略;同时,为粒子位置更新引入感知因子,使粒子在同一维度上动态自适应控制自身与其他粒子的空间距离,从而避免陷入局部最优。通过测试函数实验、算法对比分析实验、随机参数影响实验和算法复杂性实验,验证了该算法在求解高维空间中的复杂函数等问题时,具有明显的优越性和鲁棒性。

关键词:粒子群优化算法;随机变异因子;动态感知因子;局部最优;全局最优

中图分类号:TP182 **文献标志码:**A **文章编号:**1000-2758(2023)02-0428-11

粒子群优化算法(PSO)是一种群体性智能搜索算法^[1-3],由Kenney于1995年提出,并因实现容易、精度高、收敛快等优点成为国内外研究的热点。但PSO算法在求解高维多峰问题时容易出现“早熟”现象,不能完全保证求得全局最优^[4]。针对此问题,国内外学者提出了许多改进策略^[5-15],如调整参数、精英选择和混合算法等,或在前人的基础上增加优化因子。文献[7-9]通过动态和自适应控制惯性权重提高算法的搜索和挖掘能力,但其搜索范围不一定能够覆盖整个空间,依然存在出现局部最优的情况。文献[10]提出了一种无惯性的自适应精英变异策略,加快了算法的收敛过程,扩大了种群搜索范围,但在高维问题求解上还需进一步实验验证。文献[11]根据个体与全局最优粒子间的距离分别对惯性系数 ω 、学习因子 c_1 和 c_2 求导,给出了3种参数的确定性变化方向,达到参数自适应控制的目的,但对局部最优分散在整个搜索空间,并且与全局最优相距很远的复杂问题求解能力不强。文献[12]在自适应调整惯性权重的量子粒子群优化算法基础上,对粒子位置进行了周期性变异,提高了全局搜索精度,但全局判据仅作为判断优化结果全局

性的依据,不会扩大算法搜索范围。文献[13]为解决粒子种群多样性和收敛性的矛盾,引入了动态划分多种群重构粒子作为引导因子,对执行阶段的最优个体实施混合变异,对时变概率实施反向学习或邻域扰动策略,提高算法的开发与勘探能力,但频繁使用该策略反而会降低部分函数的求解精度。文献[14]在前人基础上,提出了一种自适应局部搜索启动策略,提高了算法的收敛速度和精度,但其全局搜索能力有待验证。文献[15-16]分别将混沌云和单纯形与基本粒子群算法相结合,以提高算法的全局搜索能力,多用于多模态复杂问题求解,对目标函数求解问题需要进一步验证。文献[17]提出粒子速度或位置小概率随机变异与自适应逃逸策略相结合,但求解高维多峰等复杂问题时,其收敛速度较慢,需要迭代2000次以上,才能求出最优值。针对多模态优化问题,文献[18]通过构建集成代理辅助模型,解决了区间多模态粒子群优化计算代价高昂问题,但对多目标多模态的适用性仍需进一步验证。总之,以上算法的改进大多采取参数选择或参数自适应控制策略,或混合其他算法,针对高维复杂问题的求解能力略显不足,不能从根本上克服“早熟”的

收稿日期:2022-06-22

基金项目:国家自然科学基金(72001214)资助

作者简介:黄懿(1993—),空军工程大学讲师,主要从事最优化理论与算法研究。

通信作者:宋占福(1992—),空军工程大学博士研究生,主要从事防空反导作战运筹研究。e-mail:2011songfzy@sina.com

固有弊端。

基于以上分析,根据粒子群在空间中的搜索和分布特点,通过增加随机变异和感知因子,改进粒子群的速度和位置更新策略,提出带随机变异因子和感知因子的粒子群优化算法(PSORMP),扩大粒子在空间中的搜索范围,有效解决了粒子群因搜索范围不足或粒子过于聚集而陷入局部最优问题。通过典型函数测试、算法对比实验、参数影响分析等仿真实验,验证了PSORMP算法具有很强的快速收敛、全局搜索与精确挖掘能力。

1 基本 PSO 算法

设一个 D 维空间中,由 N 个初始搜索粒子组成一个群落,则第 k 代第 i 个粒子的 D 维向量表示为

$$\mathbf{X}_i^k = (x_{i1}^k, x_{i2}^k, \dots, x_{iD}^k), i = 1, 2, \dots, N \quad (1)$$

第 k 代第 i 个粒子的飞行速度为

$$\mathbf{V}_i^k = (v_{i1}^k, v_{i2}^k, \dots, v_{iD}^k), i = 1, 2, \dots, N \quad (2)$$

截至 k 代,第 i 个粒子经历的最优位置称为个体最优,记为

$$\mathbf{p}_i^k = (p_{i1}^k, p_{i2}^k, \dots, p_{iD}^k) \quad (3)$$

截至 k 代,整个粒子群经历的最优位置称为全局最优,记为

$$\mathbf{p}_g^k = (p_{g1}^k, p_{g2}^k, \dots, p_{gD}^k) \quad (4)$$

由此,基本 PSO 算法粒子的速度和位置更新公式为

$$\begin{aligned} v_{id}^{k+1} &= \omega v_{id}^k + c_1 r_1 (p_{id}^k - x_{id}^k) + c_2 r_2 (p_{gd}^k - x_{id}^k) \\ x_{id}^{k+1} &= x_{id}^k + v_{id}^{k+1} \end{aligned} \quad (5)$$

式中: c_1 和 c_2 为学习因子; r_1 和 r_2 为 $[0, 1]$ 范围内的随机数。

针对惯性系数 ω , 采用非线性递减权重策略

$$\omega = \begin{cases} \omega_{\min} - \frac{(\omega_{\max} - \omega_{\min}) \times (f - f_{\min})}{f_{\text{avg}} - f_{\min}}, & f \leq f_{\text{avg}} \\ \omega_{\max}, & f > f_{\text{avg}} \end{cases} \quad (6)$$

式中: f 表示粒子实时的目标函数值; f_{avg} 和 f_{\min} 分别表示当前粒子群的平均值和最小目标值^[19]。

$$f_{\text{avg}} = \frac{1}{N} \sum_{i=1}^N f(x_i^k), i = 1, 2, \dots, N \quad (7)$$

$$f_{\min} = \min \{f(x_i^k)\}, i = 1, 2, \dots, N \quad (8)$$

由基本 PSO 算法的迭代公式可以发现,粒子的寻优方向由粒子群的自身历史最优位置和全局最优

位置所决定,因此,如果全局最优位置是解空间的局部最优位置,就很容易导致粒子群陷入局部收敛。许多高维空间求解问题都是复杂多峰函数,如果算法跳出局部最优的能力不足,这些峰值很容易吸引粒子群发生“早熟”现象,算法的可靠性和稳定性难以保证。

2 PSORMP 算法

为了使算法更具跳出局部最优能力,根据粒子群的运动特性,提出带随机变异及感知因子的 PSO 算法,改变粒子的速度和位置更新策略,以提高全局搜索与挖掘能力。

2.1 带随机变异因子的速度更新策略

为扩大粒子的搜索范围,避免粒子群受初始种群的限制,在速度更新公式中增加一个基于粒子自身邻域、个体最优和全局最优的随机变异因子,以提高粒子的运动能力,更新的速度公式为

$$\begin{aligned} v_{id}^{k+1} &= \omega_1 v_{id}^k + c_1 r_1 (p_{id}^k - x_{id}^k) + c_2 r_2 (p_{gd}^k - x_{id}^k) + \\ &\quad \omega_2 r_3 (p_{it}^k - x_{id}^k) \end{aligned} \quad (9)$$

式中,第一部分表示粒子的“惯性”部分,反映了粒子维持先前速度的趋势;第二部分和第三部分表示粒子的“认知”和“社会”部分,反映了粒子向自身历史最佳位置和群体历史最佳位置逼近的趋势。第四部分为随机变异部分,是粒子对自身局部空间的探索,增加粒子的动力,从而丰富粒子群的多样性。 r_1, r_2, r_3 为 $[0, 1]$ 的随机数,为简化算法的复杂程度,提高后期的挖掘能力,令 $\omega_1 = \omega_2$, 均采用非线性递减权重策略, p_{it}^k 为随机变异粒子,通过简化复合形法的方式得到,具体操作方法如下。

基于邻域质疑策略,以及个体最优和全局最优,给出一种确定随机变异粒子 p_{it}^k 的方法,具体步骤如下:

首先,在 x_i^k 邻域范围内随机搜索一点 $p_i^{k'}$ 作为随机变异粒子的临时位置,则

$$p_i^{k'} = x_i^k + r_4 x_{\max} \quad (10)$$

式中: r_4 为 $[-0.5, 0.5]$ 的随机数(由函数测试得出,具体见本文 3.3 节); x_{\max} 为最大可行变异区间,这里同自变量的取值范围。

其次,由随机变异粒子 $p_i^{k'}$ 的位置、个体最优粒子 p_i^k 位置和种群最优粒子 p_g^k 位置构成变异父辈,选出负理想点 x^f (负理想点为当前状态下,随机变异

粒子 $p_i^{k'}$ 、个体最优粒子 p_i^k 和种群最优粒子 p_g^k 适应值最差的点)

$$x^F : f(x^F) = \max \{ f(p_i^{k'}), f(p_i^k), f(p_g^k) \} \quad (11)$$

假设负理想点为 $p_i^{k'}$, 求其他两点的中点 x^C

$$x^C = \frac{p_i^k + p_g^k}{2} \quad (12)$$

最后, 进行负理想点映射, 得映射点 x^R

$$x^R = x^C + \alpha(x^C - x^F) \quad (13)$$

式中, α 为负理想点映射系数, 一般取 $\alpha = 1.3 \sim 0.5$ 递减。

此时, 映射点 x^R 即为随机变异点 p_{it}^k 的位置, 为简化计算过程, 当负理想点映射失败时, 不再重新选取随机点进行负理想点映射, 直接令 $p_{it}^k = p_i^{k'}$, 进行迭代寻优运算。

图 1 ~ 2 分别表示在二维空间下的粒子运动状态, 其中, 黄色点代表个体最优粒子, 红色点代表全局最优粒子, 灰色点代表随机变异映射点位置。图 1 中, 当 $p_i^{k'}$ 为负理想点时, 图 1a) 表示随机变异粒子 p_{it}^k 的选取过程, 图 1b) 表示带随机变异因子的粒子运动过程, 序号 ① ~ ⑤ 为随机变异因子后粒子运动轨迹, ⑤ 为粒子最终运动位置, 假如没有随机变异因子, 则粒子运动最终位置为 ④, 由图可以看出, 粒子的运动范围明显扩大。图 2 与此同理。

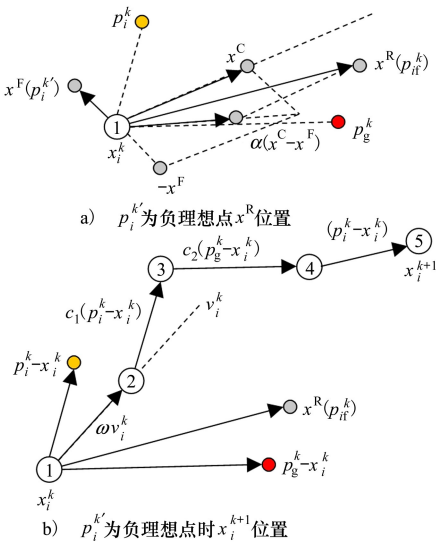


图 1 $p_i^{k'}$ 为负理想点时粒子运动过程

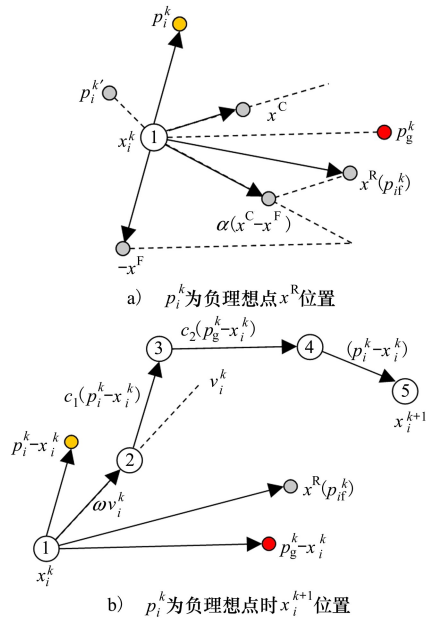
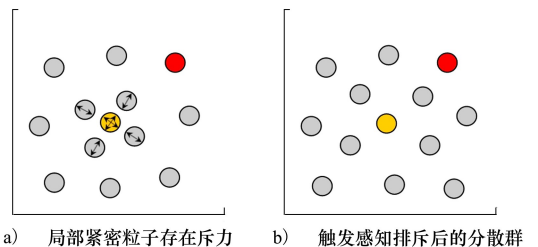


图 2 p_i^k 为负理想点时粒子运动过程

2.2 带感知因子的位置更新策略

针对算法容易陷入局部最优的不足, 本文提出一种带感知因子的位置更新修正策略, 使种群粒子尽可能分散于整个搜索空间, 提升全局搜索能力。其主要思想为: 粒子自身虚拟感知与其他粒子之间的距离, 粒子间的距离小于平均距离的粒子, 该部分粒子自身触发感知排斥, 将自身与其他粒子的距离控制在平均距离之外, 从而保持跳出局部搜索。如图 3 所示, 黄色点代表个体最优粒子, 红色点代表全局最优粒子, 若粒子群出现图 3a) 表示的情况, 个体最优附近粒子群数量比全局最优数量多, 且距离较近, 容易使结果陷入局部最优。此时, 局部紧密粒子存在斥力, 触发感知排斥, 而经过感知因子调整后, 粒子群空间分布如图 3b) 所示, 从而防止陷入局部最优。



a) 局部紧密粒子存在斥力 b) 触发感知排斥后的分散群

图 3 感知因子对粒子群修正示意图

为更好地理解该策略, 引入以下 2 个定义。

定义 1 各维度粒子间的距离定义为

$$d^k = \|x_i^k - x_j^k\|, \forall i, j \quad i \neq j \quad (14)$$

定义 2 则各维度的平均距离定义为

$$\bar{d}^k = \frac{1}{N} \sum_{i \neq j, j=1}^N \|x_i^k - x_j^k\| \quad (15)$$

带感知因子的位置更新思路为:当某一维度粒子间距离 d^k 小于该维度粒子间的平均距离 \bar{d}^k 时,该粒子即刻触发感知策略,即在该维度上增加一定的平均距离 \bar{d}^k ,以跳出局部最优搜索;反之,则不需要触发感知策略。另外,利用惯性权重系数 ω_3 实现对触发感知策略自适应控制,不仅保证了算法初始阶段的大范围搜索能力,也保证了迭代后期小范围的挖掘能力。则带感知因子的位置更新公式为

$$x_{id}^{k+1} = \begin{cases} x_{id}^k + v_{id}^{k+1} + \omega_3 \bar{d}^k, & d^k < \bar{d}^k \\ x_{id}^k + v_{id}^{k+1}, & d^k \geq \bar{d}^k \end{cases} \quad (16)$$

为简化计算过程和保证种群收敛,令惯性权重系数 ω_3 的递减策略与惯性权重系数 ω_1 相同,采取非线性递减的方式,以达到后期的局部挖掘能力。

2.3 PSORMP 算法步骤

根据基本粒子群算法求解过程,结合带随机变异粒子的速度更新策略和带感知因子的位置更新策略,PSORMP 算法的寻优步骤为

step1 输入初始化 PSORMP 算法参数,设置初始种群规模 N ,粒子维数 D ,最大迭代次数 T_{\max} ,学习因子 c_1, c_2 ,粒子速度阈值 v_{\min}, v_{\max} 和粒子各维阈值 x_{\min}, x_{\max} ;

step2 计算并记录粒子群的位置和速度;

step3 计算粒子的适应度值;

step4 计算并更新粒子的个体极值 p_i^k 和全局极值 p_g^k ;

step4.1 利用随机变异粒子的速度更新策略

进行速度更新;

step4.2 利用感知因子的位置更新策略进行位置更新;

step5 计算更新后的粒子群中各粒子的适应度值,并更新个体最优 p_i^k 和全局最优 p_g^k ;

step6 判断是否满足终止条件 $t = T_{\max}$,若满足,则转至 step7,若不满足,则转至 step4;

step7 输出全局最优 p_g^k 和粒子最优 x_i^k 。

3 实验与结果分析

3.1 算法的选择与比较

为验证算法的有效性和优越性,本文选取 5 种算法进行对比,分别是基本 PSO 算法和 4 个基于 PSO 算法的改进算法,即 PSOPC^[20]、RPSO^[21]、NPSO^[22] 和 RFRPSO^[23],其中,①PSOPC 算法:为避免生物群在信息共享时存在自私行为导致形成被动群体,从而无法获得全局最优,提出在粒子群中加入一个被动群模型,提高算法粒子运动活力。②RPSO 算法:排斥 PSO 算法利用在粒子间设置几种排斥机制,使种群粒子从被认为个体最佳的位置移开,从而促使粒子探索空间中的新区域,并在后期切换原有探索模式,达到跳出局部最优的可能。③NPSO 算法:负粒子优化算法的优化策略是在原有粒子群优化算法的基础上,采取将粒子远离个体和群体负理想解的理念,来达到寻优目的。④RFRPSO 算法:带反向预测与斥力因子的 PSO 算法,其优化策略为降低粒子在运动过程中的惰性,引入反向预测因子以改变粒子速度更新方式,为使粒子分散于搜索空间,引入带斥力的位置修正策略。以上算法的速度更新公式如表 1 所示。

表 1 算法速度更新公式比较

算法	速度更新表达式
PSO	$v_{id}^{k+1} = \omega v_{id}^k + c_1 r_1 (p_{id}^k - x_{id}^k) + c_2 r_2 (p_{gd}^k - x_{id}^k)$
PSOPC	$v_{id}^{k+1} = \omega_1 v_{id}^k + c_1 r_1 (p_{id}^k - x_{id}^k) + c_2 r_2 (p_{gd}^k - x_{id}^k) + c_3 r_3 (p_{idrand}^k - x_{id}^k)$
RPSO	$v_{id}^{k+1} = \omega v_{id}^k + c_1 r_1 (p_{id}^k - x_{id}^k) + \omega c_2 r_2 (p_{jd}^k - x_{id}^k) + c_3 r_3 \omega p_{rand}$
NPSO	$v_{id}^{k+1} = \omega v_{id}^k + c_1 r_1 (x_{id}^k - p_{iworst}^k) + c_2 r_2 (x_{id}^k - p_{gworst}^k)$
RFRPSO	$v_{id}^{k+1} = \omega_1 v_{id}^k + c_1 r_1 (p_{id}^k - x_{id}^k) + c_2 r_2 (p_{gd}^k - x_{id}^k) - \omega_2 r_3 (p_{id}^k + p_{gd}^k)$

选择上述算法对比的原因为:①基本 PSO 算法为基准算法,通常用来进行对比测试;②PSOPC 算

法与本文算法类似,都引入了随机粒子,并以随机粒子为依托,扩大粒子群在空间中的寻优范围;

③PSO算法中也存在与本文相同的随机粒子,并且随机粒子对速度影响的大小均由惯性权重控制;④NPSO算法通过一种“反向”机制,使粒子群远离 p_d^k 和 p_g^k ,使粒子探索新区域,与本文算法的动态自适应感知因子的离散化理念类似;⑤RFRPSO算法与本文算法同样是在速度与位置2个方面进行改进,改进策略类似。

3.2 测试函数的选择与参数设置

为验证算法的稳定性和可行性,本文通过求解7个具有代表性的标准基准函数的最小值问题来验证算法,主要包括 Sphere、Quartic、Rosenbrock、Griewank、Ackley、Rastrigrin 和 Schwefel。其中,Sphere 函数除了全局极小值外,还具有 D (维度)个局部极小值,对高维粒子求解困难;Quartic 函数是存在随机干扰的单峰函数,对算法验证极具代表性;Rosenbrock 函数的全局最优位于一个狭窄的抛物线谷中,虽然山谷容易找到,但很难收敛到最小值,是很难求解极小值的典型二次函数;Griewank 函数具

有很多局部极小值,可验证算法跳出局部最优能力;Ackley 函数在二维形式中,呈现出外部平坦,中心下沉的一个深坑状态,并具有众多的局部极小值,对容易产生“早熟”现象的算法求解带来了很大困难;Rastrigrin 函数为复杂多峰函数,在求解过程中很容易陷入全局最优附近的局部极小点;如图 4 所示,Schwefel 函数具有均匀随机性,其局部最优分散在整个搜索空间,并且与全局最优相距很远,欺骗性强,算法必须拥有很强的全局搜索能力才能得到全局最优。实验过程中测试函数的相关信息如表 2 所示。

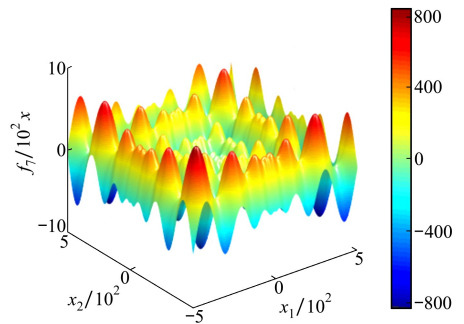


图 4 Schwefel 函数的二维图像

表 2 测试函数及参数设置

函数	名称	表达式	$[x_{\min}, x_{\max}]^D$	\mathbf{X}	f_{opt}	$[v_{\min}, v_{\max}]^D$
f_1	Sphere	$f(x) = \sum_{i=1}^n x_i^2$	$[-100, 100]^D$	$(0, 0, \dots, 0)$	0	$[-50, 50]^D$
f_2	Quartic	$f(x) = \sum_{i=1}^{30} ix_i^4 + \text{random}[0, 1)$	$[-1.28, 1.28]^D$	$(0, 0, \dots, 0)$	0	$[-0.65, 0.65]^D$
f_3	Rosenbrock	$f(x) = \sum_{i=1}^{n-1} [100(x_i^2 - x_{i+1}^2)^2 + (x_i - 1)^2]$	$[-30, 30]^D$	$(0, 0, \dots, 0)$	0	$[-15, 15]^D$
f_4	Griewank	$f(x) = \frac{1}{4000} \sum_{i=2}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]^D$	$(0, 0, \dots, 0)$	0	$[-300, 300]^D$
f_5	Ackley	$f(x) = -20 \times \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=2}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum \cos(2\pi x_i)\right) + 20 + e$	$[-32, 32]^D$	$(0, 0, \dots, 0)$	0	$[-16, 16]^D$
f_6	Rastrigrin	$f(x) = \sum_{i=1}^n [x_i^2 - 10\cos(2\pi x_i) + 10]$	$[-5.12, 5.12]^D$	$(0, 0, \dots, 0)$	0	$[-2, 2]^D$
f_7	Schwefel	$f(x) = -\sum_{i=1}^{30} (x_i \sin(\sqrt{ x_i }))$	$[-500, 500]^D$	$(420.9687, 420.9687, \dots, 420.9687)$	-12 596.5	$[-250, 250]^D$

实验过程中,设置初始粒子群规模 $N = 200$,最大迭代次数 $T_{\max} = 1000$,惯性权重系数 $\omega_{\max} = 0.9$, $\omega_{\min} = 0.5$,维数 $D = 30$,学习因子 $c_1 = c_2 = 2$,可接受

误差为 0.1。实验环境为:AMD Ryzen 7 4800H with Radeon Graphics 2.90 GHz, RAM 16.0GB, Windows10 操作系统, MATLAB2019a。比较算法的参数根据文

献[19-23]的最佳参数而定,如表 3 所示。

表 3 对比算法的参数设置

算法	参数
PSO	$c_1 = c_2 = 2, \omega_{\max} = 0.9, \omega_{\min} = 0.5$
PSOPC	$c_1 = c_2 = 0.5, c_3 = 0.6, \omega_{\max} = 0.9, \omega_{\min} = 0.7$
RPSO	$c_1 = c_2 = c_3 = 1.5, \omega_{\max} = 0.9, \omega_{\min} = 0.5$
NPSO	$c_1 = c_2 = 2$
RFRPSO	$c_1 = c_2 = 2, \omega_{\min} = 0.9, \omega_{\text{end}} = 0.5$

3.3 测试结果及分析

将每个测试函数独立运行 100 次,记录每个算

法的成功率(S ,在规定的可接受误差范围内,成功求解的次数与总运行次数的比值)、平均最优值(B_m ,每种算法对每个测试函数独立运行 100 次后得到的平均最优值,此结果能衡量算法的稳定性)、最终适应值(B_f ,表示每种算法对每个测试函数独立运行 100 次后得到的最优适应值,此结果可以衡量算法的求解精度)、平均运行时间(T_m 每个算法对测试函数独立运行 100 次的平均运行时间)和 Adjusted p-value(P ,表示本文算法与其他算法对比的显著性差异水平),如表 4 ~ 10 所示,除 Adjusted p-value 外,最优值用粗体显示,次优值用斜体显示。各类对比算法对 7 个测试函数的收敛过程如图 5 所示。

表 4 函数 f_1 测试结果对比

算法	S	B_m	B_f	T_m	P
PSO	0.000 0	59.612 4	8.641 2	5.632 3	2.32×10^{-8}
PSOPC	4.70×10^{-1}	16.612 3	0.000 0	39.645 5	1.65×10^{-6}
RPSO	6.30×10^{-1}	14.252 5	18.546 8	11.634 2	4.32×10^{-5}
NPSO	9.80×10^{-1}	1.6515 6	56.146 5	6.181 2	2.21×10^{-6}
RFRPSO	9.50×10^{-1}	2.43×10^{-46}	4.98×10^{-55}	1.05×10^{-45}	1.22×10^{-2}
本文	1.000 0	0.000 0	0.000 0	5.23×10^{-56}	

表 5 函数 f_2 测试结果对比

算法	S	B_m	B_f	T_m	P
PSO	6.80×10^{-1}	1.75×10^{-1}	6.83×10^{-3}	2.34×10^{-2}	2.45×10^{-9}
PSOPC	9.10×10^{-1}	5.35×10^{-2}	1.22×10^{-2}	5.91×10^{-1}	1.24×10^{-7}
RPSO	9.90×10^{-1}	9.81×10^{-3}	4.69×10^{-4}	5.08×10^{-3}	2.54×10^{-5}
NPSO	9.50×10^{-1}	4.46×10^{-2}	1.21×10^{-2}	1.80×10^{-2}	6.25×10^{-1}
RFRPSO	9.60×10^{-1}	7.18×10^{-2}	3.09×10^{-4}	6.31×10^{-5}	8.24×10^{-2}
本文	1.000 0	7.07×10^{-5}	7.14×10^{-7}	2.58×10^{-4}	

表 6 函数 f_3 测试结果对比

算法	S	B_m	B_f	T_m	P
PSO	9.40×10^{-1}	1.19×10^{-5}	1.21×10^{-7}	7.02×10^{-5}	3.21×10^{-9}
PSOPC	8.90×10^{-1}	3.93×10^{-5}	2.09×10^{-8}	2.30×10^{-9}	8.54×10^{-4}
RPSO	9.90×10^{-1}	4.80×10^{-10}	4.93×10^{-32}	1.32×10^{-5}	6.25×10^{-6}
NPSO	9.20×10^{-1}	2.74×10^{-5}	6.61×10^{-8}	4.81×10^{-5}	7.56×10^{-5}
RFRPSO	9.10×10^{-1}	7.03×10^{-6}	1.65×10^{-8}	1.09×10^{-5}	4.23×10^{-1}
本文	9.80×10^{-1}	3.13×10^{-6}	2.18×10^{-23}	3.89×10^{-9}	

表 7 函数 f_4 测试结果对比

算法	S	B_m	B_f	T_m	P
PSO	0.000 0	1.511 2	1.101 5	1.27×10^{-1}	2.35×10^{-12}
PSOPC	9.100 0	1.135 4	0.000 0	3.27×10^{-1}	1.24×10^{-8}
RPSO	3.40×10^{-1}	1.032 4	5.40×10^{-1}	1.39×10^{-1}	8.95×10^{-6}
NPSO	2.100 0	1.172 4	1.073 2	5.79×10^{-2}	2.92×10^{-9}
RFRPSO	9.30×10^{-1}	0.000 0	0.000 0	0.000 0	3.73×10^{-1}
本文	1.000 0	0.000 0	0.000 0	0.000 0	

表 8 函数 f_5 测试结果对比

算法	S	B_m	B_f	T_m	P
PSO	0.000 0	4.102 1	2.253 1	5.34×10^{-1}	3.94×10^{-43}
PSOPC	1.20×10^{-1}	3.593 1	2.043 2	1.262 5	4.36×10^{-37}
RPSO	5.60×10^{-1}	2.443 4	1.574 8	4.17×10^{-1}	2.94×10^{-32}
NPSO	2.20×10^{-1}	3.554 6	2.019 2	5.49×10^{-1}	9.24×10^{-20}
RFRPSO	9.60×10^{-1}	0.000 0	0.000 0	0.000 0	5.29×10^{-2}
本文	9.50×10^{-1}	0.000 0	0.000 0	0.000 0	

表 9 函数 f_6 测试结果对比

算法	S	B_m	B_f	T_m	P
PSO	0.000 0	47.741 2	14.758 6	11.154 8	2.65×10^{-20}
PSOPC	0.000 0	50.346 4	27.712 5	16.736 4	9.46×10^{-18}
RPSO	2.10×10^{-1}	1.71×10^2	1.37×10^2	8.653 2	9.35×10^{-42}
NPSO	5.50×10^{-1}	51.546 5	25.543 6	12.645 1	6.15×10^{-16}
RFRPSO	8.40×10^{-1}	35.216 1	0.000 0	11.932 1	4.34×10^{-6}
本文	1.000 0	0.000 0	0.000 0	0.000 0	

表 10 函数 f_7 测试结果对比

算法	S	B_m	B_f	T_m	P
PSO	0.000 0	-8.60×10^3	-1.06×10^4	8.18×10^2	2.32×10^{-9}
PSOPC	3.30×10^{-1}	-8.49×10^3	-1.04×10^4	7.93×10^2	1.65×10^{-26}
RPSO	0.000 0	-3.79×10^3	-4.62×10^3	1.55×10^3	4.32×10^{-43}
NPSO	6.50×10^{-1}	-8.35×10^3	-1.06×10^4	8.46×10^2	2.21×10^{-41}
RFRPSO	8.50×10^{-1}	-9.87×10^3	-1.16×10^4	7.67×10^2	1.22×10^{-8}
本文	8.60×10^{-1}	-1.18×10^4	-1.26×10^4	2.75×10^{-2}	

表 4~10 的实验结果表明:在求解成功率上, PSORMP 算法取得 5 个最高值, 2 个次高值, 平均成

功率为 97%, 明显高于其他算法; 在平均最优值上, PSORMP 算法得到 6 个平均最优结果和 1 个平均次

优结果,PSO、RPSO、RFRPSO 3 种算法共得到 3 个平均最优结果和 6 个次优平均结果,验证了本文算法的寻优能力;在最终适应值上,PSORMP 算法得到 6 个最优结果和 1 个次优结果,PSO、RPSO、RFRPSO 3 种算法共得到 5 个最优结果和 6 个次优结果,验证了本文算法的求解精度;在平均运行时间上,PSORMP 算法得到 5 个最优结果,2 个次优结果;根据 Adjusted p-value,仅在函数 $f_3 \sim f_5$ 测试结果中,与 RFRPSO 算法的对比结果为 $P > 0.05$,即没有显著差别,其他测试结果显示本文算法显著优于其

他算法,验证了本文算法的优越性和稳定性。从算法的求解精度看,本文算法针对 $f_1, f_4 \sim f_7$ 共 5 个函数求得理想的全局最优,对 f_2, f_3 函数求得最优值与理想值非常接近,并优于其他算法。分析其主要原因在于依托增加随机变异因子和粒子感知因子后,使粒子群在种群空间中的多样性和聚集性达到了合理分配,从而能够有效求解高维复杂多峰函数,特别是对函数 $f_1, f_4 \sim f_7$,取得了理想全局最优解,并且在成功率和稳定性上也优于其他算法,表现了算法较强的鲁棒性。

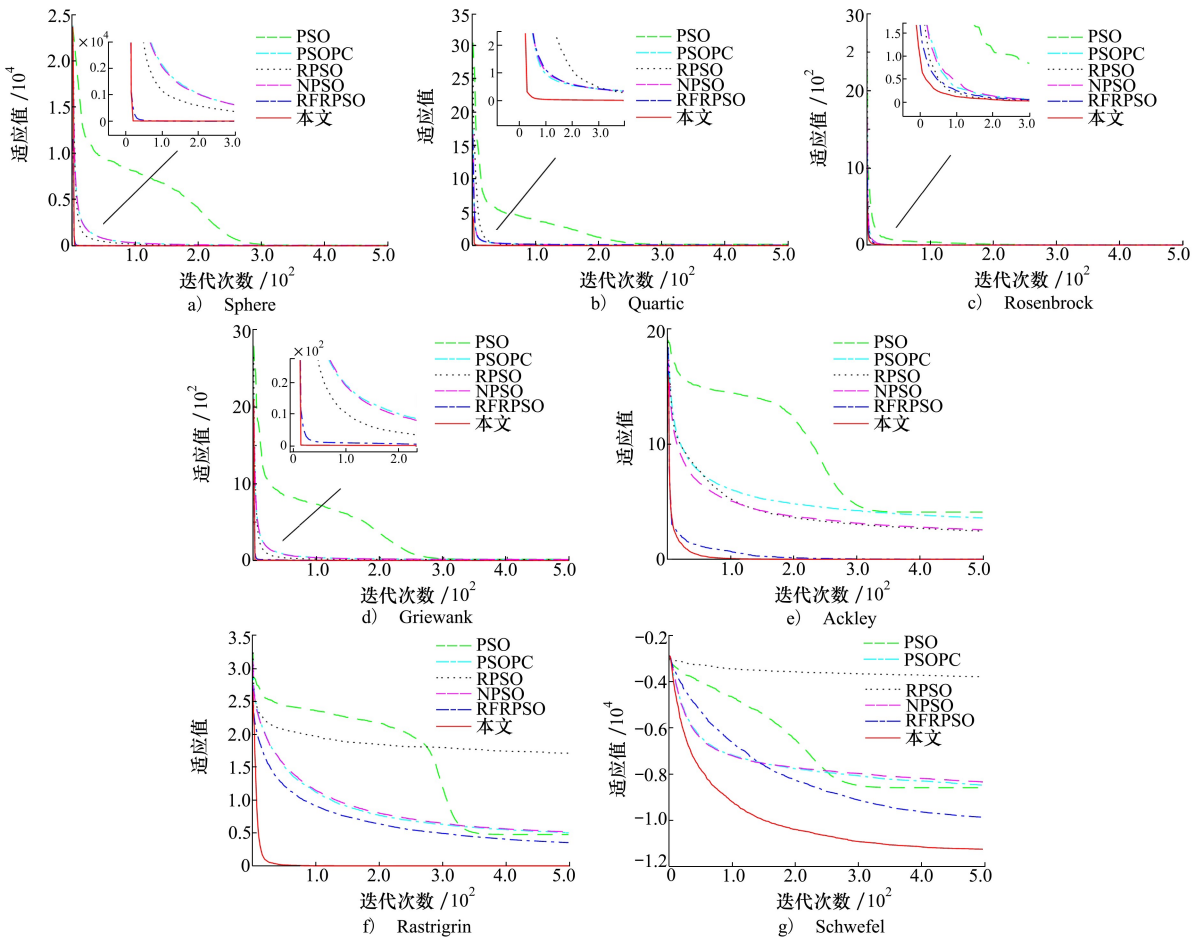


图 5 测试函数收敛曲线图

本文通过增加随机变异因子和感知因子,动态调整了粒子的散布态势,扩大了搜索空间,调整了粒子聚集时间。从函数的收敛图像可以看出,针对函数 $f_1 \sim f_4$,各类算法均求得了最优值,根本原因是函数 f_1 和 f_2 本身就是单峰函数,求解最优值相对容易,而函数 f_3 在维度超过 15 维后,函数特性趋向于单峰,函数 f_4 的全局最优与可达到的局部最优之间有

一道狭窄的山谷,求解最优也相对容易,但本文算法在收敛速度和求解精度上相对于其他算法更有优势;函数 $f_5 \sim f_7$ 为具有大量局部最优的复杂多峰函数,在求解过程中容易陷入局部最优,但本文算法在收敛速度、求解精度上明显优于其他算法,尤其是相对于 PSO、PSOPC、RPSO、NPSO 算法,在迭代 500 次时,本文算法均收敛并取得最优值,而其他算法还未

收敛或未求解全局最优值.由此表明,PSORMP 算法具有很好的跳出局部最优和快速求解能力。

3.4 算法参数的影响分析

PSORMP 算法与基本 PSO 算法的不同之处在于增加了随机变异因子和动态感知因子,其中随机变异因子 $p_{id}^k = x_i^k + r_4 x_{max}$,为测试分析 r_4 的取值对算法寻优的影响,选取 Schwefel 函数作为测试对象,将 r_4 取值划分 $[-0.1,0.1] \sim [-0.9,0.9]$ 共 9 个梯度,其他参数设置和运行环境与本文 3.2 节相同,每个 r_4 取值范围对函数独立运行 50 次。如图 6 所示,给出了 9 个梯度的平均适应度收敛曲线,可以看出,当 r_4 的取值范围由 $[-0.1,0.1]$ 向 $[-0.5,0.5]$ 扩大时,算法的收敛速度和求解精度越来越好,而当 r_4 取值

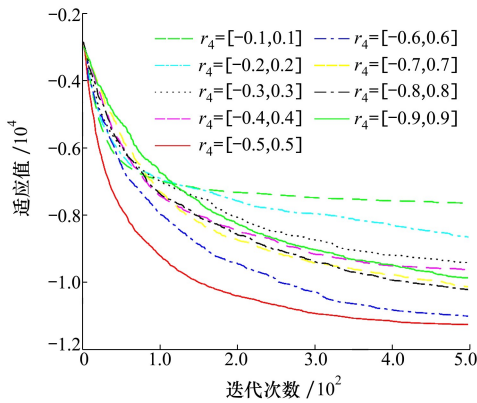


图 6 不同 r_4 取值范围 Schwefel 函数收敛曲线图

范围在 $[-0.6,0.6]$ 之后再扩大,算法的收敛速度变慢,求解精度呈现出不规律的状态。因此,当时 $r_4 = [-0.5,0.5]$,随机变异因子的变异效果最佳。

3.5 算法复杂性分析

本文引入的随机变异因子和感知因子,是在基本粒子群算法的基础上引入的速度和位置更新策略,需进一步分析 PSORMP 算法的计算复杂度,以证明算法的优越性。假设 T 表示最大迭代次数, D 表示维度, N 表示初始粒子群规模。则随机变异因子的复杂度为 $C_{r1}(N) = D \times N \times T$,感知因子的复杂度为 $C_{r2}(N) = N \times T$,基本粒子群算法的复杂度为 $C_{r3}(N) = D \times N \times T$ 。则 PSORMP 算法复杂度为 $C_r(N) = D \times N \times T + N \times T + D \times N \times T \approx D \times N \times T = C_{r3}(N)$ 。所以,PSORMP 算法与基本 PSO 算法的复杂度在理论上属于同一数量级。

为进一步验证上述结论,采取 3.1 节和 3.2 节的参数设置,选用基本 PSO 算法、PSOPC、RPSO、NPSO、RFRPSO 算法及本文算法 PSORMP,对 7 个测试函数在同一初始种群条件下,独立运行 100 次,记录每个函数平均运行时间 T_m ,系统运行环境与 3.2 节相同,最终结果如表 11 所示。其中最优值加粗表示,次优值用斜体表示。通过给出的结果可以看出,PSORMP 算法与其他算法运行时间处于同一数量级,引入的因子没有增加计算的复杂程度。

表 11 算法运行时间对比

算法	f_1	f_2	f_3	f_4	f_5	f_6	f_7
PSO	2.79	3.39	3.18	3.71	3.04	2.85	3.07
PSOC	3.08	6.07	3.95	4.34	4.33	4.07	4.32
RPSO	2.76	3.43	3.76	4.46	4.21	4.09	4.58
NPSO	2.61	6.15	4.01	4.64	4.69	4.16	4.10
RFRPSO	3.21	3.03	4.16	4.28	4.24	4.19	4.82
PSORMP	2.71	3.25	3.60	3.29	4.12	3.46	3.96

4 结 论

本文为解决高维空间中粒子群算容易产生早熟问题,根据粒子群在空间中的运动规律和散布特点,提出了一种带随机变异因子和动态感知因子的粒子群优化算法,改进了粒子的速度和位置更新策略,有

效解决了传统 PSO 算法在求解高维复杂多峰函数时容易陷入局部最优问题,提高了跳出局部最优能力和收敛速度。并通过测试函数和算法对比,验证了算法的有效性和优越性,适合解决工程应用中高维空间中的复杂函数的优化问题。另外,算法是否适用于动态优化求解及可能存在的问题是未来研究的重点。

参考文献:

- [1] KENNEDY J, EBERHART R. Particle swarm optimization[C]//Proceedings of IEEE International Conference on Neural Networks Proceedings, 1995: 1942-1948
- [2] EBERHART R, KENNEDY J. A new optimizer using particle swarm theory[C]//Proceedings of the Sixth International Symposium on Micro Machine and Human Science, 1995: 39-43
- [3] MENDES R, KENNEDY J, NEVES J. The fully informed particle swarm: simpler, maybe better[J]. IEEE Trans on Evolutionary Computation, 2004, 8(3): 204-210
- [4] VAN DEN BERGH F. An analysis of particle swarm optimizers[D]. Hatfield, South Africa: University of Pretoria, 2002
- [5] CHEN Weineng, ZHANG Jun. A novel set based particle swarm optimization method for discrete optimization problems[J]. IEEE Trans on Evolutionary Computation, 2010, 14(2): 278-300
- [6] CLERC M, KENNEDY J. The particle swarm: explosion, stability and convergence in multidimensional complex space[J]. IEEE Trans on Evolutionary Computation, 2002, 6(1): 58-73
- [7] YANG X F, LIU S L. Dynamic adjustment strategies of inertia weight in particle swarm optimization algorithm[J]. International Journal of Control and Automation, 2014, 7(5): 353-364
- [8] 张顶学, 关治洪, 刘新芝. 一种动态改变惯性权重的自适应粒子群算法[J]. 控制与决策, 2008, 23(11): 1254-1257
ZHANG Dingxue, GUAN Zhihong, LIU Xinzhi. Adaptive particle swarm optimization algorithm with dynamically changing inertia weight[J]. Control and Decision, 2008, 23(11): 1254-1257 (in Chinese)
- [9] 黄泽霞, 俞攸红, 黄德才. 惯性权自适应调整的量子粒子群优化算法[J]. 上海交通大学学报, 2012, 46(2): 228-232
HUANG Zexia, YU Youhong, HUANG Decai. Quantum-behaved particle swarm algorithm with self-adapting adjustment of inertia weight[J]. Journal of Shanghai Jiaotong University, 2012, 46(2): 228-232 (in Chinese)
- [10] 康岚兰, 董文永, 宋婉娟, 等. 无惯性自适应精英变异反向粒子群优化算法[J]. 通信学报, 2017, 38(8): 66-78
KANG Lanlan, DONG Wenyong, SONG Wanjuan, et al. Non-inertial opposition-based particle swarm optimization with adaptive elite mutation[J]. Journal on Communications, 2017, 38(8): 66-78 (in Chinese)
- [11] MENG H, TERESA W, WEIR J D. An adaptive particle swarm optimization with multiple adaptive methods[J]. IEEE Trans on Evolutionary Computation, 2013, 17(5): 705-720
- [12] 徐珊珊, 金玉华, 张庆兵. 带全局判据的改进量子粒子群优化算法[J]. 系统工程与电子技术, 2018, 40(9): 2131-2137
XU Shanshan, JIN Yuhua, ZHANG Qingbing. Improved quantum-behaved particle swarm optimization with global criterion[J]. Systems Engineering and Electronics, 2018, 40(9): 2131-2137 (in Chinese)
- [13] 唐可心, 梁晓磊, 周文峰, 等. 具有重组学习和混合变异的动态多种群粒子群优化算法[J]. 控制与决策, 2021, 36(12): 2871-2880
TANG Kexin, LIANG Xiaolei, ZHOU Wenfeng, et al. Dynamic multi-population particle swarm optimization algorithm with recombined learning and hybrid mutation[J]. Control and Decision, 2021, 36(12): 2871-2880 (in Chinese)
- [14] CAO Y L, ZHANG H, Li W F, et al. Comprehensive learning particle swarm optimization algorithm with local search for multimodal functions[J]. IEEE Trans on Evolutionary Computation, 2018, 23(4): 1-15
- [15] LI Mingwei, KANG Haigui, ZHOU Pengfei. Hybrid optimization algorithm based on chaos, cloud and particle swarm optimization algorithm[J]. Journal of Systems Engineering and Electronics, 2013, 24(2): 324-334
- [16] WANG Panpan, SHI Liping, ZHANG Yong. A hybrid simplex search and modified bare-bones particle swarm optimization[J]. Chinese Journal of Electronics, 2013, 22(1): 104-108
- [17] XIA C C, JIANG T T, CHEN W F. Particle swarm optimization of aero dynamic shapes with nonuniform shape parameter-based radial basis function[J]. Journal of Aerospace Engineering, 2017, 30(3): 1-12
- [18] 季新芳, 张勇, 巩敦卫, 等. 异构集成代理辅助的区间多模态粒子群优化算法[J/OL]. (2021-12-02)[2022-08-30]. <http://www.aas.net.cn/cn/article/doi/10.16383/j.aas.c210223>
JI Xinfang, ZHANG Yong, GONG Dunwei, et al. Interval multimodal particle swarm optimization algorithm assisted by heterogeneous ensemble surrogate [J/OL]. (2021-12-02) [2022-08-30]. <http://www.aas.net.cn/cn/article/doi/10.16383/j.aas.c210223> (in Chinese)

- [19] 张岩, 吴水根. MATLAB 优化算法[M]. 北京: 清华大学出版社, 2017
ZHANG Yan, WU Shuigen. MATLAB optimization algorithm[M]. Beijing: Tsinghua University Press, 2017 (in Chinese)
- [20] LEE H, BAEK S W, KIM K W. Inverse radiation analysis using repulsive particle swarm optimization algorithm[J]. International Journal of Heat and Mass Transfer, 2008, 51(11/12): 2772-2783
- [21] HE S, WU Q, WEN J. A particle swarm optimizer with passive congregation[J]. Biosystems, 2004, 78(1/2/3): 135-147
- [22] YANG C M, SIMON D. A new particle swarm optimization technique[C]//Proceedings of 18th international Conference on Systems Engineering, 2005: 164-169
- [23] 范成礼, 邢清华, 李响, 等. 带反向预测及斥力因子的改进粒子群优化算法[J]. 控制与决策, 2015, 30(2): 311-315
FAN Chengli, XING Qinghua, LI Xiang, et al. Particle swarm optimization and variable neighbourhood search algorithm with convergence criterions[J]. Control and Decision, 2015, 30(2): 311-315 (in Chinese)

Improved particle swarm optimization algorithm with random mutation and perception

HUANG Yi¹, LIANG Fangchi¹, FAN Chengli², SONG Zhanfu²

(1.Fundamentals Department, Air Force Engineering University, Xi'an 710051, China;
2.School of Air and Missile Defense, Air Force Engineering University, Xi'an 710051, China)

Abstract: Since traditional particle swarm optimization (PSO) is prone to premature phenomenon when solving complex functions in high-dimensional space, a particle swarm optimization algorithm with random variation and dynamic perception factors in terms of the movement laws and dispersion characteristics of particles in space is proposed. In order to encourage individual particles to explore their own neighborhoods and reduce the premature phenomenon of particles due to over-reliance on individual optimality and global optimality, a random mutation factor with a questioning strategy for neighborhoods is added to the basic algorithm to improve the speed update. At the same time, a perception factor is added to the particle position update, so that the particle can dynamically and adaptively control the spatial distance between itself and other particles in the same dimension, so as to avoid falling into local optimum. The algorithm has obvious superiority and robustness in solving complex functions in high-dimensional space through test function experiments, algorithm comparison analysis experiments, random parameter influence experiments and algorithm complexity experiments.

Keywords: particle swarm optimization algorithm; random variation factor; dynamic perception factor; local optimum; global optimum

引用格式: 黄懿, 梁放驰, 范成礼, 等. 带随机变异及感知因子的粒子群优化算法[J]. 西北工业大学学报, 2023, 41(2): 428-438

HUANG Yi, LIANG Fangchi, FAN Chengli, et al. Improved particle swarm optimization algorithm with random mutation and perception[J]. Journal of Northwestern Polytechnical University, 2023, 41(2): 428-438 (in Chinese)