

DOI: 10.7652/xjtuxb201806008

采用投影螺旋搜索的改进粒子群算法

高嘉乐, 邢清华, 李龙跃, 范成礼
(空军工程大学防空反导学院, 710051, 西安)

摘要: 针对传统粒子群算法在求解高维决策空间问题时容易陷入局部最优的缺点,提出采用投影螺旋搜索的改进粒子群算法。该算法提出了一种基于投影空间的螺旋搜索粒子更新方式,并应用于粒子群算法中以解决早熟问题;为了增强寻优能力,引入混沌策略生成螺旋搜索的参数以提高搜索的随机性;使用自适应算子选择策略分配螺旋搜索更新和传统粒子更新的选择概率,使 2 种粒子更新方式在不同搜索阶段发挥最大效用。仿真实验表明:与基本粒子群算法相比,提出的算法能够以较少的迭代次数收敛,寻优精度最大可提高 10^{-13} ,适合于求解一类具有连续空间复杂多峰值特点的工程应用问题。

关键词: 螺旋搜索;粒子群算法;混沌变异;自适应算子选择

中图分类号: TP183 **文献标志码:** A **文章编号:** 0253-987X(2018)06-0048-07

An Improved Particle Swarm Optimization Algorithm with Projective Spiral Searches

GAO Jiale, XING Qinghua, LI Longyue, FAN Chengli
(Air and Missile Defense College, Air Force Engineering University, Xi'an 710051, China)

Abstract: An improved PSO algorithm with spiral search (SSPSO) is presented to solve the shortcoming that the traditional particle swarm optimization (PSO) algorithm is easy to fall into a local optimum when complex functions in high-dimensional decision space is solved. A new particle update method based on spiral searches in projection space is proposed and applied to PSO to solve the premature convergence problem; Chaos perturbation is introduced to improve the randomness of spiral search so that the search ability is improved. An adaptive selection strategy of operators is also used to balance the usage of spiral search and basic particle update strategy and to maximize their effectiveness in different search stages. Experimental results and a comparison with the traditional PSO indicate that SSPSO converges with fewer iterations and improves the computation accuracy up to 10^{-13} . The algorithm is suitable for solving engineering application problems with characteristics of complexity, multiple peaks and continuous space.

Keywords: spiral search; particle swarm optimization; chaos perturbation; adaptive operator selection

受鸟群捕食的行为启发,Kenney 等于 1995 年提出了粒子群优化(PSO)算法^[1-2],这种算法具有收敛速度快、结构简单等特点,由于它对许多优化问题表现出较高的求解效率而受到国内外广泛的关

注^[3-5]。但是在求解高维空间的复杂问题时,PSO算法容易早熟且陷入局部最优,在多峰值函数的求解问题中表现更为明显。针对这一问题,很多学者对PSO算法进行优化和改进,主要集中在参数调整和其他智能算法相结合两个方面。增加随机搜索范围和频率的方式可以避免PSO算法陷入局部最优,采用的方法如模糊逻辑^[6]、混沌理论^[7]、变化临域粒子辅助更新^[8]等。大部分改进算法将基本PSO算法与全局搜索能力较强的智能算法结合,例如多种群方法^[9]、人工蜂群(ABC)算法^[10]、布谷鸟算法^[11]、采用螺旋样式的涡流搜索(VS)算法等。

目前,螺旋搜索(SS)主要应用于反潜搜索路径规划、磁盘阵列搜索等领域,但在连续函数优化方面应用较少。文献[12]提出了一种螺旋粒子群搜索算法,采用了一种螺旋样式的搜索行为以解决PSO早熟问题,这种方法的参数是随机设置的,并且搜索区域尺寸固定,相当于在一个固定尺度搜索空间中的随机搜索,虽然可以避免陷入局部最优,但效率不高。与螺旋搜索样式相似,文献[13]提出一种涡流样式的算法,以当前种群个体为中心在一个衰减半径的区域内进行随机搜索,尽管搜索范围衰减,但这种搜索方式仍然是一种无目的的搜索。文献[14]提出了量子涡流算法,将涡流搜索行为投影到Bloch球面中,投影搜索行为可以有效地避免在搜索后期大范围的无目的的搜索。

螺旋搜索具有很好的全局搜索能力,但是参数多、后期搜索效率低,因此在连续空间的最优化问题中应用较少。针对上述问题,提出了一种基于投影空间的螺旋搜索的粒子更新算法(SSPSO),并应用于粒子群算法中以解决早熟问题。为了增强算法的寻优能力,引入自适应算子选择方法和混沌变异策略。仿真实验表明,本文算法能够以较少的迭代次数收敛,且具有较高的寻优精度和寻优稳定性。

1 基本粒子群算法

粒子群算法的基本概念源于鸟群捕食行为。在粒子群算法中,每一只鸟看作一个粒子,食物是适应度,头鸟则是每一代的全局最优解,所有粒子向着全局最优解进行收敛。每个粒子的速度和位置更新公式为

$$v_{i,d}^{t+1} = c_0 v_{i,d}^t + c_1 r_1 (p_{i,d} - x_{i,d}^t) + c_2 r_2 (p_d^g - x_{i,d}^t) \quad (1)$$

$$x_{i,d}^{t+1} = x_{i,d}^t + v_{i,d}^{t+1} \quad (2)$$

式中: c_0 为惯性因子; c_1 和 c_2 为加速因子,控制粒子

收敛速度; r_1 和 r_2 是 $[0,1]$ 区间的随机数,保证了种群的多样性; $x_{i,d}^t$ 表示第 t 代第 i 个个体的第 d 维的解; $v_{i,d}^t$ 表示 $x_{i,d}^t$ 对应的速度; $p_{i,d}$ 和 p_d^g 分别表示个体 i 历史最优解的第 d 维和全局最优解的第 d 维。 c_0 、 c_1 和 c_2 这3个参数一般依据经验设置, c_0 在 $(0,1)$ 范围内取值, c_1 和 c_2 通常设为2。Clerc通过大量的仿真实验分析了不同参数对算法寻优性能的影响,建议 c_0 设为0.729, $(c_1 + c_2)/2$ 设为1.494^[15]。

2 基于螺旋搜索的粒子群算法

2.1 螺旋搜索策略

智能算法的搜索方式主要有随机搜索和个体合作两种方法。随机搜索可以分为两种:一种是全局随机搜索,这种方法简单但搜索效率低;另一种是小范围的随机扰动,这种方法是在上一代优秀个体周围进行小范围的随机搜索,搜索效率高且特别适用于求解多峰值问题。个体合作通常用于加快收敛,在优秀个体周围进行搜索容易寻找到更优的个体,但是这种方法也最容易陷入局部最优。受VS算法的启发,结合上述两种搜索策略,本文提出了一种新的螺旋搜索方式。

由于搜索行为是建立在螺旋样式的曲线上,对于螺旋曲线的选择没有固定的要求,本文仅以双曲螺旋线为例

$$\left. \begin{aligned} a &= \frac{\sin w + \theta_1}{w} \\ b &= \frac{-\cos w + \theta_2}{w} \end{aligned} \right\} \quad (3)$$

式中: w 是旋转角度, $w \in (-\infty, 0) \cup (0, +\infty)$; θ_1 和 θ_2 是位置扰动参数, $\theta_1, \theta_2 \in (-1, 1)$ 。当 w 趋近于0时,曲线可以近似看作为一条直线;当 w 趋近于无穷时,曲线则是呈螺旋的样式收敛到 $(0, 0)$ 点。依据上述分析,只要旋转角度不趋近于0,曲线即可呈现螺旋样式。为了便于计算,选取一个整数作为旋转角度的下界,本文设 $w \in (2\pi, +\infty)$,使收束曲线呈螺旋形状。

文献[12]中的螺旋搜索方法采用绝对尺度的螺旋曲线在决策空间搜索,搜索后期依然还有很大的搜索范围,虽然绝对尺度的螺旋搜索具有全局搜索的功能,但搜索到更好的个体难度较大。受文献[14]启发,本文提出了一种相对尺度的搜索,即将搜索投影到一个固定的螺旋曲线的空间中,使得搜索尺度随着搜索过程的进行而变化,效率更高。

2.2 基于螺旋搜索的粒子更新

依据个体合作策略选择两个粒子进行合作,一

个是当前粒子 x_i , 另一个在种群中随机选择粒子 x_j 。然后在螺旋曲线上随机选择一个点 (a_1, b_1) 作为 x_i 的投影点, (a_1, b_1) 称为第一投影点; x_j 投影到 $(0, 0)$, $(0, 0)$ 为第二投影点。假设点 (a_1, b_1) 的旋转角度取值 w_1 , 在螺旋曲线上生成一个点作为新个体 (a_2, b_2) , (a_2, b_2) 的旋转角度 w_2 取值公式为

$$w_2 = w_1 + L\pi \quad (4)$$

式中: L 是角度控制参数。螺旋搜索示意图如图 1 所示。图 1 中 2 个三角点分别是螺旋线的起点和终点, 它们的旋转角度 $w=2\pi$ 和 $w=10\pi$, 依据式(3)计算 2 个点的位置, 在图 1 中标注为三角点。新粒子在螺旋线的投影为 (a_2, b_2) 。当 L 趋近于 $+\infty$ 时, (a_2, b_2) 趋近于 $(0, 0)$, (a_2, b_2) 在决策空间映射的个体与 $(0, 0)$ 对应的 x_j 相似度增高。当 w 趋近于 $+\infty$ 时, 螺旋线上点的变化趋于 0, 此时变异差别极小, 搜索能力可以忽略不计。

为了避免对个体周围进行过度的局部搜索, 定义起始点的旋转角度 w_1 和控制参数 L 的取值范围, $w_1 \in (2\pi, 10\pi)$ 和 $L \in (0, 500)$ 。两个参数的下限依据旋转角度 w 的取值范围定义。 w_1 主要用于确定第一投影点的位置, 是螺旋搜索行为的起点。在实验过程中发现, w_1 对于搜索过程影响不大, 因此定义 w_1 是 $(2\pi, 10\pi)$ 范围内的随机数。随机生成的目的是为了保证每次搜索行为的随机性, 最大限度地避免重复个体的生成。 L 是螺旋搜索的主要参数, L 过大偏重于对第二投影点的周围进行搜索, L 过小偏重全局搜索。在实际的参数测试过程中, L 的上界较小时对于搜索影响不大, 但是 L 超过 500 时, 算法的搜索效率下降, 因此将其上限设为 500。

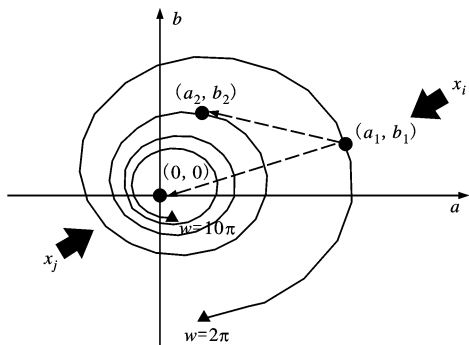


图 1 螺旋搜索示意图

螺旋搜索每次只选择粒子的一个维度, 即将粒子的一个维度投影到螺旋线 (a_1, b_1) 上, 得到新生成粒子的两个候选解

$$\left. \begin{aligned} v_{1,d} &= \frac{((\sin w_{2,d} + \theta_1)/w_{2,d} - a_1)}{0 - a_1} (x'_{j,d} - x'_{i,d}) \\ v_{2,d} &= \frac{((- \cos w_{2,d} + \theta_2)/w_{2,d} - b_1)}{0 - b_1} (x'_{j,d} - x'_{i,d}) \end{aligned} \right\} \quad (5)$$

由式(3)可以看出, 当 w 取值趋于 $\pi/2$ 的整数倍时, a 或 b 趋近于 0。在求解式(5)之后, 两个候选解可能超出决策空间的范围, 为了避免此类现象发生, 选取两个候选解中速度 v 较小的解作为粒子的下一代解的速度, 速度和粒子位置更新如下式

$$v_{i,d}^{t+1} = \begin{cases} v_{1,d}, & |v_{1,d}| < |v_{2,d}| \\ v_{2,d}, & |v_{1,d}| \geq |v_{2,d}| \end{cases} \quad (6)$$

$$x_{i,d}^{t+1} = v_{i,d}^{t+1} + x_{i,d}^t \quad (7)$$

从螺旋搜索粒子更新方式可以看出, 螺旋搜索的全局寻优策略的优势主要有不同维度变异无相关性和反向搜索能力两方面。对于不同维度变异无相关性来说, 每一维度搜索的结果都是不同的, 虽然新粒子需要两个父代个体的信息才能产生, 但新粒子的速度方向与两个父代个体的连线方向没有直接关系, 两个方向呈现正交关系都是可能的。对于反向搜索能力来说, 如果新粒子的投影点 (a_2, b_2) 与父代个体 (a_1, b_1) 在对角象限上, 由式(5)可知, 新粒子的速度方向为负, 即在决策空间上的位置与 x_j 的方向相反, 而速度小于 2 倍的 x_i 与 x_j 之间的距离。

2.3 基于混沌扰动策略的参数设置

完全随机搜索是一种相对低效的但是简单的全局搜索行为, 而小概率的扰动可相对高效地提高算法的寻优能力。本文为螺旋搜索行为设置了多个随机参数, 这些参数通过扰动促使搜索行为不重复。本文的随机扰动参数主要有 4 个: 第一投影点位置参数 w_1 , 曲线扰动参数 θ_1 和 θ_2 , 以及控制参数 L 。

为了提高搜索效率, 将混沌思想引入螺旋搜索中, 选择经典的 Logistic 映射^[16], 计算公式为

$$z^{t+1} = 4z^t(1 - z^t) \quad (8)$$

式中: $z^t \in [0, 1]$ 为混沌变量, 正整数 t 为迭代次数。

利用 Logistic 混沌映射对初始值的敏感点, 给式(8)赋 4 个微小差异的初始值, 即可得到 4 个混沌变量 z'_1, z'_2, z'_3 和 z'_4 , 然后依据下式得到螺旋搜索的 4 个参数

$$\theta_1^{t+1} = 2z_1^{t+1} - 1 \quad (9)$$

$$\theta_2^{t+1} = 2z_2^{t+1} - 1 \quad (10)$$

$$w_1^{t+1} = 10\pi z_3^{t+1} + 2\pi \quad (11)$$

$$L^{t+1} = 500z_4^{t+1} \quad (12)$$

理论上这种经典的 Logistic 映射的值域是一个闭区间。为了避免取到参数的边界值,本文添加参数重新生成机制,即如果参数在混沌变异的过程中产生边界值,那么随机取值重新开始混沌变异。

2.4 自适应算子选择

传统 PSO 个体更新策略具有良好的收敛性,但容易陷入早熟;螺旋搜索具有良好的扰动性,能够提高算法全局寻优能力,但是在收敛速度方面不如传统 PSO 的更新策略。本文提出一种自适应算子选择策略,在搜索阶段的初期以较大的概率选择传统的 PSO 粒子更新策略,提高算法的收敛速度,在后期以较大的概率选择螺旋搜索方式,以避免搜索陷入局部最优。定义算子选择概率

$$P = \frac{7\pi}{10}\arctan\left(\frac{9t}{T} - 4\right) + 0.5 \tag{13}$$

式中: T 表示最大迭代次数。在自适应算子选择策略中,首先产生一个随机数, $r_3 \in (0,1)$ 区间,如果随机数小于 P ,则采用传统 PSO 更新策略,否则使用螺旋搜索。

2.5 算法描述和分析

下面对算法步骤进行详细描述。

输入:加速因子 c_1 和 c_2 、惯性权重因子 c_0 、种群规模 N 、最大迭代次数 T 。

输出:全局最优解 p^g 。

- 步骤 1 初始化种群粒子位置和速度;
- 步骤 2 计算种群的适应度值,并寻找个体历史最优解 p_i 和全局最优解 p^g ;
- 步骤 3 依据 2.4 节中的自适应算子选择策略,生成随机数 r_3 ,如果 $r_3 < P$,转至步骤 4,否则转至步骤 5;
- 步骤 4 使用式(3)和式(4)更新粒子速度与位置,转至步骤 7;
- 步骤 5 依据 2.3 节混沌扰动策略生成螺旋搜索参数;
- 步骤 6 使用螺旋搜索更新粒子速度与位置;
- 步骤 7 更新个体历史最优解 p_i 和全局最优解 p^g ;
- 步骤 8 迭代终止条件判断,若满足,转至步骤 9,否则转至步骤 3;
- 步骤 9 输出全局最优解 p^g ,算法结束。

3 实验分析

3.1 测试函数及参数设置

从文献[17]选取常用的高维测试函数检验本文算法性能,测试函数的名称、编号、搜索范围以及理论最优解见表 1。其中 f_1 为单峰可分离函数; $f_2 \sim f_5$ 为单峰不可分离函数; $f_6 \sim f_7$ 为多峰不可分离函

数; $f_8 \sim f_{10}$ 为旋转多峰不可分离函数。

将本文提出的 SSPSO 算法与基本 PSO 算法、VS 算法^[13]、量子衍生涡流(QIVS)算法^[14]对比。其中,PSO 的 c_0 设为 0.729,学习因子 c_1 和 c_2 设为 0.747,VS 和 QIVS 的参数采用自适应方法生成,无需设置。在仿真中,种群规模 $N=100$,最大迭代次数 $T=200$,测试函数维度 $n=30$ 。4 种算法在 10 个测试函数中运行 30 次,将每种算法在每个测试函数上运行结果的平均值、标准差和平均运行时间 t_{run} 进行比较。仿真实验运行环境为 64 位 Win7 操作系统,CPU 为 Core i5 3.30 GHz,RAM 为 4 GB。

表 1 测试函数描述

函数	函数名称	搜索范围	最优解
f_1	Sum squares	$(-10,10)$	0
f_2	Dixon	$(-10,10)$	0
f_3	Schwefel 2.22	$(-10,10)$	0
f_4	Schwefel 1.2	$(-100,100)$	0
f_5	RosenBrock	$(-30,30)$	0
f_6	Schwefel 2.26	$(-500,500)$	-12 569.5
f_7	Rastrigin	$(-5.12,5.12)$	0
f_8	Levy	$(-10,10)$	0
f_9	Ackley	$(-32,32)$	0
f_{10}	Griewank	$(-600,600)$	0

3.2 测试和结果分析

4 种算法的对比结果见表 2,其中黑体的数据为对比结果最优值。从表 2 中可以看出,本文提出的算法性能良好,10 个测试函数中,有 8 个测试函数得到结果最好。QIVS 算法比 PSO 算法和 VS 算法的运行结果好,但与 SSPSO 算法相比只在 f_5 和 f_8 上获得了最好结果,且 QIVS 算法的平均值与 SSPSO 算法的结果在同一个数量级。比较 PSO 算法和 VS 算法的平均值,PSO 算法总体上略优于 VS 算法。VS 算法的不定向变异虽然不容易陷入局部最优,但是其搜索效率是最低的,因此在有限的迭代次数中表现较差。

图 2 给出了 4 种算法在迭代过程中的收敛曲线。VS 算法整体表现最差,在测试函数 f_1 、 $f_4 \sim f_{10}$ 都是在 100 次迭代以后才收敛,而且寻优能力相对较弱。PSO 算法具有较好的性能,表现出较好的寻优能力,在 $f_1 \sim f_3$ 、 f_5 和 f_{10} 能够以较少的迭代次数收敛。在收敛速度方面,PSO 算法弱于 QIVS 算法和 SSPSO 算法,在寻优精度方面,PSO 算法在 f_4 、 f_6 、 f_9 表现明显不如 QIVS 算法和 SSPSO 算法。

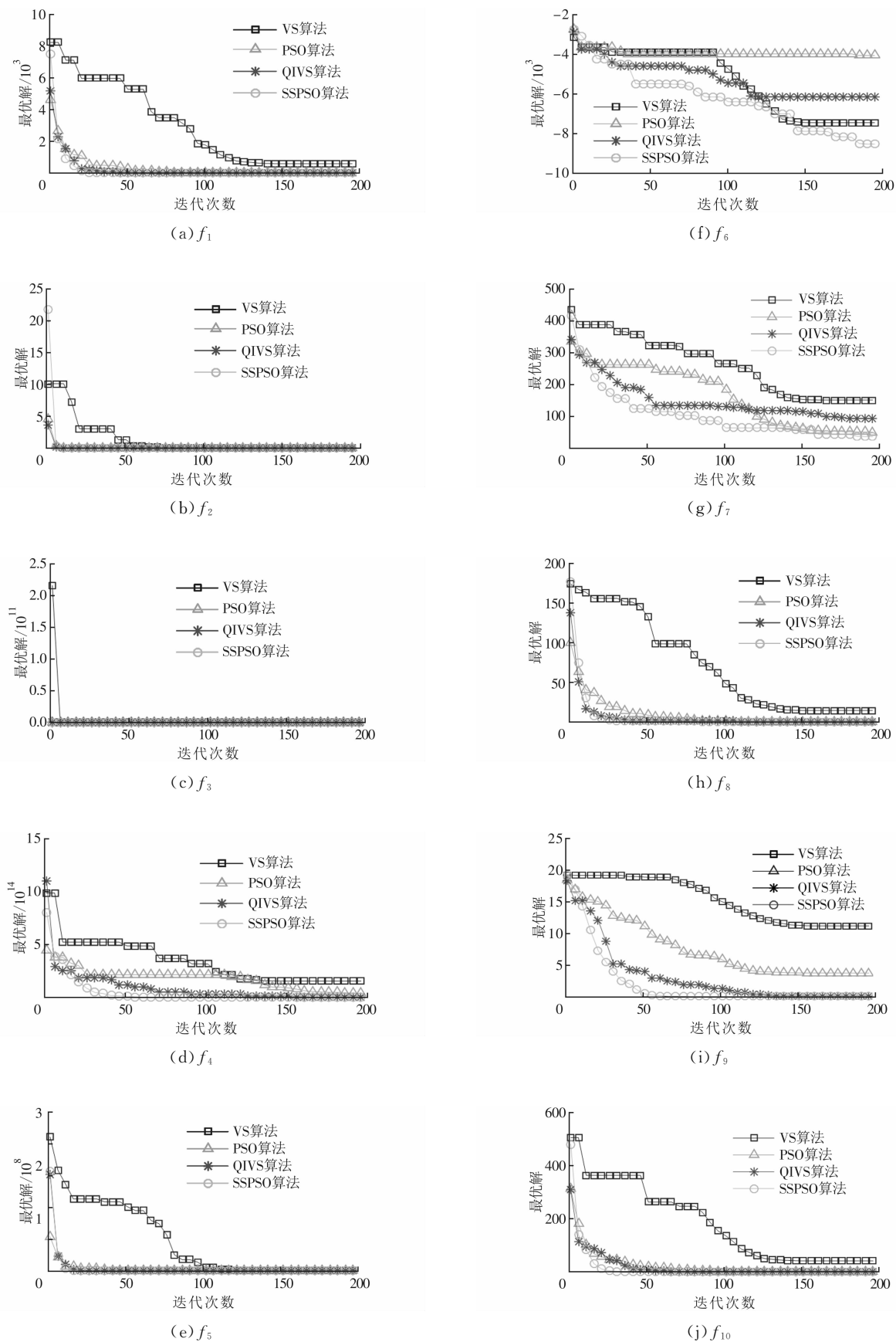


图2 4种算法10种测试函数收敛特性曲线对比

表 2 4 种算法运行结果的平均值和标准差对比

函数	运行结果的平均值				运行结果的标准差			
	VS 算法	PSO 算法	QIVS 算法	SSPSO 算法	VS 算法	PSO 算法	QIVS 算法	SSPSO 算法
f_1	6.07×10^2	4.51×10^{-1}	1.47×10^{-5}	3.57×10^{-14}	8.90×10^1	8.91×10^{-1}	1.11×10^{-5}	1.67×10^{-14}
f_2	2.44×10^{-11}	1.67×10^{-16}	2.22×10^{-12}	7.84×10^{-21}	3.75×10^{-11}	5.29×10^{-16}	3.62×10^{-12}	2.31×10^{-20}
f_3	4.11×10^1	2.86×10^0	2.26×10^{-3}	1.61×10^{-8}	4.38×10^0	1.48×10^0	1.06×10^{-3}	3.32×10^{-9}
f_4	1.50×10^4	2.47×10^3	1.78×10^2	2.40×10^{-2}	2.65×10^3	8.90×10^2	1.13×10^2	1.49×10^{-2}
f_5	1.36×10^6	3.52×10^2	2.80×10^1	2.83×10^1	3.67×10^5	1.20×10^2	6.61×10^{-1}	8.58×10^{-2}
f_6	-7.53×10^3	-4.73×10^3	-6.18×10^3	-8.18×10^3	2.20×10^2	6.63×10^2	3.86×10^2	3.99×10^2
f_7	1.51×10^2	4.16×10^1	7.99×10^1	3.82×10^1	1.23×10^1	1.09×10^1	1.30×10^1	7.00×10^0
f_8	1.35×10^1	6.98×10^{-1}	2.67×10^{-1}	5.27×10^{-1}	3.64×10^0	2.91×10^{-1}	1.13×10^{-1}	1.48×10^{-1}
f_9	1.27×10^1	2.91×10^0	2.23×10^{-3}	9.69×10^{-8}	3.25×10^{-1}	7.95×10^{-1}	8.61×10^{-4}	2.45×10^{-8}
f_{10}	4.27×10^1	7.53×10^{-1}	2.55×10^{-3}	7.14×10^{-13}	7.62×10^0	2.30×10^{-1}	4.87×10^{-3}	3.30×10^{-13}

PSO、QIVS、SSPSO 3 个算法在 f_4 、 f_6 、 f_7 和 f_9 4 个测试函数中寻优性能差异明显,其中 SSPSO 算法能够以较少的迭代次数收敛,且具有明显优于其他 3 种算法的寻优能力。

表 3 给出了 4 种算法在 10 个测试函数运行时的平均运行时间。PSO 算法和 VS 算法的运行时间在同 一个量级,但是 PSO 算法比 VS 算法略好。而本文算法和 QIVS 算法则相对花费较长时间。

表 3 4 种算法平均运行时间对比

函数	t_{run}/s			
	SSPSO 算法	QIVS 算法	VS 算法	PSO 算法
f_1	0.296	0.203	0.047	0.046
f_2	0.297	0.171	0.062	0.032
f_3	0.296	0.203	0.047	0.031
f_4	0.312	0.203	0.062	0.047
f_5	0.078	0.062	0.031	0.015
f_6	0.312	0.203	0.063	0.063
f_7	0.312	0.203	0.062	0.046
f_8	0.094	0.062	0.031	0.016
f_9	0.296	0.188	0.063	0.047
f_{10}	0.297	0.218	0.078	0.047

4 结 论

为解决高维空间中复杂多峰函数的优化问题,本文提出了基于投影空间的螺旋搜索粒子更新方式,并引入自适应算子选择和混沌策略,提出了投影螺旋搜索粒子群算法。通过测试函数的实验分析,本文提出的算法在性能上优于传统 PSO 算法,具有较高的求解精度,能够以较少的迭代次数收敛,适合

于求解具有连续空间复杂多峰值特点的工程应用问题。下一步工作主要采用大量的仿真实验和逆向推导的方法优化算法参数设置以及研究不同的投影点选取对于搜索的影响。

参考文献:

[1] KENNEDY J, EBERHART R C. Particle swarm optimization [C]//Proceedings of the IEEE International Conference on Neural Networks. Piscataway, NJ, USA: IEEE, 1995: 1942-1948.

[2] EBERHART R, KENNEDY J. A new optimizer using particle swarm theory [C]//Proceedings of the International Symposium on Micro Machine and Human Science. Piscataway, NJ, USA: IEEE, 1995: 39-43.

[3] 张卿祎,王兴伟,黄敏. 基于 PSO 和 SA 混合优化的智能容错 QoS 路由机制 [J]. 东北大学学报(自然科学版), 2017, 38(3): 325-330.

ZHANG Qingyi, WANG Xingwei, HUANG Min. An intelligent fault-tolerant QoS routing mechanism based on PSO and SA hybrid optimization [J]. Journal of Northeastern University (Natural Science), 2017, 38 (3): 325-330.

[4] DE SÁ A O, NEDJAH N, DE MOURELLE L M. Distributed efficient localization in swarm robotics using min-max and particle swarm optimization [J]. Expert Systems with Applications, 2016, 50(C): 55-65.

[5] WITEK H A, CHOU C P, MAZUR G, et al. Automated parameterization of the density-functional tight-binding method: II Two-center integrals [J]. Journal of the Chinese Chemical Society, 2016, 63(1): 57-68.

- [6] OLIVAS F, VALDEZ F, CASTILLO O, et al. Dynamic parameter adaptation in particle swarm optimization using interval type-2 fuzzy logic [J]. *Soft Computing*, 2016, 20(3): 1057-1070.
- [7] XU Xiaolong, RONG Hanzhong, TROVATI M, et al. CS-PSO: chaotic particle swarm optimization algorithm for solving combinatorial optimization problems [J]. *Soft Computing*, 2016, 20(22): 1-13.
- [8] SUN Wei, LIN Anping, YU Hongshan, et al. All-dimension neighborhood based particle swarm optimization with randomly selected neighbors [J]. *Information Sciences*, 2017, 405(9): 141-156.
- [9] PORNSING C, SODHI M S, LAMOND B F. Novel self-adaptive particle swarm optimization methods [J]. *Soft Computing*, 2016, 20(9): 3579-3593.
- [10] LI T H S, LIN C J, KUO Pinghuan, et al. Grasping posture control design for a home service robot using an ABC-based adaptive PSO algorithm [J]. *International Journal of Advanced Robotic Systems*, 2016, 13(10): 1-15.
- [11] LI Xiangtao, YIN Minghao. A particle swarm inspired cuckoo search algorithm for real parameter optimization [J]. *Soft Computing*, 2016, 20(4): 1389-1413.
- [12] 滕弘飞, 张英男. 螺旋粒子群优化算法的研究简报 [EB/OL]. (2012/4/10)[2017-06-23]. <http://www.docin.com/p-379958059.html>.
- [13] DOĞAN B, ÖLMEZ T. A new metaheuristic for numerical function optimization: vortex search algorithm [J]. *Information Sciences*, 2015, 293(1): 125-145.
- [14] 李盼池, 卢爱平. 量子衍生涡流搜索算法 [J]. *控制与决策*, 2015, 31(6): 990-996.
- LI Panchi, LU Aiping. Quantum-inspired vortex search algorithm [J]. *Control and Decision*, 2015, 31(6): 990-996.
- [15] CLERC M, KENNEDY J. The particle swarm: explosion, stability and convergence in a multi-dimensional complex space [J]. *IEEE Transactions on Evolutionary Computation*, 2002, 6(2): 58-73.
- [16] LIU Bo, WANG Liang, JIN Yihui, et al. Improved particle swarm optimization combined with chaos [J]. *Chaos, Solitons and Fractals*, 2005, 25(5): 1261-1271.
- [17] 纪震, 廖惠连, 吴青华. 粒子群算法及应用 [M]. 北京: 科学出版社, 2009: 73-113.

(编辑 武红江)

(上接第 47 页)

- [11] FOFACK N C, DEHGHAN M, TOWSLEY D, et al. On the performance of general cache networks [C]// *Proceedings of the 8th International Conference on Performance Evaluation Methodologies and Tools*. New York, USA: ACM, 2014: 106-113.
- [12] DEHGHAN M, MASSOULIE L, TOWSLEY D, et al. A utility optimization approach to network cache design [C]// *Proceedings of the 2016 IEEE INFOCOM*. Piscataway, NJ, USA: IEEE, 2016: 1-9.
- [13] HONIG M L, STEIGLITZ K. Usage-based pricing of packet data generated by a heterogeneous user population [C]// *Proceedings of the 14th Joint Conference of the IEEE Computer and Communication Societies*. Piscataway, NJ, USA: IEEE, 1995: 867-874.
- [14] NEVO A, TURNER J L, WILLIAMS J W, et al. Usage-based pricing and demand for residential broadband [J]. *Econometrica*, 2016, 84(2): 411-443.
- [15] MA R T. Usage-based pricing and competition in congestible network service markets [J]. *IEEE ACM Transactions on Networking*, 2016, 24(5): 3084-3097.
- [16] SHEN H, BASAR T. Optimal nonlinear pricing for a monopolistic network service provider with complete and incomplete information [J]. *IEEE Journal on Selected Areas in Communications*, 2007, 25(6): 1216-1223.
- [17] ZEKRI M, HADJI M, JOUABER B, et al. A Nash Stackelberg approach for network pricing, revenue maximization and vertical handover decision making [C]// *Proceedings of the 36th IEEE Conference on Local Computer Networks*. Piscataway, NJ, USA: IEEE, 2011: 622-629.

(编辑 武红江)