

粒子群优化算法的性能分析和参数选择

王东风¹ 孟丽¹

摘要 惯性权重和加速因子是影响粒子群算法优化性能的重要参数. 基于常用的 12 个测试函数, 本文通过实验研究了不同参数组合下粒子的探索能力和算法的优化性能, 在此基础上推荐了一组固定的参数组合. 通过惯性权重和加速因子的不同变化策略组合对算法性能影响的实验分析, 推荐了一种变化的参数设置方法. 基于 CEC2015 发布的 15 个基准函数进一步验证了本文推荐的参数选取方法的有效性. 最后讨论了粒子群优化 (Particle swarm optimization, PSO) 算法在连续优化和离散优化方面的应用问题.

关键词 粒子群优化, 探索能力, 算法性能, 参数选取

引用格式 王东风, 孟丽. 粒子群优化算法的性能分析和参数选择. 自动化学报, 2016, 42(10): 1552–1561

DOI 10.16383/j.aas.2016.c150774

Performance Analysis and Parameter Selection of PSO Algorithms

WANG Dong-Feng¹ MENG Li¹

Abstract Inertia weight and acceleration factors have significant impact on the performance of particle swarm optimization (PSO) algorithm. Through simulation experiments on twelve classical benchmark functions, this paper studies the algorithm's exploitation ability and optimization performance with different parameters. Based on the experimental results, we recommend a setting for fixed parameters. Furthermore, we study the situation where inertia weight remains unchanged and acceleration factors change with iterations. Then a setting for varying parameters is recommended. The recommended parameters setting methods are verified through 15 benchmark functions that were published in CEC2015. At the end of the paper, a discussion of the PSO application issue on continuous optimization problems and discrete optimization problems is given.

Key words Particle swarm optimization (PSO), exploitation ability, optimization performance, parameter selection

Citation Wang Dong-Feng, Meng Li. Performance analysis and parameter selection of PSO algorithms. *Acta Automatica Sinica*, 2016, 42(10): 1552–1561

粒子群优化 (Particle swarm optimization, PSO) 算法是 Kennedy 和 Eberhart 在 1995 年提出的一种群智能优化算法^[1], 源于对鸟群捕食行为的研究. 1998 年, Shi 等^[2] 在原始 PSO 中引入了惯性权重, 后来被称为标准 PSO. 由于 PSO 结构简单、易于实现, 且不需要借助问题的特征信息, 已受到众多学者的关注, 在算法的性能改进和分析方面不断取得新的成果^[3–4], 在多个领域得到广泛应用^[5].

在 PSO 算法中, 有一些需要调节的参数: 种群规模 (种群的个体数目) N , 速度限值 V_{\max} , 位置限

值 X_{\max} , 惯性权重 w , 加速因子 c_1 和 c_2 . 其中, w , c_1 和 c_2 对算法性能的影响较大, 目前有很多学者对其设定和调节方式进行了研究. 在参数选取方面, 文献 [6–9] 通过实验或理论分析, 推荐了一组固定参数值. 在时变参数的调节方面, 对于惯性权重的调整, 文献 [2, 10] 提出了随迭代次数减小惯性权重, 文献 [11–12] 给出了随机惯性权重策略, 文献 [13–15] 根据种群的信息自适应地调节惯性权重; 对于加速因子的调整, 文献 [16] 提出在迭代的过程中, c_1 和 c_2 线性递减, 文献 [17] 则提出 c_1 线性递减、 c_2 线性递增, 文献 [18] 基于个体的更新信息给出了一种 c_1 和 c_2 的自适应调整策略. 以上这些参数的调整方法, 都在一定程度上提高了算法的性能, 但是, 当涉及到时变参数时, 并没有考虑到惯性权重和加速因子之间的配合作用. 单独依靠调整惯性权重或加速因子, 并不能在种群的局部开发能力 (Exploitation ability) 和全局探索能力 (Exploration ability) 之间进行平衡. 事实上, 各个参数之间需要相互配合, 才能够达到预定的效果. 本文考虑了各个参数之间的配合, 基于在感兴趣的参数范围内进行的优化测试实

收稿日期 2015-11-18 录用日期 2016-03-26
Manuscript received November 18, 2015; accepted March 26, 2016

国家自然科学基金 (61203041), 教育部高等学校博士学科点专项科研基金 (20120036120013), 中央高校基本科研基金 (20140139) 资助
Supported by National Natural Science Foundation of China (61203041), Specialized Research Fund for the Doctoral Program of Higher Education (20120036120013), and the Fundamental Research Funds for the Central Universities (20140139)

本文责任编辑 董海荣
Recommended by Associate Editor DONG Hai-Rong

1. 华北电力大学自动化系 保定 071003
1. Department of Automation, North China Electric Power University, Baoding 071003

验数据, 通过定义最优解超越次数, 推荐了一组固定和时变的参数值.

本文其余部分安排如下: 第 1 节先给出标准粒子群算法的形式; 第 2 节通过仿真实验研究不同参数对粒子探索能力、算法成功率和算法性能的影响, 推荐一组固定参数值; 第 3 节研究算法中的认知参数 c_1 和社会参数 c_2 的变化策略对算法性能的影响, 并推荐一组变参数的组合设定方式; 第 4 节基于 CEC2015 发布的 15 个基准函数进一步验证本文推荐的参数选取方法的有效性; 第 5 节讨论 PSO 算法在连续优化和离散优化方面的应用问题; 第 6 节对全文进行了总结.

1 标准粒子群 (SPSO) 算法

在粒子群算法中, 每个粒子代表寻优空间中一个潜在的解, 有一个由被优化的函数决定的适应值. 在每一次迭代进化中, 粒子通过自身和群体的历史最优位置更新当前的速度和位置. 在任意 $t+1$ 时刻, 粒子群算法中第 i 个粒子第 d 维的速度和位置更新公式为

$$v_{id}(t+1) = wv_{id}(t) + c_1r_{1d}(t)(p_{id}(t) - x_{id}(t)) + c_2r_{2d}(t)(p_{gd}(t) - x_{id}(t)) \quad (1)$$

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

其中, v_{id} 和 x_{id} 分别为粒子的速度和位置, w 为惯性权值, c_1 和 c_2 称为加速因子, 分别为认知参数和社会参数, p_{id} 和 p_{gd} 分别为个体和群体的历史最优位置, r_{1d} 和 r_{2d} 为两个相互独立的服从 $[0, 1]$ 之间均匀分布的随机数, 正是这两个随机数的引入, 使得算法的进化过程具有一定的不确定性, 也因此赋予了算法一定的空间探索能力, 从而有利于找到问题的最优解.

2 粒子群算法的性能分析

下面通过实验说明粒子的探索能力和算法的性能与参数之间的关系. 我们感兴趣的参数区域是 $w \in [-1, 1]$, $c_1 + c_2 \in [0, 8]$, 这也是绝大多数文献研究的区域. 所有的实验都基于以下 12 个常用的基准函数, 由于这些基准函数被普遍应用, 在此只给出函数的名称, 即: Fun 1: Sphere; Fun 2: Rosenbrock; Fun 3: Schwefel's P2.21; Fun 4: Schwefel's P2.22; Fun 5: Schwefel's P1.2; Fun 6: Rastrigin; Fun 7: Griewank; Fun 8: Ackley; Fun 9: Schwefel; Fun 10: Weierstrass; Fun 11: Penalized1; Fun 12: Penalized2.

2.1 参数对粒子探索能力的影响

粒子群算法参数 w, c_1, c_2 的选取对算法的优化性能有很大影响. 不同的参数组合下, 粒子的轨迹形式是不同的, 决定了粒子探索能力的大小. 在种群进化过程中的第 t 代, 若粒子能够找到比 $t-1$ 代的全局最优解更好的解, 那么则认为粒子具有一定的探索能力. 我们通过以下实验观察不同的参数对粒子探索能力的影响. 设种群规模为 N , 为每个粒子设置不同的参数, 设置变量 $Pnum_i$ 记录具有不同参数的粒子 i 在进化的过程中超越上一代的全局最优解的次数:

$$Pnum_i(t) = \begin{cases} Pnum_i(t-1) + 1, & \text{若 } f(p_i(t)) < f(p_g(t-1)) \\ Pnum_i(t-1), & \text{若 } f(p_i(t)) \geq f(p_g(t-1)) \end{cases}$$

其中, $t = 2, 3, \dots, t_{\max}$.

将上面定义的 $Pnum_i$ 称为粒子 i 的第 t 代最优解超越次数, 它表征了粒子探索能力的大小. 显然, 探索能力大的粒子更有可能找到全局最优解.

对于所有的粒子设置相同参数的粒子群算法, 则所有的粒子在统计意义上具有相同的探索能力; 随着算法进化进程的推进, 具有不同的状态的粒子在当前迭代步则具有不同的探索能力; 被设置不同参数的粒子, 一般则具有不同的探索能力. 这已得到作者们进行的大量仿真优化计算实例的证实. 图 1 为 4 组不同的 PSO 种群在优化函数 Sphere 时每个粒子的 $Pnum$ 值: 种群大小均为 50, 函数设置为 10 维, 最大迭代次数为 500, 每组种群运行 50 次, 记录每个粒子 $Pnum$ 值的平均值. 图 1(a) 为种群中粒子设置为相同的参数时 ($w = 0.6, c_1 = c_2 = 1.7$) 每个粒子的 $Pnum$ 值; 图 1(b)、图 1(c) 和图 1(d) 为种群中粒子设置不同参数时每个粒子的 $Pnum$ 值, 其中图 1(b) 中, 第 1~50 个粒子的 w 值均为 0.6, $c_1 = c_2$ 为 0.04 : 0.04 : 2; 图 1(c) 中, 第 1~50 个粒子的 w 值均为 1.7, $c_1 = c_2$ 分别为 0.02 : 0.02 : 1, 图 1(d) 中, 第 1~10 个粒子的 w 值均为 0.2, $c_1 = c_2$ 分别为 0.2 : 0.2 : 2, 第 11~20 个粒子的 w 均为 0.4, $c_1 = c_2$ 分别为 0.2 : 0.2 : 2, 依此类推, 第 41~50 个粒子的 w 均为 1.0, $c_1 = c_2$ 分别为 0.2 : 0.2 : 2.

在 $w \in [-1, 1]$, $c_1 + c_2 \in [0, 8]$ 的参数区域上对所用的 12 个基准函数进行蒙特卡罗实验. 每个函数进行两组实验: 1) 种群中的个体取相同的 $w, c_1 = c_2$ 则取不同的数值. 本组实验中, 根据种群参数 w 的值分为 51 组子实验, w 的取值按 0.04 的间隔从 $-1 \sim 1$, 每个子实验中, 种群大小均为 81, $c_1 = c_2$ 按 0.05 的间隔从 $0 \sim 4$ 取 81 个值分配给不同的粒子, 函数运行 50 次, 记录不同粒子在 50 次运行中的

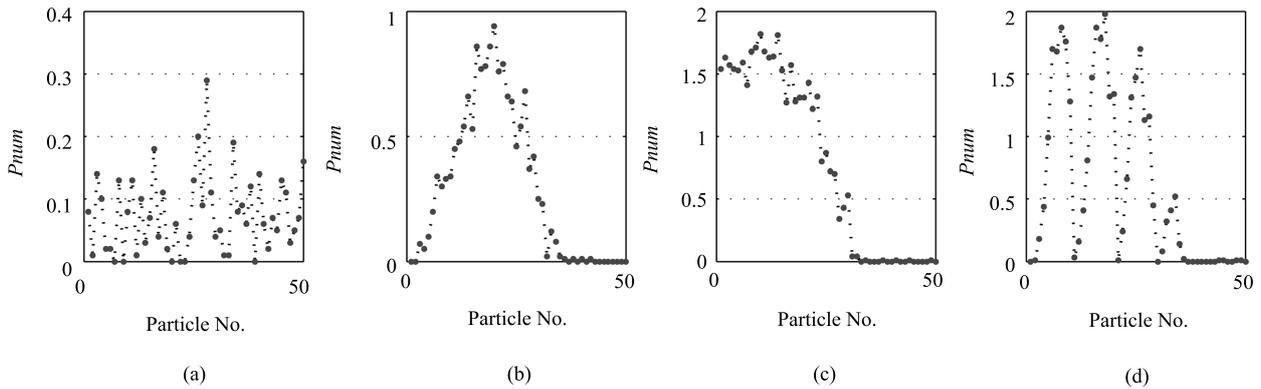


图1 具有不同参数的种群优化函数 Sphere 时粒子的 $Pnum$ 值

Fig. 1 The $Pnum$ value of function Sphere optimized by PSO with different population scale

$Pnum$ 的平均值. 2) 种群中的个体取不同的 w , 而 $c_1 = c_2$ 取相同的固定值. 本组实验中, 根据种群参数 c_1 (c_2) 的值分为 81 组子实验, $c_1 = c_2$ 的取值按 0.05 的间隔从 0~4, 每个子实验中, 种群大小均为 51, w 按 0.04 的间隔从 -1~1 共取 51 个值分配给不同的粒子, 函数运行 50 次, 记录不同粒子在 50 次运行中的 $Pnum$ 的平均值. 将两组实验中得到的相同参数组合下粒子 i 的 $Pnum_i$ 进行求和, 得到此参数组合在不同种群环境中的综合探索性能, 进而得到整个参数区域内不同参数组合的综合超越次数. 相比于超越次数的绝对值, 我们更关心不同参数组合下粒子探索能力的相对关系, 因此对每个函数得到的结果分别进行归一化处理, 结果如图 2 所示. 值得注意的是, 在 $6 < c_1 + c_2 \leq 8$ 的区域中, 对于任意的 w 运行结果 $Pnum_i$ 均为 0, 因此, $6 < c_1 + c_2 \leq 8$ 的区域并未在图中进行展示. 从图 2 可以明显地看到粒子群算法的参数取值对不同基准函数的优化性能的共性和特性, 即对每个基准函数来说, 算法的探索能力对参数的取值表现出明显的区域性, 总体来说大同小异, 这为 PSO 的参数选择提供了很好的指导作用.

2.2 参数对算法成功率的影响

在 $w \in [-1, 1]$, $c_1 + c_2 \in [0, 8]$ 矩形域上按照等间隔的网格取点, ($c_1 + c_2$) 轴取点间隔为 0.02, 共 101 点, w 轴取点间隔为 0.04, 共 201 点, 共可得到 20301 组不同的参数组合. 对每一组参数组合进行蒙特卡罗仿真实验, 优化 12 个基准函数. 种群规模 N 为 50, 最大迭代次数 t_{max} 为 500, 每个函数独立运行 50 次. 为每个函数设置寻优精度 goal, 记录不同参数组合下达到所设置寻优精度的成功率, 绘制成灰度图像. 每个函数的寻优精度为图中的 goal 所示, 是通过实验测试得到的, 原则是不会使测试结果得到的最佳参数组合区域过大而失去指导意义, 也

不会使其过小而失去一般性参考价值. 设置一定的寻优精度, 实验结果如图 3 所示.

由图 3 可以看到, 除了 Fun 2、Fun 6、Fun 7 和 Fun 9, 其他 8 个函数的最优参数区域在形状上非常相似, 并且都出现在靠近过渡区域的边界部分. 这一部分的参数使得粒子轨迹能够以一定的幅值振荡, 使得探索能力和开发能力获得很好的平衡. 对于 Fun 2、Fun 6、Fun 7 和 Fun 9 这 4 个函数, Fun 2 为很难极小化的典型病态二次函数; Fun 6、Fun 7 为典型的具有大量局部最优点的复杂多峰函数; Fun 9 的全局最优点与第 2 最优点相距很远, 算法往往朝着错误的方向 (第 2 最优点) 收敛. 这 4 个函数只靠调节算法参数很难得到满意的解, 运行结果的规律性没有其他函数那么明显.

将图 2 与图 3 对比可以看到, 图 2 和图 3 并不是完全对应的. 图 2 中展示的粒子的探索能力为粒子的局部探索能力和全局探索能力的总和, 而不同的函数具有不同的特性, 对群体的局部探索能力和全局探索能力的要求是不同的. 虽然总的探索能力与寻优性能并不是完全对应的, 但是, 没有探索能力的参数区域, 基本上是不可取的.

2.3 几种典型参数选取策略的性能对比及一组固定参数推荐值

由图 3 可以看到, 使得算法获得优良性能的参数区域因函数的不同而有一些差异, 基本上没有一组参数能够使得所有的函数同时获得最优, 尤其可以看到, Fun 2 和 Fun 9 的较优参数区域的交集部分是非常小的. 尽管如此, 仍然可以选择出一些参数, 使算法在所有函数上获得比较令人满意的综合性能. 为了能够更加容易地选择出在 12 个函数上综合性能较优的参数, 我们将不同函数下得到的运行结果进行整合, 得到算法在不同参数下的综合性能的示意图, 见图 4.

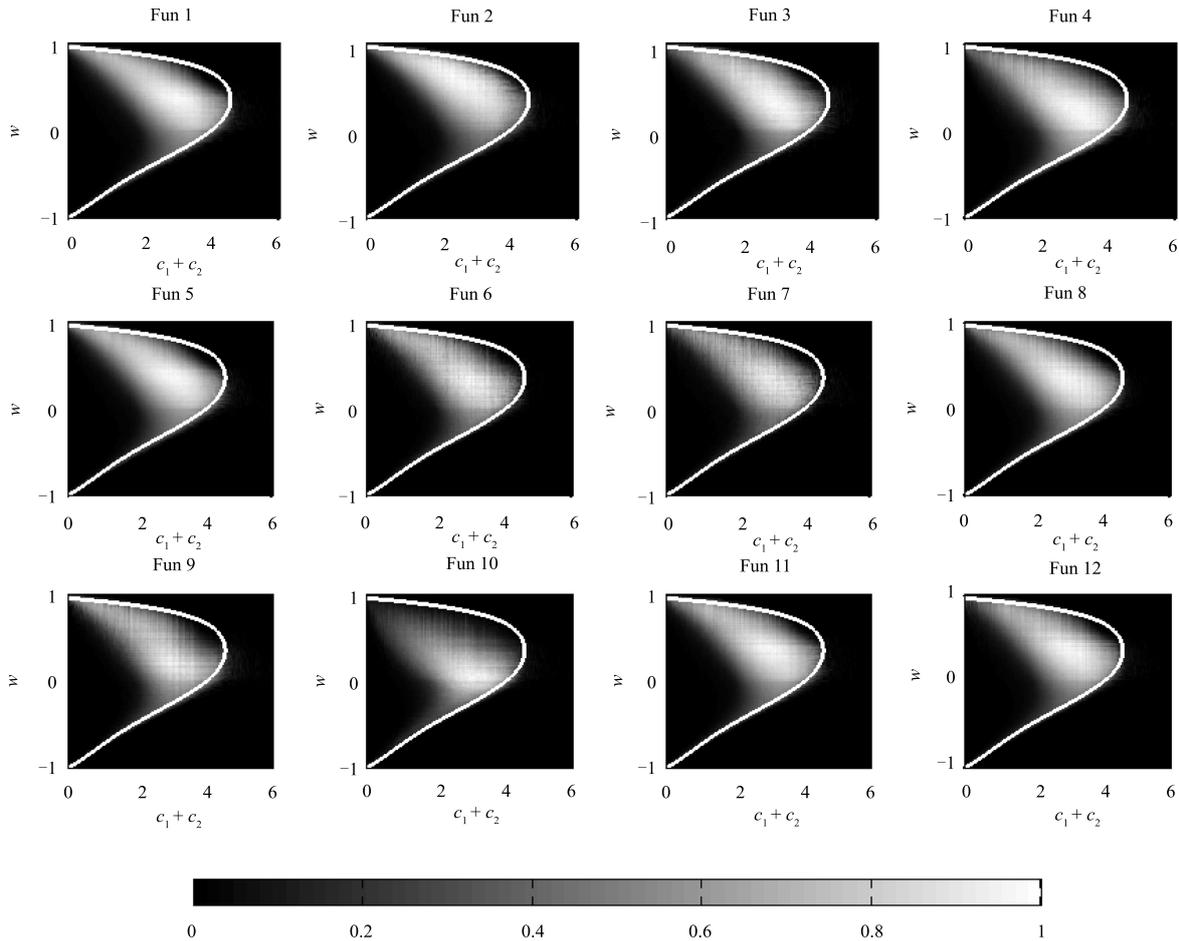


图 2 粒子探索能力与参数的关系

Fig. 2 Relationship between particle exploration ability and its parameters

图 4 中标示的 PARA1 和 PARA2 分别为文献 [6] 和文献 [7] 推荐的参数取值. 其中, 参数组合 PARA1 ($w = 0.7298$, $c_1 = c_2 = 1.49618$) 是 Eberhart 等^[6] 将带有惯性权重和带有收缩因子的两种形式的粒子群算法进行比较后推荐的一组参数; 参数组合 PARA2 ($w = 0.6$, $c_1 = c_2 = 1.7$) 是 Trelea^[7] 在分析粒子轨迹收敛域的基础上推荐的参数. 参数组合 PARA3 是我们根据性能综合图, 即图 4 推荐的一组参数: $w = 0.4$, $c_1 = c_2 = 2$. 在另外的工作中, Samal 等^[8] 在理论分析的基础上给出了一组具有较快收敛速度的参数: $w = 0.6$, $c_1 r_1 = 0.103$, $c_2 r_2 = 2.897$ (参数组合 PARA4); Carlisle 等^[9] 在文献 [6] 的基础上进行了进一步的实验, 考虑了 $c_1 \neq c_2$ 的情况, 通过仿真实验得到, 当 $c_1 = 2.8$, $c_2 = 1.3$ 时, 算法的性能较优, 即 $w = 0.7298$, $c_1 = 2.0434$, $c_2 = 0.9487$ (参数组合 PARA5). 我们将这 5 组参数在 12 个函数上进行测试, 种群规模、最大迭代次数和每个函数独立运行的次数都与第 2.2 节一致. 记录运行数次结果中的最小

值 (min)、最大值 (max)、平均值 \pm 标准差 (mean \pm std), 以及优化成功率 (Success rate, SR), 并且记录了不同参数在每个函数指定寻优精度下的成功率, 结果列于表 1 中. 为了清晰地显示参数对算法性能的影响, 此处为 12 个函数指定的寻优精度依次为: 10^{-20} , 3, 10^{-5} , 10^{-10} , 10^{-20} , 5, 10^{-1} , 10^{-10} , 500, 10^{-3} , 10^{-20} , 10^{-20} . 以下各测试试验中所用的寻优精度均与此相同. 首先来看参数 PARA1~PARA3 的运行结果, 这三个参数组合基本上都在图 4 中的较优区域里. 三组参数在 Fun 1, Fun 3, Fun 4, Fun 8, Fun 11 和 Fun 12 上的运行结果差别不大. 在 Fun 2, Fun 5, Fun 6, Fun 7, Fun 9 和 Fun 10 上的差别相对较大, 参数 PARA1 在 Fun 2, Fun 7 上表现较优, 在 Fun 5 上表现较差; 参数 PARA2 在 Fun 6, Fun 9 上表现较差; 参数 PARA3 在 Fun 6, Fun 9 和 Fun 10 上表现较优, 在 Fun 2, Fun 7 上表现较差. 正如从图 3 中看到的, 每组参数都有其适应的函数. 再来看参数 PARA4 和参数 PARA5, 同样的, 每组参数在不同函数上表现出的性能也是有差别的. 参数

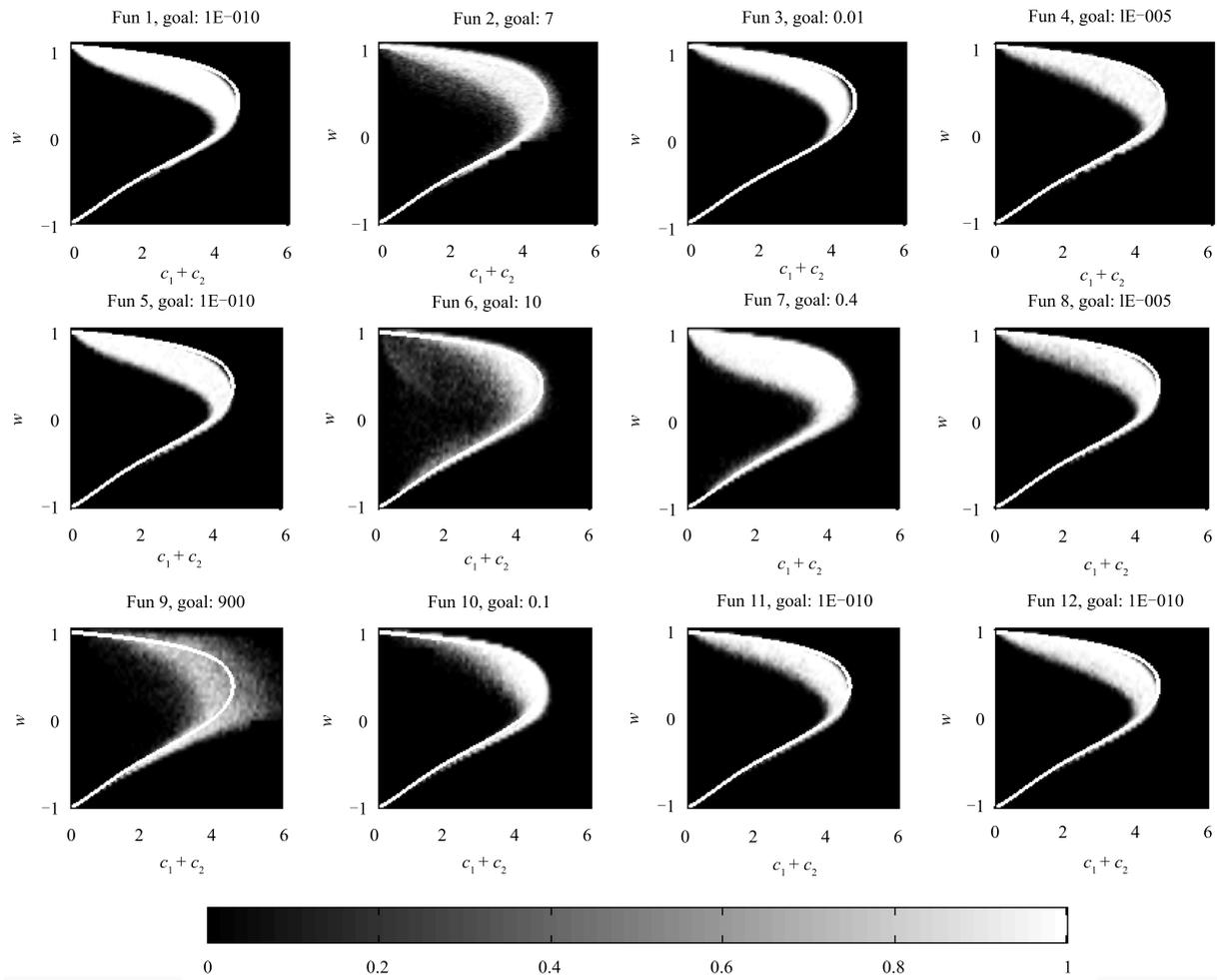


图 3 算法成功率与参数的关系

Fig. 3 Relationship between the algorithm success rate and particle parameters

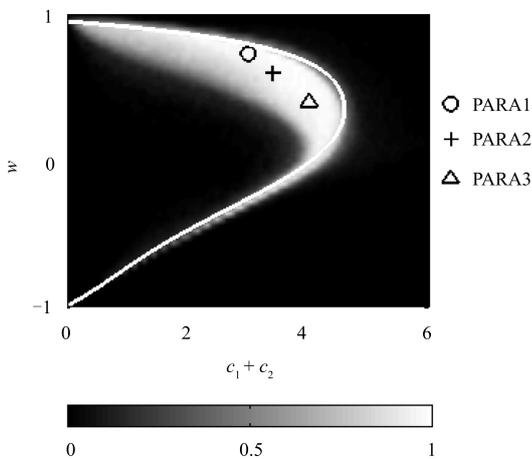


图 4 算法在 12 个测试函数上的性能总和
Fig. 4 Total optimization performance on the 12 test functions

PARA4 的综合性能在所有的参数中是比较差的. 参数 PARA5 是在参数 PARA1 的基础上发展而来

的, 保持 c_1 与 c_2 之和不变, 改变 c_1 和 c_2 的比例, 除了在 Fun 2 上, 参数 PARA5 的性能要优于参数 PARA1.

为了能够比较全面地了解 c_1 和 c_2 的比例对算法的影响, 我们保持参数 PARA1 中的 $w = 0.7298$, $c_1 + c_2 = 2.992$, 将 c_1 在 $c_1 + c_2$ 的总和中所占的比率由 0 逐渐增加到 1, 在 12 个函数上的运行结果如图 5 所示. Fun 2, Fun 6, Fun 7 和 Fun 9 在整个比率范围内的变化情况较为复杂, 图 5(a) 为这 4 个函数的变化情况, 另外 8 个函数的优化情况展示在图 5(b) 中. 当 c_1 在 $c_1 + c_2$ 之和中所占比例 $c_1/(c_1 + c_2)$ 在 0.6~0.8 之间时, 图 5(b) 中的 8 个函数都获得了很好的性能, 而图 5(a) 中, 算法在 Fun 6, Fun 7 和 Fun 9 上的成功率随 $c_1/(c_1 + c_2)$ 的增加而增大, 在 0.8 附近到达最好值; Fun 2 则相反, 成功率随着 $c_1/(c_1 + c_2)$ 的增加而减小, 在 0.8 时的成功率非常低. 当 $c_1/(c_1 + c_2)$ 取为 0.683 时, 便可得到参数组合 PARA5, 对于图 5(b) 的 8 个函数来

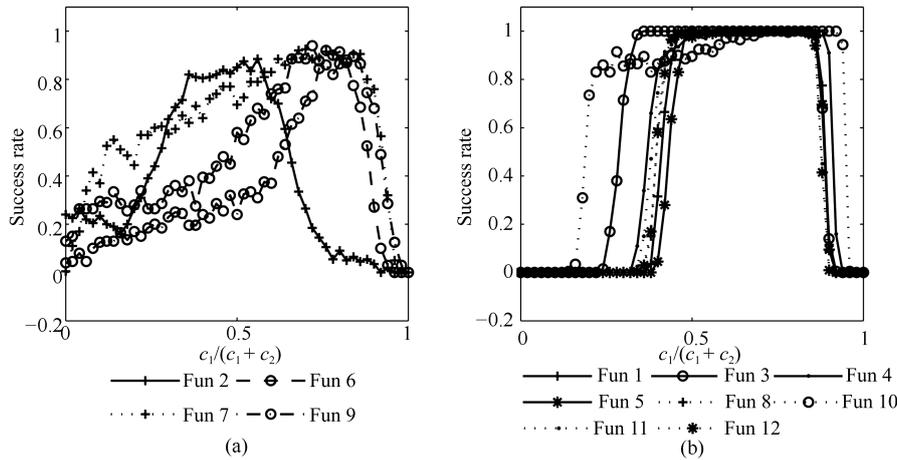


图 5 c_1 所占比率对算法成功率的影响

Fig. 5 The influence of ratio c_1 on algorithm's success rate

表 1 5 组参数的运行结果对比

Table 1 Comparison of running results on 5 parameters

函数	PARA	min	max	mean ± std	SR	函数	PARA	min	max	mean ± std	SR
Fun 1	1	1.47E-25	8.64E-22	6.32E-23 ± 1.34E-22	1.00	Fun 7	1	1.47E-02	1.64E-01	7.49E-02 ± 3.35E-02	0.83
	2	9.05E-36	5.64E-31	1.73E-32 ± 7.16E-32	1.00		2	7.39E-03	1.84E-01	7.77E-02 ± 3.52E-02	0.77
	3	1.98E-40	2.94E-34	1.31E-35 ± 4.50E-35	1.00		3	1.47E-02	2.21E-01	7.94E-02 ± 4.10E-02	0.74
	4	1.81E-57	2.84E-33	2.86E-35 ± 2.84E-34	1.00		4	1.72E-02	3.05E-01	1.12E-01 ± 5.99E-02	0.53
	5	1.63E-27	1.45E-23	2.71E-25 ± 1.46E-24	1.00		5	1.11E-15	1.69E-01	6.15E-02 ± 3.39E-02	0.86
Fun 2	1	2.72E-02	7.29E+00	2.40E+00 ± 1.43E+00	0.84	Fun 8	1	2.38E-13	1.99E+01	1.99E-01 ± 1.99E+00	0.99
	2	3.18E-03	1.20E+02	3.75E+00 ± 1.19E+01	0.76		2	3.55E-15	7.10E-15	3.94E-15 ± 1.11E-15	1.00
	3	1.20E-03	8.27E+00	3.45E+00 ± 1.96E+00	0.30		3	3.55E-15	7.10E-15	3.97E-15 ± 1.16E-15	1.00
	4	6.03E-02	8.96E+00	4.30E+00 ± 1.57E+00	0.07		4	3.55E-15	1.15E+00	1.15E-02 ± 1.15E-01	0.97
	5	2.21E-02	8.45E+00	3.45E+00 ± 1.40E+00	0.31		5	1.42E-14	6.18E-13	1.51E-13 ± 1.19E-13	1.00
Fun 3	1	6.55E-10	8.21E-08	1.38E-08 ± 1.58E-08	1.00	Fun 9	1	0.00E+00	1.30E+03	6.41E+02 ± 2.28E+02	0.35
	2	5.68E-14	1.07E-10	9.80E-12 ± 1.70E-11	1.00		2	0.00E+00	1.42E+03	6.83E+02 ± 2.86E+02	0.30
	3	2.98E-11	7.36E-09	8.82E-10 ± 1.32E-09	1.00		3	1.18E+02	1.54E+03	6.41E+02 ± 3.04E+02	0.44
	4	7.06E-12	4.01E-04	6.10E-06 ± 4.09E-05	0.94		4	1.18E+00	1.19E+03	6.40E+02 ± 2.51E+02	0.36
	5	1.23E-10	1.02E-08	1.63E-09 ± 1.77E-09	1.00		5	0.00E+00	9.49E+02	4.48E+02 ± 1.86E+02	0.67
Fun 4	1	1.06E-14	1.00E+01	5.00E-01 ± 2.19E+00	0.95	Fun 10	1	3.55E-10	1.57E+00	1.46E-01 ± 4.41E-01	0.86
	2	7.73E-20	1.00E+01	1.00E-01 ± 1.00E+00	0.99		2	0.00E+00	3.07E+00	1.69E-01 ± 5.28E-01	0.86
	3	7.38E-23	7.89E-20	9.10E-21 ± 1.60E-20	1.00		3	0.00E+00	1.50E+00	6.00E-02 ± 2.96E-01	0.96
	4	8.37E-20	1.00E+01	1.00E-01 ± 1.00E+00	0.98		4	1.81E-13	3.00E+00	1.40E-01 ± 4.50E-01	0.46
	5	1.64E-15	1.12E-13	1.82E-14 ± 1.87E-14	1.00		5	0.00E+00	3.46E-02	3.46E-04 ± 3.46E-03	0.99
Fun 5	1	1.14E-25	1.00E+04	2.00E+02 ± 1.40E+03	0.00	Fun 11	1	5.76E-27	5.30E-21	5.90E-23 ± 5.30E-22	1.00
	2	6.54E-35	1.82E-29	2.24E-31 ± 1.82E-30	1.00		2	4.71E-32	1.90E-31	4.94E-32 ± 1.53E-32	1.00
	3	1.74E-39	2.36E-33	1.04E-34 ± 3.60E-34	1.00		3	4.71E-32	3.11E-01	1.55E-02 ± 6.81E-02	0.95
	4	2.45E-54	6.74E-34	6.74E-36 ± 6.74E-35	1.00		4	4.71E-32	3.11E-01	1.24E-02 ± 6.12E-02	0.95
	5	1.44E-26	7.31E-24	5.72E-25 ± 9.94E-25	0.32		5	5.16E-29	3.55E-24	5.37E-26 ± 2.57E-25	1.00
Fun 6	1	9.91E-13	1.39E+01	5.73E+00 ± 3.06E+00	0.55	Fun 12	1	6.86E-27	6.65E-22	2.13E-23 ± 7.32E-23	1.00
	2	9.94E-01	1.59E+01	6.61E+00 ± 3.23E+00	0.43		2	1.34E-32	4.65E-31	2.13E-32 ± 4.68E-32	1.00
	3	0.00E+00	1.39E+01	4.87E+00 ± 2.94E+00	0.68		3	1.34E-32	2.72E-26	5.45E-28 ± 3.85E-27	1.00
	4	9.94E-01	4.47E+01	9.87E+00 ± 7.05E+00	0.23		4	1.34E-32	4.23E-13	4.23E-15 ± 4.26E-14	0.99
	5	1.66E-13	7.95E+00	3.29E+00 ± 1.84E+00	0.85		5	2.78E-28	1.93E-23	2.82E-25 ± 1.93E-24	1.00

说,是最好的选择. 而对于图 5(a) 的 4 个函数来说,是综合考虑的折中选择.

3 几种典型参数变化策略的性能对比及一种变化参数推荐值

以上的实验和讨论都是基于在种群进化过程中 c_1 和 c_2 为定值的情形下进行的,从表 1 和图 5 的运行结果可以看到,无论 c_1 和 c_2 是否相等,采用 c_1 和 c_2 定值的调节算法在算法性能上是有局限性的. 为了进一步提高算法的性能,下面观察在迭代的过程中 c_1 和 c_2 按照不同的方式变化时对算法性能的影响. 实验仍然基于 12 个测试函数,对于固定的 w , c_1 和 c_2 有 9 种不同的调整策略,为简洁表示,引入 3 个符号:“ \rightarrow ”表示参数 c_1 或 c_2 保持不变,为 1.5;“ \uparrow ”表示参数 c_1 或 c_2 在迭代的过程中线性地由 0.5 上升到 2.5;“ \downarrow ”表示参数 c_1 或 c_2 在迭代的过程中

线性地由 2.5 下降到 0.5. 则 c_1 和 c_2 的 9 种调节策略表示为:

- 1) $c_1 \rightarrow, c_2 \rightarrow$; 2) $c_1 \rightarrow, c_2 \uparrow$; 3) $c_1 \rightarrow, c_2 \downarrow$;
- 4) $c_1 \uparrow, c_2 \rightarrow$; 5) $c_1 \uparrow, c_2 \uparrow$; 6) $c_1 \uparrow, c_2 \downarrow$;
- 7) $c_1 \downarrow, c_2 \rightarrow$; 8) $c_1 \downarrow, c_2 \uparrow$; 9) $c_1 \downarrow, c_2 \downarrow$.

对于每个函数,进行如下实验: w 从 0 逐渐增加到 1,对于每个 w 的值,分别用以上 9 种策略进行实验. 每个优化实验运行 50 次. 记录最终结果到达指定精度的成功率. 运行结果如图 6 所示,纵轴为算法寻优的成功率,横轴为 w 值,每个子图中的 9 条曲线,分别对应上述 c_1 和 c_2 的 9 种取值策略.

由图 6 可以看到,每一种 c_1 和 c_2 的取值策略都有与其对应的 w 的最佳取值范围. 除了 Fun 2, Fun 6, Fun 7 和 Fun 9,各种策略的性能随着 w 的变化情况是类似的,即:沿着 w 轴的正方向, w 的整个区间可以分为 5 个区域: 1) 初始阶段,成功率几乎为 0; 2) 紧随其后的很小的区域内,成功率由 0 迅速

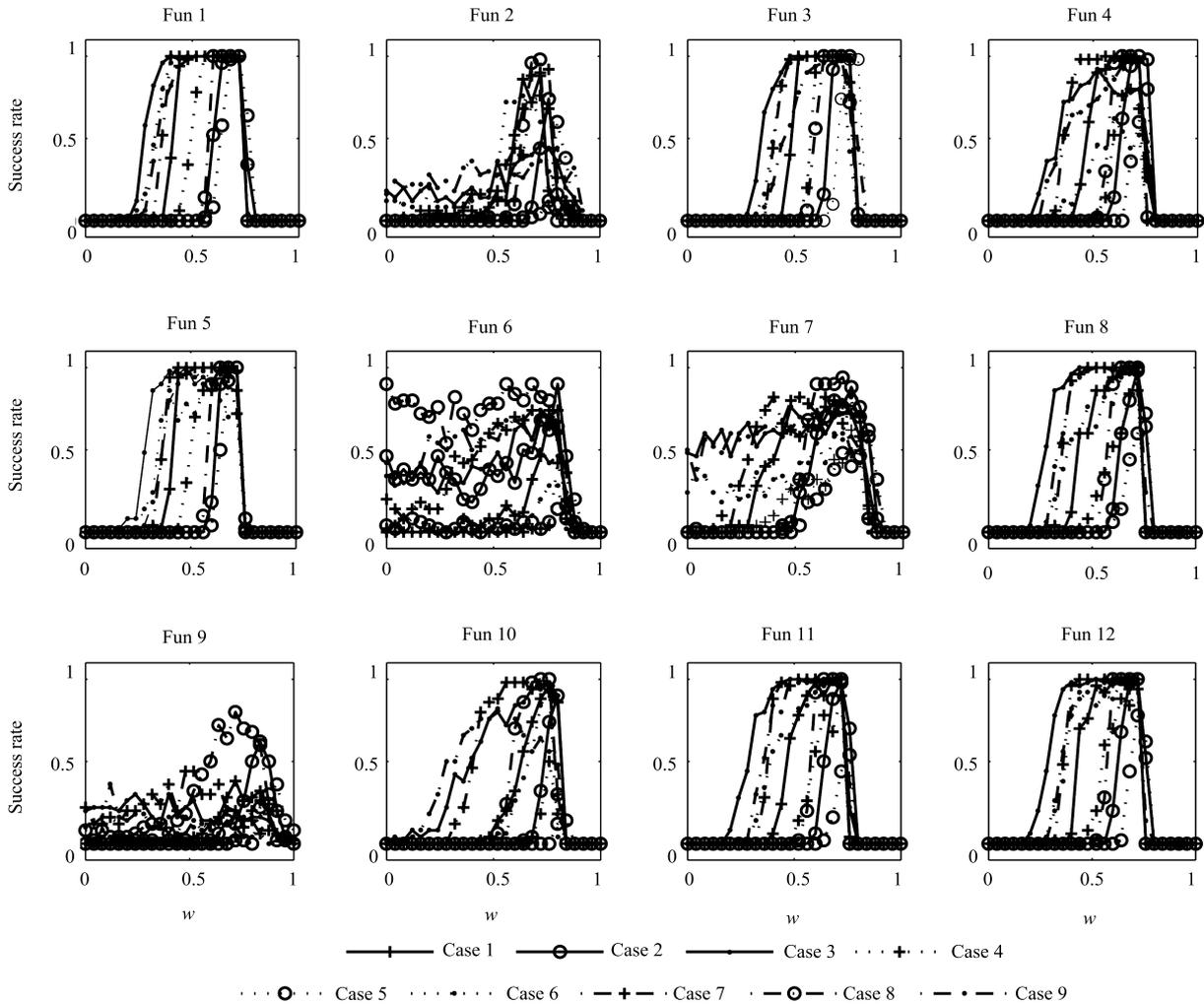


图 6 变化的 c_1 和 c_2 取值策略对算法成功率的影响
Fig. 6 The influence of varying c_1 and c_2 on algorithm's success rate

上升达到曲线的最大值; 3) 在一段区域内, 成功率保持为最大值; 4) 紧随其后的很小的区域内, 成功率迅速下降; 5) 最后区域, 成功率几乎为零. 区域 1 和区域 3 的大小与参数的取值策略紧密相关. 而所有的策略在 $w > 0.68$ 时, 成功率几乎均为 0. 区域 3 正是我们需要的部分. 在这 8 个函数上, 策略 3、策略 7 和策略 9 使得区域 3 较为宽广. 再来看 Fun 6, Fun 7 和 Fun 9, 策略 2 取得了最好的成功率, 其次为策略 8. 在 Fun 6 和 Fun 7 上, 对于某些 c_1 和 c_2 的取值策略, w 的取值可以比较小, 也能够使算法获得不错的性能. 在 Fun 2 上, 各个策略相适应的 w 值范围都很小. 相比较而言, 不论是从 w 的适用范围的大小还是从成功率的高低来看, 策略 4、策略 5 和策略 6 都是不可取的, 即增大的调节方式总是不可取的. 综合 12 个测试函数来看, 策略 8 在 Fun 6, Fun 7 和 Fun 9 上, 在 $w = 0.68$ 时具有很突出的效果, 并且, 在其他 8 个函数上, 虽然策略 8 的 w 的适用范围并不宽广, 但在 $w = 0.68$ 时, 恰恰都具有很好的性能, 因此, $w = 0.68$, c_1 在迭代过程中由 2.5 线性减小到 0.5, c_2 由 0.5 线性增加到 2.5, 是一种很好的参数设置方式. 这与 Ratnaweera 等^[17] 提出的 c_1 递减和 c_2 递增的调整策略是相符的, 在文献 [17] 中, c_1 和 c_2 的变化范围与本文相同, 而 w 的取值由 0.9 随迭代线性减小到 0.4. 本文提出的和文献 [17] 中的参数设置方法在 12 个函数上的运行结果如表 2 所示. 在表 2 中, Method-ours 指本文推荐的 $w = 0.68$, c_1 在迭代过程中由 2.5 线性减小到 0.5, c_2 由 0.5 线性增加到 2.5. Method-[17] 是指文献 [17] 给出的 w 的取值由 0.9 随迭代线性减小到 0.4, c_1 和 c_2 的变化范围与 Method-ours 相同. 由表 2 的运行结果可以看到, 本文的参数设置方法要优于文献 [17] 中的方法. 值得注意的是, c_1 和 c_2 的变化范围, 与使其获得较优性能的 w 也是相互配合的, 在我们的实验中发现, 当变化范围为 [0.5, 2.05] 时, w 取 0.75 的性能要优于 0.68. 因此, 3 个参数之间的相互配合是非常重要的.

4 本文参数设置方式优化性能的进一步验证

为进一步验证本文参数设置方法的有效性, 将第 2 节的一组固定参数推荐值和第 3 节的时变参数设置方式, 应用于 2015 年的进化计算世界大会, 即 CEC2015 发布的 15 个基准函数^[19].

对于参数优化能力的验证, 种群大小与前面所有试验设置相同 (设置为 50), 函数维数为 10, 最大迭代次数为 5 000, 每个函数运行次数为 100. 试验仅记录达到指定精度的成功率 (Success rate, SR), 其中, F1~F15 的指定精度分别为: 3 000, 1 500, 20, 10, 500, 1 000, 10, 200, 110, 1 000, 500, 110,

30, 3 000, 110.

表 2 本文与文献 [17] 参数设置的优化结果比较

Table 2 Comparison of optimization results between the parameters set in this paper and [17]

函数	Method	min	max	mean \pm std	SR
Fun 1	Ours	3.91E-33	1.42E-29	1.20E-30 \pm 2.51E-30	1.00
	[17]	5.01E-33	6.95E-29	3.09E-30 \pm 9.83E-30	1.00
Fun 2	Ours	9.79E-02	2.26E+00	1.30E+00 \pm 5.31E-01	1.00
	[17]	1.00E+00	5.72E+00	2.40E+00 \pm 7.33E-01	0.84
Fun 3	Ours	5.90E-13	2.27E-10	2.13E-11 \pm 3.88E-11	1.00
	[17]	6.03E-13	2.48E-10	2.99E-11 \pm 4.90E-11	1.00
Fun 4	Ours	5.58E-18	7.20E-16	1.60E-16 \pm 1.52E-16	1.00
	[17]	8.12E-18	2.87E-13	6.86E-15 \pm 4.09E-14	1.00
Fun 5	Ours	7.24E-32	3.86E-28	1.60E-29 \pm 5.49E-29	1.00
	[17]	2.12E-32	3.44E-28	1.65E-29 \pm 5.70E-29	1.00
Fun 6	Ours	9.94E-01	7.95E+00	3.34E+00 \pm 1.55E+00	0.92
	[17]	1.98E+00	1.09E+01	5.49E+00 \pm 2.36E+00	0.54
Fun 7	Ours	0.00E+00	1.42E-01	5.20E-02 \pm 2.94E-02	0.96
	[17]	0.00E+00	1.84E-01	7.81E-02 \pm 3.58E-02	0.76
Fun 8	Ours	3.55E-15	7.10E-15	4.68E-15 \pm 1.67E-15	1.00
	[17]	3.55E-15	2.13E-14	7.81E-15 \pm 3.58E-15	1.00
Fun 9	Ours	1.18E+02	1.04E+03	4.71E+02 \pm 1.99E+02	0.64
	[17]	1.18E+02	1.19E+03	5.48E+02 \pm 2.16E+02	0.42
Fun 10	Ours	0.00E+00	1.44E-02	6.04E-04 \pm 2.51E-03	0.94
	[17]	0.00E+00	1.15E-01	3.03E-03 \pm 1.64E-02	0.84
Fun 11	Ours	4.71E-32	2.90E-27	9.03E-29 \pm 4.35E-28	1.00
	[17]	4.71E-32	1.34E-28	3.09E-30 \pm 1.90E-29	1.00
Fun 12	Ours	1.34E-32	6.16E-29	3.88E-30 \pm 9.45E-30	1.00
	[17]	1.34E-32	1.09E-02	2.19E-04 \pm 1.55E-03	0.98

对图 4 给出的 3 种典型固定参数, 即文献 [6] 和文献 [7] 推荐的参数取值 PARA1 和 PARA2, 以及本文的参数推荐取值 PARA3, 表 3 给出了应用于 CEC2015 的 15 个基准函数上运行结果的对比. 可以看出, 本文的参数推荐值 PARA3 在 15 个函数上的表现略优于 PARA1 和 PARA2.

对第 3 节推荐的时变参数设置方式和文献 [17] 的参数设置方式, 表 4 给出了应用于 CEC2015 的 15 个基准函数上运行结果的对比. 可以看出, 本文推荐的时变参数设置方式在 15 个函数上的表现优于文献 [17] 的参数设置方式.

5 本文推荐的 PSO 参数应用于连续优化问题和离散优化问题的讨论

PSO 算法最初是针对连续优化问题提出的, 相

表 3 本文给出的固定参数推荐值在 CEC2015 基准函数 F1~F15 上的运行结果

Table 3 Running results on CEC2015 Benchmark functions F1~F15 with fixed parameters

PARAM	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15
[6]	0.02	0.02	0.30	0.24	0.45	0.24	0.95	0.08	0.90	0.56	0.90	0.92	0.04	0.12	0.40
[7]	0.08	0.12	0.40	0.14	0.40	0.10	1.00	0.30	0.85	0.68	0.80	0.91	0.02	0.16	0.48
Ours	0.02	0.10	0.10	0.18	0.50	0.14	1.00	0.15	0.90	0.60	0.95	0.95	0.04	0.30	0.50

表 4 本文给出的时变参数设置方式在 CEC2015 基准函数 F1~F15 上的运行结果

Table 4 Running results on CEC2015 Benchmark functions F1~F15 with time-varying parameters

Method	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15
Ours	0.70	0.26	0.30	0.94	1.00	0.64	1.00	0.50	1.00	0.95	1.00	1.00	0.25	0.40	0.96
[17]	0.16	0.14	0.01	0.82	0.85	0.25	1.00	0.22	1.00	0.85	1.00	1.00	0.20	0.40	0.90

关研究也主要集中在连续函数优化方面^[20-22], 即第 1 节描述的标准算法被广泛研究和应用. 本文研究的出发点也主要是针对连续优化问题, 文中给出的 12 个标准测试函数和 CEC2015 发布的 15 个基准函数^[19] 的优化测试, 验证了本文方法在连续优化问题上的有效性.

对于离散优化 (或组合优化) 问题而言, 解空间是离散点的集合, 而非连续区域, 因此利用 PSO 算法解决离散优化问题, 一般需要修正位置和速度更新公式, 或者对问题进行变形, 这方面的工作大致可分为如下三类^[21-22]: 1) 直接将连续 PSO 用于离散优化问题的求解; 2) 将速度作为位置变化的概率; 3) 重新定义 PSO 算法的操作算子. 对于前两种情形, 本文推荐的参数取值方法仍然适用, 因为算法在本质上依然遵从标准的 PSO 连续问题求解规则. 但对于第三种情形, 一般都是针对要求解的具体问题, 定义不同的 PSO 操作算子, 使得 PSO 算法在形式上与标准 PSO 算法差别较大, 从而本文推荐的参数取值方法难以保证其优化效果, 这方面的问题有待进一步研究.

6 结论

粒子群算法的可调参数共同影响着算法的优化性能, 本文将研究重点放在了惯性权重 w 和加速因子 c_1, c_2 的设定和调节上. 基于 12 个常用的、不同类型的基准函数, 在 $w \in [-1, 1]$, $c_1 + c_2 \in [0, 8]$ 的参数区域内进行了全面的仿真研究. 通过实验结果可以看到, 常常被提到的“惯性权重起着调节种群全局搜索能力和局部搜索能力”的说法是欠全面的, 种群的搜索能力和算法的性能, 是依靠 w, c_1 和 c_2 的相互配合来调节的, 并不是仅靠一个参数可以决定的. 在对固定参数的情形进行仿真并给出参数推荐

值后, 还研究了时变参数对算法的影响, 给出了一种推荐的时变参数设置方法. 最后通过 CEC2015 发布的 15 个基准函数进一步验证了本文推荐的参数选取方法的有效性, 并讨论了 PSO 算法在连续优化问题和离散优化问题方面的应用.

References

- Kennedy J, Eberhart R. Particle swarm optimization. In: Proceedings of the 1995 IEEE International Conference on Neural Networks. Perth, Australia: IEEE, 1995. 1942-1948
- Shi Y, Eberhart R. A modified particle swarm optimizer. In: Proceedings of the 1998 IEEE International Conference on Evolutionary Computation. Anchorage, AK, USA: IEEE, 1998. 69-73
- Zhang Yong, Gong Dun-Wei, Zhang Wan-Qiu. A simplex method based improved particle swarm optimization and analysis on its global convergence. *Acta Automatica Sinica*, 2009, **35**(3): 289-298
(张勇, 巩敦卫, 张婉秋. 一种基于单纯形法的改进微粒群优化算法及其收敛性分析. *自动化学报*, 2009, **35**(3): 289-298)
- Pan Feng, Zhou Qian, Li Wei-Xing, Gao Qi. Analysis of standard particle swarm optimization algorithm based on Markov chain. *Acta Automatica Sinica*, 2013, **39**(4): 381-389
(潘峰, 周倩, 李位星, 高琪. 标准粒子群优化算法的马尔科夫链分析. *自动化学报*, 2013, **39**(4): 381-389)
- Liu Zhao-Hua, Zhou Shao-Wu, Liu Kan, Zhang Jing. Permanent magnet synchronous motor multiple parameter identification and temperature monitoring based on binary-modal adaptive wavelet particle swarm optimization. *Acta Automatica Sinica*, 2013, **39**(12): 2121-2130
(刘朝华, 周少武, 刘侃, 章兢. 基于双模态自适应小波粒子群的永磁同步电机多参数识别与温度监测方法. *自动化学报*, 2013, **39**(12): 2121-2130)
- Eberhart R C, Shi Y. Comparing inertia weights and constriction factors in particle swarm optimization. In: Proceedings of the 2000 Congress on Evolutionary Computation. La Jolla, CA: IEEE, 2000. 84-88

- 7 Trelea I C. The particle swarm optimization algorithm: convergence analysis and parameter selection. *Information Processing Letters*, 2003, **85**(6): 317–325
- 8 Samal N R, Konar A, Das S, Abraham A. A closed loop stability analysis and parameter selection of the particle swarm optimization dynamics for faster convergence. In: Proceedings of the 2007 IEEE Congress on Evolutionary Computation. Singapore: IEEE, 2007. 1769–1776
- 9 Carlisle A, Dozier G. An off-the-shelf PSO. In: Proceedings of the 2001 Workshop on Particle Swarm Optimization. Indianapolis, USA, 2001. 1–6
- 10 Hu Jian-Xiu, Zeng Jian-Chao. Selection on inertia weight of particle swarm optimization. *Computer Engineering*, 2007, **33**(11): 193–195
(胡建秀, 曾建潮. 微粒群算法中惯性权重的调整策略. *计算机工程*, 2007, **33**(11): 193–195)
- 11 Yan Li-Ping, Zeng Jian-Chao. Particle swarm optimization with self-adaptive stochastic inertia weight. *Computer Engineering and Design*, 2006, **27**(24): 4677–4679, 4706
(延丽平, 曾建潮. 具有自适应随机惯性权重的 PSO 算法. *计算机工程与设计*, 2006, **27**(24): 4677–4679, 4706)
- 12 Chalermchaiarbha S, Ongsakul W. Stochastic weight trade-off particle swarm optimization for nonconvex economic dispatch. *Energy Conversion and Management*, 2013, **70**: 66–75
- 13 Ao Yong-Cai, Shi Yi-Bing, Zhang Wei, Li Yan-Jun. Improved particle swarm optimization with adaptive inertia weight. *Journal of University of Electronic Science and Technology of China*, 2014, **43**(6): 874–880
(敖永才, 师奕兵, 张伟, 李焱骏. 自适应惯性权重的改进粒子群算法. *电子科技大学学报*, 2014, **43**(6): 874–880)
- 14 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
- 15 Alfi A. PSO with adaptive mutation and inertia weight and its application in parameter estimation of dynamic systems. *Acta Automatica Sinica*, 2011, **37**(5): 541–549
- 16 Suganthan P N. Particle swarm optimiser with neighbourhood operator. In: Proceedings of the 1999 Congress on Evolutionary Computation. Washington DC, USA: IEEE, 1999. 1958–1962
- 17 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
- 18 Yamaguchi T, Yasuda K. Adaptive particle swarm optimization: self-coordinating mechanism with updating information. In: Proceedings of the 2006 IEEE International Conference on Systems, Man, and Cybernetics. Taipei, China: IEEE, 2006. 2303–2308
- 19 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, Zhengzhou China, and Technical Report, Nanyang Technological University, Singapore, 2014.
- 20 Cui Zhi-Hua, Zeng Jian-Chao. *Particle Swarm Optimization*. Beijing: Science Press, 2011.
(崔志华, 曾建潮. 微粒群优化算法. 北京: 科学出版社, 2011.)
- 21 Wang Ling, Liu Bo. *Particle Swarm Optimization and Scheduling Algorithms*. Beijing: Tsinghua University Press, 2008.
(王凌, 刘波. 微粒群优化与调度算法. 北京: 清华大学出版社, 2008.)
- 22 Guo Wen-Zhong, Chen Guo-Long. *Discrete Particle Swarm Optimization Algorithm and Its Application*. Beijing: Tsinghua University Press, 2012.
(郭文忠, 陈国龙. 离散粒子群优化算法及其应用. 北京: 清华大学出版社, 2012.)



王东风 博士, 华北电力大学控制与计算机工程学院教授. 主要研究方向为群智能优化算法和智能控制.

E-mail: wangdongfeng@ncepu.edu.cn
(WANG Dong-Feng Ph.D., professor at the School of Control and Computer Engineering, North China Electric Power University. His research

interest covers swarm intelligence-based optimization and intelligent control.)



孟丽 华北电力大学控制与计算机工程学院博士研究生. 主要研究方向为智能优化算法及其应用. 本文通信作者.

E-mail: mengli2014@163.com
(MENG Li Ph.D. candidate at the School of Control and Computer Engineering, North China Electric Power University. Her research interest covers

swarm intelligence-based optimization and its application. Corresponding author of this paper.)