

Hybrid Optimization Algorithm of PSO and Cuckoo Search

Fan Wang

School of Science
Xi'an Polytechnic University
Xi'an, China
xigua_912@163.com

Ligui Luo

School of Science
Xi'an Polytechnic University
Xi'an, China
Luoligui803@163.com

Xing-shi He

School of Science
Xi'an Polytechnic University
Xi'an, China
xingshi_he@163.com

Yan Wang

School of Science
Xi'an Polytechnic University
Xi'an, China
Wangyan011105@163.com

Abstract—By making use of the advantages of Cuckoo Search (CS) algorithm and Particle Swarm Optimization (PSO), a hybrid optimization algorithm of PSO and CS was proposed. By CS-PSO, the search area of PSO was extended, and the defect of PSO is easily fall into point of local extremum that was improved. The simulation experiment results show that CS-PSO hybrid algorithm is better.

Keywords—PSO algorithm; Cuckoo Search algorithm; CS-PSO hybrid algorithm; global optimum; intelligent algorithm

I. 引言

Particle Swarm Optimization(PSO)^[1]是 1995 年由 Kennedy 和 Eberhar 提出的一种智能优化算法,其算法思想来源于对鸟类等生物群体觅食行为的研究,该算法具有简单易实现,搜索速度快等特点,现在已成功的应用在函数优化,工程优化等领域中,具有广泛的实用性。但是,由于 PSO 的搜索过程主要是靠不断的追逐当前全局最优解来得到最终的最优解,所以,粒子在搜索后期缺乏活力并逐渐同一化,导致算法容易陷入局部极值点,搜索精度差。针对该不足,众多研究者提出了不同的改进方法^{[2]-[7]},以提高基本 PSO 的搜索能力。

Cuckoo Search(CS)算法^[8]是 2009 年由剑桥大学的 Xin-She Yang 等人从模拟布谷鸟寻窝产卵行为中受到启发,提出的一种新型的智能优化算法。该算法具有选用参数少,搜索路径优,寻优能力强等特点,目前已在工程优化问题中得到了成功的应用^[9]。利用 CS 算法的这些特点,当 PSO 搜索到每代的局部最优解和全局最优解时不直接进入下一代,而是将 CS 算法结合进去继续进行搜索,这样就加大了原来 PSO 的搜索范围,增加了粒子搜索的活力,从而有效的弥补了 PSO 易陷入局部极值点的不足。

II. 基本 PSO 和 CS 算法

A. 基本 PSO

PSO 是通过 n 个粒子不断跟随当代各自的最优解和全局最优解来搜索全局最优解的,每个粒子的状态都是由三个元素构成:粒子的位置,粒子的速度,粒子的适应度。PSO 首先是在可行解空间中随机初始化粒子的初始位置和初始速度,计算粒子的适应度,然后进入算法的迭代更新过程。例如, d 维搜索空间中的第 i 个粒子的位置 $X_i = (x_{i1}, x_{i2}, \dots, x_{id})$, 速度是 $V_i = (v_{i1}, v_{i2}, \dots, v_{id})$, $1 \leq i \leq n$, 通过评价各粒子的适应度,确定第 t 代的每个粒子的最优位置 $p_i = (p_{i1}, p_{i2}, \dots, p_{id})$ 和全局最优位置 $p_g = (p_{g1}, p_{g2}, \dots, p_{gd})$, 再按如下公式^[1]进行粒子的速度和位置更新。

$$v_{id}(t+1) = \omega v_{id}(t) + c_1 r_1 [p_{id} - x_{id}(t)] + c_2 r_2 [p_{gd} - x_{id}(t)], i=1, \dots, d$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1), i=1, \dots, d$$

其中, ω 为惯性权重,其大小决定了粒子继承当前速度的多少; c_1, c_2 为两个正常数的学习因子,通常取值为 2; r_1, r_2 是 $[0,1]$ 之间的服从均匀分布的随机数。另外,通过设置每一个粒子的速度范围 $[v_{\min}, v_{\max}]$ 和位置范围 $[x_{\min}, x_{\max}]$, 可以对粒子的移动进行的适当的限制。

一般情况下,速度更新公式中的 ω 是由最大权重 ω_{\max} 线性减小到最小权重 ω_{\min} , 即:

$$\omega(t) = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{T_{\max}} \cdot t \quad (1)$$

其中 t 是当代迭代数; T_{\max} 是总的迭代次数;

This work supported by the Nature Sciences Foundation of Shaanxi Provincial Education Administration Bureau (2010JK563) and the Graduate Student Innovation Funds of Xi'an Polytechnic University (chx110922)

B. CS 算法

CS 算法模拟了布谷鸟为寻找适合产卵的鸟窝而随机游走的寻窝过程, 在寻窝过程中, 首先, 在可行解空间中随机初始化 n 个鸟窝的位置, 计算每个鸟窝位置的适应度, 并保留适应度最优的鸟窝位置到下一代, 然后进入鸟窝位置的迭代过程, 设 d 维搜索空间中的第 i 个鸟窝的位置 $X_i = (x_{i1}, x_{i2}, \dots, x_{id})$, $1 \leq i \leq n$, 通过评价每个鸟窝的适应度, 确定第 t 代的每个鸟窝的最优位置 $p_i = (p_{i1}, p_{i2}, \dots, p_{id})$ 和全局最优位置 $p_g = (p_{g1}, p_{g2}, \dots, p_{gd})$, 保留第 t 代鸟窝的全局最优位置, 再按如下公式进行位置更新,

$$x_i(t+1) = x_i(t) + \alpha \oplus Levy(\lambda), \quad i = 1, 2, \dots, n$$

其中 $x_i(t)$ 表示第 i 个鸟窝在第 t 代的鸟窝位置, \oplus 为点对点乘法, α 表示步长控制量, $Levy(\lambda)$ 为随机搜索路径. 并且 $Levy \sim u \cdot t^{-\lambda}, (1 < \lambda \leq 3)$. 通过位置更新后, 用随机数 $r \in [0, 1]$ 与鸟窝的主人发现外来鸟的概率 p_a 对比, p_a 通常取值为 0.25. 若 $r > p_a$, 则对 $x_i(t+1)$ 进行随机改变, 反之不变。

III. PSO 与 CS 的混合优化算法

PSO 与 CS 的混合优化算法(CS-PSO)的基本思想: 在一次迭代中, 先用 PSO 进行速度和位置更新, 得到一组粒子的最优位置 $P_i = (p_1, p_2, \dots, p_i, \dots, p_n)^T$, 和一个全局最优位置 p_g , 保留全局最优位置 p_g , 将这组粒子的最优位置 P_i 进入 CS 算法中继续进行迭代更新, 算法具体流程如图 1.

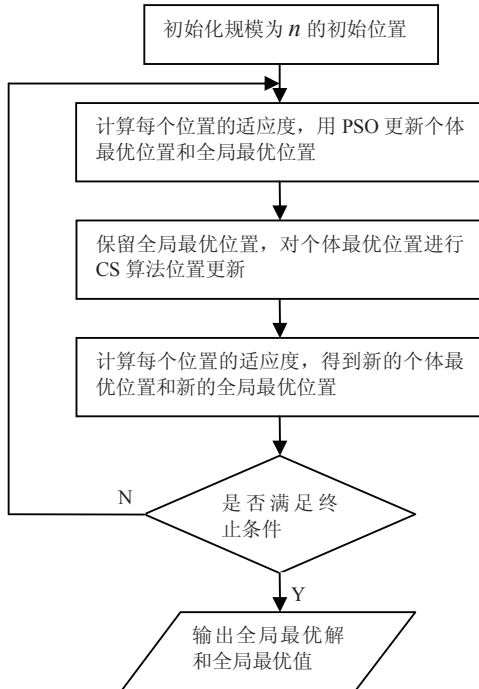


图 1 CS-PSO 混合优化算法的流程图

IV. 仿真实验

A. 实验设计及 CS-PSO 算法与 PSO 的进化曲线比较

为了评价 CS-PSO 混合算法的优化性能, 实验设计利用以下 5 个基准测试函数分别对 PSO 和 CS-PSO 两种算法进行测试。

$$f_1(x) = \sum_{i=1}^d x_i^2, \quad x \in [-100, 100];$$

$$f_2(x) = \sum_{i=1}^d |x_i| + \prod_{i=1}^d |x_i|, \quad x \in [-10, 10];$$

$$f_3(x) = \sum_{i=1}^d [x_i^2 - 10 \cos(2\pi x_i) + 10], \quad x \in [-5.12, 5.12];$$

$$f_4(x) = \sum_{i=1}^d [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2], \quad x \in [-30, 30];$$

$$f_5(x) = \frac{1}{4000} \sum_{i=1}^d x_i^2 - \prod_{i=1}^d \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, \quad x \in [-600, 600].$$

以上五个基准测试函数的理想最优值均为 0。

仿真实验中, 算法参数设置为: 维数是 10, 种群规模为 40, ω 惯性权重按 (1) 式线性递减, 并设置 $\omega_{\max} = 0.9, \omega_{\min} = 0.4$, 学习因子 $c_1 = 2, c_2 = 2$, 发现概率 $p_a = 0.25$, 最大迭代次数为 2000. 分别用 PSO 和 CS-PSO 两种算法对以上 5 个基准测试函数进行测试, 图 2-图 6 给出了以上 5 个测试函数的最优适应值在两种算法中的对比进化曲线, 为了对比清楚, 画图时对两个算法的适应值均取以 10 为底的对数进行对比。由图可见, CS-PSO 比 PSO 的优化精度较高。

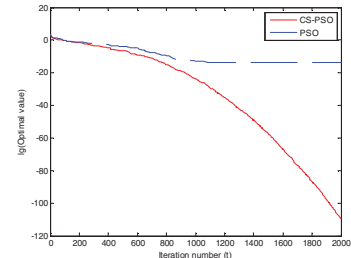


图 2 f_1 的进化曲线

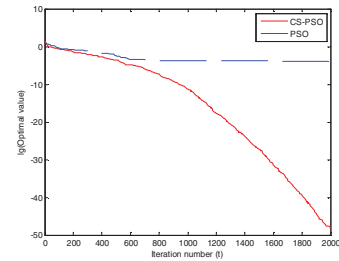


图 3 f_2 的进化曲线

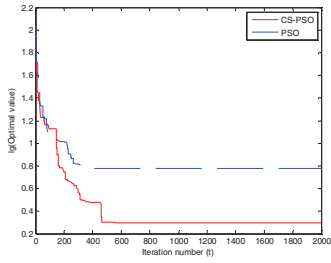


图4 f_3 的进化曲线

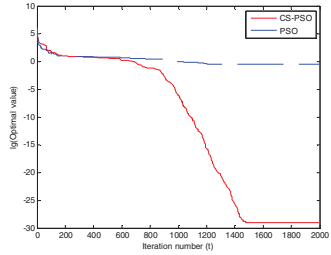


图5 f_4 的进化曲线

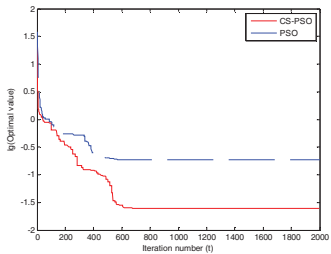


图6 f_5 的进化曲线

B. PSO, CS 和 CS-PSO 优化精度的比较

将 PSO, CS 和 CS-PSO 三种算法分别经过 50 次独立的模拟仿真, 参数设置同上文一致, 比较三种算法的精度, 如表 1.

表 1 三种优化算法的适应值对比表

函数	算法名称	平均适应值	最大适应值	最小适应值	适应值标准差
f_1	PSO	4.8192e-012	9.0694e-011	4.2781e-021	1.8598e-011
	CS	5.2196e-030	4.7958e-029	4.1459e-031	7.5224e-030
	CS-PSO	8.6006e-104	3.6128e-102	7.7156e-112	5.1257e-103
f_2	PSO	8.5316e-004	5.3869e-003	1.0224e-005	1.1067e-003
	CS	3.6977e-013	1.8355e-012	8.5754e-014	3.1067e-013
	CS-PSO	1.7580e-048	1.8511e-047	2.2911e-052	4.5418e-048
f_3	PSO	1.0863e+001	1.7912e+001	3.9794e+000	3.6515e+000
	CS	3.0866e+000	5.2073e+000	1.2142e+000	9.2501e-001
	CS-PSO	3.0645e+000	7.9597e+000	0	1.6556e+000
f_4	PSO	2.4945e+000	8.0316e+000	2.4682e-003	2.2097e+000
	CS	3.5001e-002	6.3982e-001	2.3915e-005	9.8400e-002
	CS-PSO	7.9721e-002	3.9866e+000	0	5.6381e-001
f_5	PSO	1.1765e-001	3.0785e-001	1.2327e-002	6.0996e-002
	CS	2.9917e-002	6.9976e-002	3.3457e-003	1.3648e-002
	CS-PSO	6.5201e-002	2.2635e-001	7.3960e-003	3.7009e-002

由表 1 可以看出, CS-PSO 混合算法与 PSO 相比计算精度明显提高, 搜索最优解的波动较小, CS-PSO 混合算法与 CS 算法相比较, 通过 f_1 和 f_2 函数测试结果精度也有较为明显的提高, 而 f_3 , f_4 和 f_5 这三个测试函数较为复杂, 通过这三个函数测试后结果相差不大. 说明, CS-PSO 混合算法综合了 PSO 和 CS 两者的优点, 优化效果较好.

C. SA-PSO, PSOD, CS-PSO 优化性能比较

CS-PSO 与文献[10]中的优化算法 SA-PSO, PSODE 进行对比, 设置最大, 最小速度分别为搜索上, 下限的一半, 迭代次数为 1500 次, 独立运行 20 次取其平均值进行对比, 其他参数设置同上文一致, 实验结果如表 2, 可以看出 CS-PSO 优化结果优于 SA-PSO, PSODE 的优化结果.

表 2 SA-PSO, PSODE 与 CS-PSO 的平均最优值对比表

测试函数	SA-PSO	PSODE	CS-PSO
f_1	1.3778e-012	1.5486e-055	1.1318e-079
f_4	6.5346	0.8783	0.0797
f_5	0.0982	0.0313	0.0150

V. 结束语

CS-PSO 混合优化算法是基于 PSO 与 CS 的混合算法, 它即保持了 PSO 的搜索性能, 又继承了 CS 算法的全局搜索能力强的特点, 从而提高了算法的优化能力和搜索精度, 并有效的克服了基本 PSO 容易陷入局部最优的缺陷. 实验结果表明, CS-PSO 混合优化算法无论与基本算法还是和文献中的混合算法比较都具有更好的优化能力.

致谢

感谢陕西省教育厅自然科学基金和西安工程大学研究生创新基金给予的资助. 同时感谢导师贺兴时教授给予的悉心的指导和无私的帮助!

References

- [1] Kenned J, Eberhart R, "Particle Swarm Optimization," Proc. of IEEE Int'l Conf, on Neural Networks, Perth, Australia, 1995, pp. 1942-1948.
- [2] Angeline P J, "Evolutionary optimization versus particle swarm optimization: philosophy and performance differences," Evolutionary Programming, vol.7, 1998, pp.601- 610.
- [3] Higashi N, Iba H, "Particle Swarm Optimization with Gaussian Mutation," Proc. of the IEEE Swarm Intelligence Symposium, Indianapolis, Indiana, USA, IEEE Press, 2003, pp.72-79.
- [4] Li-juan Luan, Li-jing, Tan Ben, A novel hybrid optimization algorithm based on particle swarm optimization a differential evolution, Information and control, vol. 36(6), 2007, pp708- 714 (In Chinese).
- [5] Gang Xiao, Shou-zhi Li, Xuan-hong Wang, Rui Xiao, "A Solution to Unit Commitment Problem by ACO and PSO Hybrid Algorithm," Proc. of the IEEE Intelligent Control command Automation, WCICA, 2006, pp. 7475-7479.
- [6] Wen-zhou Yi, Chao-ying Zhang, Qiang Wang, Ya-mei Xu, Jin-ling Zhou, Hybrid Algorithm Based on Improved PSO and DE, Computer Engineering, vol.36 (10), 2010, pp.233- 235 (In Chinese).
- [7] Lian-guo Wang, Qiu-hong Shi, Yi Hong, Hybrid Optimization Algorithm of PSO and AFSA, Computer Engineering, vol.36 (5), 2010, pp.176-178 (In Chinese).

- [8] X. S Yang, S. Deb, "Cuckoo search via Levy flights," Proceedings of World Congress on Nature & Biologically Inspired Computing, India, IEEE Publications, 2009, pp.210-214.
- [9] X.S Yang, S. Deb, "Engineering optimization by cuckoo search," Int. J. Math .Modeling & Num. Optimization, vol. 1(4), 2010, pp.330-343.
- [10] Li Li, Ben Niu, "Particle Swarm Optimization algorithm," metallurgy industry press, Beijing, 2009, pp.82-91 (In Chinese).