

经验自举粒子群优化算法

徐明亮, 须文波, 何 胜

XU Ming-liang, XU Wen-bo, HE Sheng

江南大学 信息工程学院, 江苏 无锡 214122

School of Information Technology, Jiangnan University, Wuxi, Jiangsu 214122, China

E-mail: xml1973@126.com

XU Ming-liang, XU Wen-bo, HE Sheng. Experience improving particle swarm optimizer. Computer Engineering and Applications, 2008, 44(31): 87-89.

Abstract: This paper presents a variants of particle swarm optimizers named Experience Improving Particle Swarm Optimizer (EIP-SO) in which a operator called Experience Improving (EI) is introduced. The EI operator initializes the part of the particle's experience and gets the other experience. The new particle's experience is selected from these two experiences according to their fitness. In each iteration of step, EI operator is performed with the probability of p , the evolution of particles is executed with the probability of $1-p$. The new optimizer enables the diversity of the swarm to be preserved to discourage premature convergence. The result of experiments demonstrates the effect of EIPSO.

Key words: experience self-improve; particle swarm; optimizer

摘 要: 经验自举粒子群优化算法(EIPSO)是在粒子群算法中引入经验自举(EI)搜索算子, 该算子的作用就是将随机选择的粒子个体经验的局部重新初始化构成候选经验。根据候选经验和原经验的适应值确定个体的新经验。在粒子进化的每一代, 以概率 p 来执行经验自举搜索, 以概率 $1-p$ 执行经验指导下的进化搜索。EI 算子的引入使粒子的搜索范围和多样性得到保持, 同时在粒子收敛后算法仍然具有一定的搜索能力。对比实验结果表明该 EIPSO 算法的良好的综合性能。

关键词: 经验自举; 粒子群; 优化

DOI: 10.3778/j.issn.1002-8331.2008.31.025 **文章编号:** 1002-8331(2008)31-0087-03 **文献标识码:** A **中图分类号:** TP18

1 引言

粒子群算法最初是由 R.C.Eberhart 和 J.Kennedy 与 1995 年提出^[1-2]。该算法将 D 维空间中的一个点看作粒子。若干个粒子的集合称为粒子群。每个粒子记忆各自的个体经验($pbest$)和速度(v)。粒子群中最好的经验称为群体经验($gbest$)。

在标准粒子群优化算法(SPSO)中^[3], 粒子的位置和速度按照如下的公式进行更新:

$$v_i^k \leftarrow w * v_i^k + c_1 * r1 * (pbest_i^k - x_i^k) + c_2 * r2 * (gbest_i^k - x_i^k) \quad (1)$$

$$x_i^k \leftarrow x_i^k + v_i^k \quad (2)$$

在更新公式中, x_i^k 表示第 i 个粒子第 k 维位置值。 v_i^k 表示第 i 个粒子第 k 维速度值。 c_1 和 c_2 为学习因子, 表示个体经验和群体经验对粒子搜索的影响大小。 $r1$ 和 $r2$ 是一个在区间 $[0, 1]$ 上服从均匀分布的随机数。

由于粒子群算法易于实现, 因此在函数优化中得到广泛应用。但是粒子群算法存在局部收敛的问题。这种局部收敛在优化多峰函数中更加明显。

2 经验自举搜索

在标准粒子群算法中, 粒子在个体经验和群体经验指导下

进行搜索。当经验长时间没有更新, 加上粒子速度的衰减, 整个粒子群将很容易被拉到经验附近, 从而失去进一步搜索的能力。因此粒子经验在粒子群算法中起重要的导向作用。

粒子的个体经验可能包含近似合理的部分, 也可能包含不合理的部分。粒子在个体经验和群体经验的指导下不断进行迭代搜索, 更新个体经验和群体经验。在这个过程中, 粒子个体经验局部在进化的同时, 局部也可能会退化。在个体经验各维异步趋于雷同的过程中, 当所有经验的某一维趋于一致时, 粒子群搜索范围将减小, 从而加速了粒子群的早熟收敛。如果能够发挥经验的能动性, 让经验具有自我提高的能力, 即能保留合理的部分, 也能改善不合理的部分, 粒子群的搜索能力将会增强。

经验自举操作通过将经验的局部重新初始化, 如果局部初始化后得到的候选经验的适应值优于原经验的适应值, 那么就可以用候选经验更新该粒子的个体经验。这样就可以克服由于某一维早熟而失去在该维进一步搜索的能力。为此引入算子 $EI, EM(pbest, k) = (pbest^1 \cdots r \cdots pbest^p)$, 式中 $pbest$ 是粒子的经验, r 是在粒子经验第 k 维取值区间上服从均匀分布的随机数。算子 EI 的作用就是将粒子经验的第 k 维进行重新初始化。为增大经验改善的概率, 可以在该维取值区间上取 m 个随机数

作者简介: 徐明亮(1973-), 男, 讲师, 博士生, 从事进化计算、机器学习研究; 须文波, 教授, 博士生导师, 从事智能控制、进化计算研究; 何胜(1972-), 男, 讲师, 博士生, 从事生物信息学、进化计算研究。

收稿日期: 2007-12-06 **修回日期:** 2008-03-17

产生 m 个候选经验, 根据 m 个候选经验的适应值和原经验适应值选取最优的作为该粒子的经验。这种搜索称为经验自举搜索。在进化的每一代, 以概率 p 来选择执行粒子搜索, 以概率 $1-p$ 执行经验自举。经验自举搜索算法流程如图 1 所示。

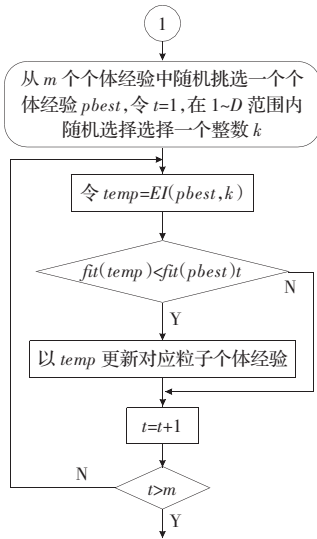


图 1 经验自举搜索

p 的选择对算法的性能有很大的影响, 通过实际实验比较, p 取 0.65~0.7 是效果较好。

EIPSO 算法流程图如图 2 所示。其中, rand 为 0~1 间的随机数。

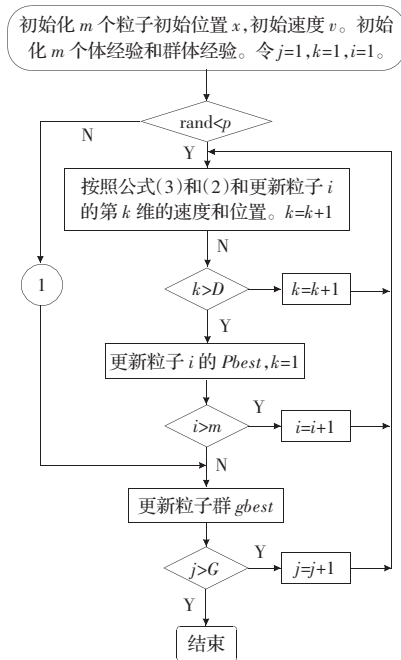


图 2 EIPSO 算法流程图

3 测试函数和参数设定

在实验中, 选取的测试函数如下:

(1) Ackley's function

$$f_1(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) + \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$$

$$x_i \in [-32.768, 32.768]^n$$

(2) Griewank's function

$$f_2(x) = \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, x_i \in [-600, 600]^n$$

(3) Weierstrass function

$$f_3(x) = \sum_{i=1}^n \left(\sum_{k=0}^j [a^k \cos(2\pi b^k(x_i + 0.5))] \right) - n \sum_{k=0}^j [\cos(2\pi b^k \cdot 0.5)]$$

$$a=0.5, b=3, j=20, x_i \in [-0.5, 0.5]^n$$

(4) Rastrigin's function

$$f_4(x) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10), x_i \in [-5.12, 5.12]^n$$

(5) Noncontinuous Rastrigin's function

$$f_5(x) = \sum_{i=1}^n (y_i^2 - 10 \cos(2\pi y_i) + 10)$$

$$y_i = \begin{cases} x_i & |x_i| < 0.5 \\ \text{round}(2x_i)/2 & |x_i| \geq 0.5 \end{cases} \text{ for } i=1, 2, 3, \dots, n$$

$$x_i \in [-5.12, 5.12]^n$$

(6) Schwefel's function

$$f_6(x) = 418.9829 \times n - \sum_{i=1}^n x_i \sin(|x_i|^2), x_i \in [-500, 500]^n$$

这些测试函数都是 PSO 研究中经常使用的多峰函数。所有函数的最优解为 0, 除 Schwefel's function 最优解为 [420.96, ..., 420.96], 其余函数最优解为 [0, ..., 0]。将应用 EIPSO 算法和文献[3]中的 SPSO 算法, 文献[4]中的 UPSO 算法、文献[5]中 wFIPSO 的算法以及文献[6]中 CLPSO 的算法对上述函数进行优化。

在优化 5 维函数时, 粒子数目取 10, 进化代数取 1500。在优化 10 维函数时, 粒子数目取 10, 进化代数取 4000。

经验自举算法中 m 取值和粒子数保持一致, 以保证每一代进化中适应值计算次数相等。其它的参数参照 SPSO 的选择和设定。其它算法的参数参照相应文献所述设定。当粒子越界时, 都作重新初始化处理。

所有测试函数运行 25 次, 取 25 次的平均值和方差进行比较。

4 实验结果与讨论

表 1 所示是上述函数在维数取 5 维情况下的优化结果(黑体表示最优结果)。对于 Ackley's 函数, 除 UPSO 算法结果较差外, 其它算法结果相近, EIPSO 仅次于最优的结果。对于 Griewank's 函数, CLPSO 算法结果最好, 其它算法结果基本相近。对于 Weierstrass function, Noncontinuous Rastrigin's 函数和 Schwefel's 函数, EIPSO 都能够找到最优解, 部分结果优于其它算法。

表 2 所示是上述函数在维数取 10 情况下的优化结果(黑体表示最优结果)。对于 Ackley's 函数, 除 UPSO 算法结果较差, SPSO 性能下降较大外, 其它算法结果相差不大。对于 Griewank's 函数, 所有算法结果基本相近。对于 Weierstrass 函数, CLPSO 算法能取得最好的解。但是 EIPSO 算法性能优于 SPSO、UPSO 算法和 wFIPSO 算法。对于 Rastrigin's 函数, Non-

表 1 5 维测试函数优化结果

算法	测试函数 1	测试函数 2	测试函数 3	测试函数 4	测试函数 5	测试函数 6
EIPSO	2.561E-15	4.52E-02	0.00E+00	0.00E+00	0.00E+00	4.55E-14
	±1.37E-16	±7.67E-02	±0	±0	±0	±1.24E-15
CLPSO	2.98E-15	4.97E-03	0.00E+00	3.30E-02	2.67E-01	1.97E+01
	±1.12E-16	±5.34E-02	±0	±1.86E-03	±6.64E-01	±2.24E+02
SPSO	2.00E-15	5.25E-02	2.12E-04	2.95E+00	2.17E+00	6.30E+02
	±2.54E-16	±2.74E-02	±5.68E-02	±3.11E+00	±4.67E+00	±3.57E+02
UPSO	6.58E-02	8.72E-02	5.47E-02	1.82E+00	1.23E+00	1.93E+02
	±4.26E-02	±1.82E-01	±6.34E-02	±5.34E+00	±6.57E+00	±7.24E+02
wFIPS	2.98E-15	3.73E-02	3.12E-03	1.99E-01	1.00E-01	3.15E+01
	±1.06E-16	±9.33E-02	±1.47E-04	±4.87E-01	±3.21E-01	±2.43E+0

表 2 10 维测试函数优化结果

算法	测试函数 1	测试函数 2	测试函数 3	测试函数 4	测试函数 5	测试函数 6
EIPSO	3.97E-15	7.04E-02	1.00E-03	0.00E+00	0.00E+00	2.10E-13
	±1.34E-16	±2.56E-02	±3.00E-01	±0	±0	±3.00E-01
CLPSO	3.84E-15	6.11E-02	0.00E+00	1.19E-01	5.20E-01	3.79E+01
	±1.48E-16	±1.13E-02	±0	±3.16E-03	±8.57E-02	±3.36E+01
SPSO	3.00E-10	9.4E-02	3.64E-01	1.16E+01	6.96E+00	1.51E+03
	±2.65E-10	±6.77E-02	±2.75E-01	±5.34E+00	±4.75E+00	±9.87E+02
UPSO	9.80E+00	9.12E-02	9.27E-01	1.04E+00	4.00E+00	8.98E+02
	±6.14E+00	±8.46E-02	±3.47E-01	±2.11E+00	±6.48E+00	±6.11E+02
wFIPS	3.55E-15	3.75E-02	2.73E-02	1.70E+00	2.00E+00	1.57E+02
	±2.197E-15	±1.36E-02	±4.92E-02	±8.21E+00	±3.44E+00	±4.82E+02

注: 黑体表示最优结果。

continuous Rastrigin's 函数和 Schwefel's 函数, EIPSO 也都获得了比其它算法好的结果。随着函数维数的增加, 粒子群算法性能都将下降, 早熟现象愈加严重。对于经验自举粒子群算法而言, 经验的每一维要获得更多的自我提高的机会也将减少, 所以收敛到全局最优的速度将会减慢。

5 结论

在粒子群算法中增加了一种新的搜索方法, 即经验自举搜索。这种搜索方法使经验的每一维在任何时候都有机会在搜索空间内进行全局搜索, 在粒子早熟的时候仍然能够保持一定的搜索能力。EIPSO 算法在进化的每一代, 经验自举搜索和粒子在个体经验和群体经验指导下的搜索按照一定的概率选择执行, 一来可以使粒子的速度不至于过快衰减, 使粒子在较长的代数内保持搜索能力, 同时也使粒子在早熟后然而能保持一定的搜索能力。实验结果表明, 经验自举搜索具有良好的综合性能。

另外, 引进经验自举搜索保持了粒子群算法的简单, 易实现的优点。在运算时间上和标准粒子群算法非常接近, 没有增

加原算法的时间复杂度。经验自举搜索算子也很容易嵌入到其他改进的 PSO 算法中。

参考文献:

- [1] Eberhart R C, Kennedy J. A new optimizer using particles swarm theory[C]//Proc 6th Int Symp Micromachine Human Sci, Nagoya, Japan, 1995: 39-43.
- [2] Kennedy J, Eberhart R C. Particle swarm optimization[C]//Proc IEEE Int Conf Neural Networks, 1995: 1942-1948.
- [3] Shi Y, Eberhart R C. A modified particle swarm optimizer[C]//Proc IEEE Congr Evol Comput, 1998: 69-73.
- [4] Parsopoulos K E, Vrahatis M N. UPSO—a unified particle swarm optimization scheme[J]. Lecture Series on Computational Sciences, 2004: 868-873.
- [5] Mendes R, Kennedy J, Neves J. The fully informed particle swarm: simpler, may be better[J]. IEEE Trans Evol Comput, 2004, 8: 204-210.
- [6] Liang J J, Qin A K, Suganthan P N, et al. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions[J]. IEEE Trans Evol Comput, 2006, 10(3): 281-294.