基于博弈模型的合作式粒子群优化算法①

张睿哲[1], 杨照峰[2]

[1](平顶山学院 计算机科学与技术学院, 平顶山 467002)
[2](平顶山学院 软件学院, 平顶山 467002)

摘　要: 粒子群算法作为一种新兴的进化优化方法, 能够大大减轻复杂的大规模优化问题的计算负担. 根据博弈论的思想, 在传统粒子群基础上提出了一种基于博弈模型的合作式粒子群优化算法, 算法基于重复博弈模型, 在重复博弈中利用一个博弈序列, 使得每次博弈都能够产生最大效益, 并得到了相应博弈过程的纳什均衡. 通过典型基准测试函数对算法的性能进行对比实验, 实验结果表明算法是可行的、有效的, 对拓展粒子群算法研究具有重要的理论意义与实际意义.

关键词: 非合作博弈; 动态博弈; 纳什均衡; 粒子群优化算法; 进化优化

Improved Particle Swarm Optimization Algorithm Based on Game Model

ZHANG Rui-Zhe[1], YANG Zhao-Feng[2]

[1](Computer Science and Technic Academy Department, Pingdingshan University, Pingdingshan 467002, China)
[2](School of Software Engineering, Pingdingshan University, Pingdingshan 467002, China)

Abstract: Particle Swarm algorithm as a new evolutionary optimization method can greatly reduce the computational burden of complex, large-scale optimization problems. This article is based on game theory. On the basis of the particle swarm it proposeda non-cooperative game model based on particle swarm optimization algorithm, it wses a game sequence repeated game model. And in repeated games, each game all hope to produce maximum benefits. Nash equilibrium of the corresponding game process. Function through multiple benchmarks, comparing with the performance of the algorithm experimental results show that the algorithm is feasible and effective. The study has important theoretical significance and practical significance onexpand swarm intelligence algorithm.

Keywords: non-cooperative game; dynamic game; nash equilibrium; particle swarm optimization; evolutionary optimization

粒子群优化(Particle Swarm Optimization, PSO)算法是近年发展起来的一种群体智能计算方法[1]. 最先由美国的心理学家 James Kennedy 和电气工程师 Russell Eberhart 共同于 1995 年提出, 之后 Eberhart 等提出了 PSO 的离散二进制版[2]. 粒子群算法收敛速度快、运算简单、所需参数少, 但是该算法全局搜索能力弱, 易于陷入局部最优[3-4]. 为改善其全局寻优能力, Shi 等人在速度迭代公式中引入惯性权重, 另外许多学者还将粒子群算法与其它进化技术相结合, 例如梯度下降、均值聚类等[5]. 使得混合 PSO 成为粒子群算法改进研究的热点[6]. 近年来, 粒子群算法得到越来越多的研究, 并被成功地应用于解决一些像人工神经网络训练、参数优化、故障诊断以及其他工程领域优化问题[7]. 但是在解决一些多峰问题时粒子群算法仍然容易陷入局部最优. 这主要是因为除了参数对 PSO 算法的寻优性能有影响外, 进化策略对进 PSO 算法的寻优性能同样也有着很大的影响.

博弈论(Game Theory)是研究两人或多人谋略和决策问题的理论[8]. 根据决策者行为方式不同, 博弈可分为合作博弈与非合作博弈, 两者的根本区别在于是否有一个有约束力的协议. 博弈理论的任务是解决冲突问题, 以优化博弈参与者的目标函数. 优化实现

① 基金项目:河南省科技计划(102102210416)
收稿时间:2013-10-28;收到修改稿时间:2013-12-09

的过程在合作博弈中是求解全局最优策略的问题; 这点使博弃论成为一种处理具有复杂优化要求问题的有效工具. 本文根据博弈论的思想, 在传统粒子群的基础上提出了基于博弈模型的合作式粒子群优化算法.

1 简介

粒子群算法与其它进化类算法相似, 也采用“群体”与“进化”的概念, 同样也是依据个体(粒子)的适应值大小进行操作. 所不同的是, 粒子群算法不像其它进化算法那样对于个体使用进化算子, 而是将每个个体看作是一个微粒[9-10]. 在粒子群算法中, 群体中的一个微粒表述为每一个可能产生的解, 令 X 表示其中每个粒子所处的位置. 微粒通过不断调整自己的位置 X 来搜索新解. P_i 表示每个微粒都能记住自己搜索到的最好解, P_g 表示目前搜索到的全局最优解. V 表示每个粒子都有一个速度, 设微粒 i 的当前位置为 $\mathbf{X}_i = (x_{i1}, x_{i2}, \cdots, x_{in})$; 微粒 i 的当前飞行速度为 $\mathbf{V}_i = (v_{i1}, v_{i2}, \cdots, v_{in})$; 微粒 i 所经历的最好位置为 $\mathbf{P}_i = (p_{i1}, p_{i2}, \cdots, p_{in})$.

$$V_{id}(t+1) = \omega V_{id}(t) + \eta_1 rand()(P_{id}(t) - X_{id}(t)) + \eta_2 rand()(P_{gd}(t) - X_{id}(t)) \quad (1.1)$$

其中 $rand()$ 为随机数生成函数, V_{id} 表示第 i 个粒子第 d 维上的速度, w 为惯性权重, η_1, η_2 为调节 P_{id} 和 P_{gd} 的参数. 粒子的下一位置:

$$X_{id}(t+1) = X_{id}(t) + V_{id}(t+1) \quad (1.2)$$

2 基于博弈模型的合作式粒子群算法

2.1 基于博弈模型的合作式策略

为不失一般性, 本文以求函数的最小值为例, 下式始终成立.

$$F_{optimal}(t+1) \le F_{optimal}(t) \quad (2.1)$$

其中 t 为迭代次数, F 为最优值的函数. 合作博弈理论中, 参与者只根据他们的可察觉的自我利益来决策. 博弈的次序应以保证每个参与博弈的博弈方都有同等的机会而轮流进行为基本要求. 对于一个二人博弈模型而言, 分别用“A”和“B”代替两个博弈方, 当“A”首先进行策略选择时, 它会选择使自己的收益尽可能大的策略, 然后“B”可根据自身收益来考虑是接受还是拒绝由“A”提出的策略, 如果 B 不受益, B 就不接受, 如果 B 受益, B 就接受, 并且 B 自动获得下一轮的优先选择权. 那么博弈方 B 的动作定义如下:

方式(1): B 拒绝, 并由“A”重新提议.

方式(2): B 接受, 并获得下一轮选择的优先权;

如果 B 采取方式(2), 那么继续由“A”提议, 设此时“B”获益的概率为 Q, 得到的目标函数最优值为 F_2, 受损的概率为 $1-Q$, 得到的目标函数最优值为 F_3. 如果“B”采取方式(1), 那么将保持“A”的收益并在下一轮博弈中保证自己受益, 并设 F_1 为得到的目标函数最优值.

下面证明无论何种情况下博弈方都按方式(1)对待其它博弈方的提议时, 博弈方所获得的收益期望最大.

证明 1: 在最小化问题中有 $F_1 < F_3$, 且方式(2)获得的收益期望 E_2 为 $Q \times F_2 + (1-Q) \times F_3$, 方式(1)获得的收益期望 E_1 为 $1 \times F_1 = F_1$.

$$\begin{aligned} E_1 - E_2 &= F_1 - Q \times F_2 - (1-Q) \times F_3 \\ &= F_1 - F_2 + (1-Q) \times (F_2 - F_3) \end{aligned} \quad (2.2)$$

而(2.1)式确定了“A”的收益, 因此可以肯定 $Q = 0$, 而 $F_1 < F_3$, 从而 $E_1 - E_2 < 0$.

证明 2: 在某些情况下, 会出现 $F_1 > F_2$, 假设这种情况出现的概率为 P, F_2' 为获得的收益期望, 这个时候 $E_2 = Q \times F_2 + P \times F_2' + (1-Q-P) \times F_3$, P 趋向于 0. 由公式(2.1)可知 $F_1 - F_3 < 0$, 因此无论何种情况下博弈方都按方式(1)对待其它博弈方的提议时, 博弈方所获得的收益期望最大.

收到上述思想的启发, 将每个粒子看成是一个智能体, 假设它能控制种群往自己最有利的方向进行搜索, 然后将它看成是参与博弈的一个参与人. 从而利用博弈模型对基本的粒子群算法进行改进.

2.2 改进算法步骤

step1: 初始化种群粒子;

step2: 使用博弈策略方式(1);

step3: 判断是否满足算法终止条件, 若满足, 则结束算法, 若不满足, 则根据博弈策略方式(1)的进化成功率判断博弈策略方式(1)是否继续使用, 若否, 则选择博弈策略方式(1)使粒子按照公式(1.1)和(1.2)进行进化;

step4: 判断是否满足算法终止条件, 若满足, 则结束算法, 若不满足, 则根据博弈策略方式(2) 的进化成功率判断博弈策略方式(2)是否继续使用, 若否, 则选择博弈策略方式(1)使粒子按照公式(1.1)和(1.2)进行进化;

step5: 判断是否满足算法终止条件, 若满足, 则结束算法, 若不满足, 则根据博弈策略方式(2)的进化成功率判断博弈策略方式(2)是否继续使用, 若否, 则转 step2.

3 实验结果与分析

为验证本文算法的有效性以及可行性, 本文利用 3 个典型的标准测试函数对 GPSO 算法寻优性能进行测试. 对每个测试函数 10 维, 20 维, 和 30 维分别进行测试; 相应的进化代数设置为 1000 代, 1500 代和 2000 代; 种群数设为 20, 40, 80, 160. 在整个进化过程中, 惯性权重的取值区间设置为[0.0175,0.900], 即 ω 在整个进化过程中在区间[0.0175,0.900]内线性递减. 每种策略的最小迭代次数设置为 15, 最大迭代次数设置为 75, θ 设置为 0.05. 三个测试函数如下:

测试函数(1)

$$f_1(x)=\begin{cases} -x & if \quad x\le 1 \\ -2+x & if \quad 1<x\le 3 \\ 4-x & if \quad 3<x\le 4 \\ -4+x & if \quad x>4 \end{cases}$$

$$f_2(x)=(x-5)^2$$

其中 $x\in[-500,500]$;

测试函数(2)

$$f_1(x,y)=x$$

$$f_2(x,y)=(1+10y).[1-(\frac{x}{1+10y})^a-\frac{x}{1+10y}.\sin(2\pi qx)]$$

其中 $x,y\in[0,1], a=2, q=4$

测试函数(3)

$$f_1(x)=x_1$$

$$f_2(x)=g(x)[1-(x_1/g(x))^2]$$

$$g(x)=1+9(\sum_{i=2}^{n}x_i)/(n-1)$$

其中 $x\in[0,1], n=30$

将 GPSO 测试结果与文献[5]中报道的 L-PSO 和基准的 PSO 的结果进行比较. 本文的每次试验都取算法独立运行 50 次的均值. 图 1、2, 表 1、2、3 、4 分别列出了本文 GPSO 算法, 文献[5]中报道的 L-PSO 和基准的 PSO 算法对三个标准测试函数不同维数的测试结果. 从图表中列出结果可以看出, 本文 GPSO 算法的收敛精度和稳定性均要优于 L-PSO 算法和 PSO 算法.

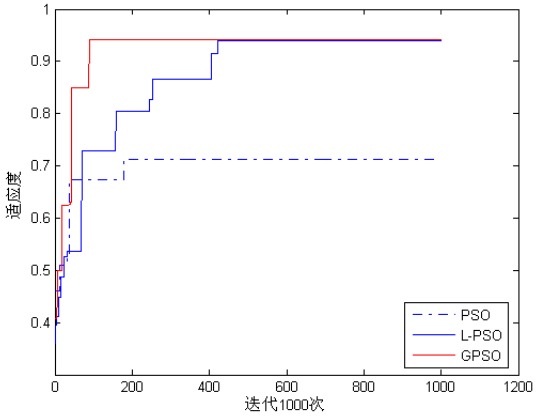

图 1 函数 $f(1)$ 的寻优曲线对比(迭代 1000)

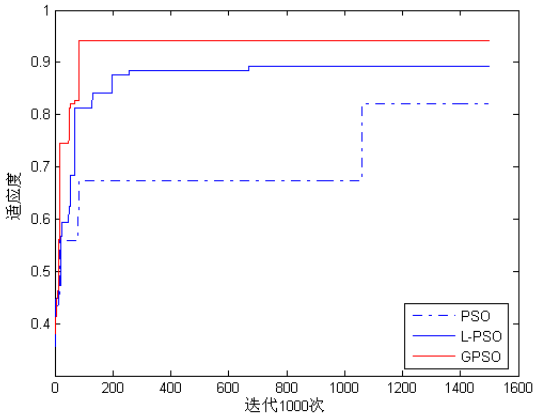

图 2 函数 $f(1)$ 的寻优曲线对比(迭代 1500)

表 1 对函数 $f(1)$ 的寻优性能比较

Size	Dim	Maxiteration	PSO	L-PSO	GPSO
20	10	1000	149.6123±321.0980	52.0489 ± 99.0456	28.5690 ±78.6559
	20	1500	198.21369 ±364.0012	111.0923± 173.9120	79.3124 ±132.9902
	30	2000	271.3690 ±368.3366	131.9045±136.0980	92.8570 ±127.5012
40	10	1000	63.0143 ±117.0067	26.1676 ± 66.1111	12.8902 ±31.1290
	20	1500	169.0146±307.9801	69.9023 ± 67.1123	39.5609 ±45.0012
	30	2000	222.8661±342.3678	87.1767± 63.9032	68.0904 ±88.0101

	10	1000	34.6390±67.8603	17.0687 ±26.5890	12.1112 ±27.3598
80	20	1600	118.1234 ±236.8891	61.0356 ±66.1245	31.8901±37.7809
	30	2000	221.6691 ±392.0031	62.3466 ±69.4503	48.1230 ±43.1290
	10	1000	20.0071 ±39.0073	16.0789 ± 26.9032	11.5501 ±25.4345
160	20	1500	79.0078±140.9901	47.8902 ±46.0023	31.3090 ±33.5566
	30	2000	104.8012±194.9903	42.9969±44.9074	41.8705 ±38.7689

表 2 对函数 $f(2)$ 的寻优性能比较

Size	Dim	Maxiteration	PSO	L-PSO	GPSO
	10	1000	6.0023 ±2.7890	4.6308 ±2.2246	3.9634 ±1.1290
20	20	1500	23.1280 ±6.8997	17.6823 ±6.0784	16.6383 ±6.2874
	30	2000	48.6432 ±11.9803	33.6934 ±9.9045	29.0874 ±9.6662
	10	1000	3.8990±2.0123	2.9692 ±1.0684	2.6022 ±1.9864
40	20	1500	16.6678 ±6.1568	12.7094 ±4.6840	10.6433 ±4.4662
	30	2000	38.8869 ±10.9062	26.9467±7.9067	22.1388 ±7.9704
	10	1000	2.4810 ±1.8034	1.8696 ±1.8903	1.2390 ±1.0227
80	20	1500	13.1076±4.2326	10.6870 ±3.0985	7.6677 ±3.3139
	30	2000	29.3049±7.1234	22.7680 ±6.9065	16.8777±6.0965
	10	1000	1.6337 ±1.1109	1.0668 ±0.9684	0.6709±0.9531
160	20	1500	9.3764 ±3.1098	8.2121 ±2.2916	6.7602 ±3.97365
	30	2000	23.9643± 6.9056	21.4186±6.9499	12.2983 ±6.9036

表 3 对函数 $f(3)$ 的寻优性能比较

Size	Dim	Maxiteration	PSO	L-PSO	GPSO
	10	1000	8.3124 ±2.7026	7.5308±2.0121	4.5160 ±1.8742
20	20	1500	34.0965± 6.2130	21.5823±6.4463	13.6383 ±5.0764
	30	2000	69.6432±11.1707	54.5934±9.0643	24.9992 ±9.0023
	10	1000	5.9407±2.0244	3.9592±1.7284	1.5022 ±1.6876
40	20	1500	19.5678 ±5.5229	16.5039±4.5840	9.6433±4.4562
	30	2000	99.8859 ±10.0956	68.6870±7.7643	21.1388 ±7.0567
	10	1000	6.4810 ±1.3439	5.8696±1.1424	4.1915 ±1.0227
80	20	1500	18.1075 ±4.2326	17.6870±3.0034	6.6577 ±3.3139
	30	2000	69.3049 ±7.0478	56.7680±6.7589	34.8777±6.0934
	10	1000	112.5337± 1.1121	102.0558±0.9584	100.5709 ±0.0845
160	20	1500	16.3754 ±3.0281	13.2121±2.0167	11.7602±3.3048
	30	2000	123.9643±5.7055	120.4185±5.9499	99.1863 ±5.0045

4 结 语

提出了基于博弈论的非合作式粒子群优化算法. 通过典型的基准函数对算法的性能进行对比实验验证, 实验结果表明算法是可行的、有效的. 研究对实时优化问题和复杂优化问题具有一定的应用价值.

参考文献

1 Diosan L, Oltean M. What else is the evolution of PSO telling us. Journal of Artificial Evolution and Applications, 2008, 8(2): 1–12

2 Rana S, Jasola S, Kumar R. A review on particle swarm optimization algorithms and their applications to data clustering. Artificial Intelligence Review, 2011,35(3): 211–222.

3 Liang JJ, Qin AK, Suganthan, PN. Comprehensive learning particle swarm optimizer for global 215 optimization of multimodal functions. IEEE Trans. on Evolutionary Computation, 2006, 10(3): 281–295.

4 Xu B, Guan Q, Chen K. Multi-agent coalition formation based on quantum-behaved particle swarm qptimization. Journal of Information & Computational Science, 2010, 7(5): 1059–1064.

5 Wang Y, Li B, Weise T, et al. Self-adaptive learning based particle swarm optimization. Information Sciences, 2011, 181(20): 4515–4538.

6 田有亮,马建峰,彭长根,姬文江.秘密共享体制的博弈论分析.电子学报,2011,39(12):2790–2795

7 弗登博格,梯若尔著,黄涛译.博弈论.北京:中国人民大学出版社,2010.

8 刘玉岭,冯登国,吴丽辉,连一峰.基于静态贝叶斯博弈的蠕虫攻防策略绩效评估.软件学报,2012, 23(3):712–723.

9 Nickabadi A, Ebadzadeh MM, Safabakhsh R. A novel particle swarm optimization algorithm with adaptive 220 inertia weight. Applied Soft Computing, 2011, 11(4): 3658–3670.

10 胥小波,郑康锋,李丹,武斌,杨义先.新的混沌粒子群优化算法.通信学报,2012,33(1):24–30,3.