

文章编号: 1001-0920(2009)07-1083-04

动态环境下的双子群 PSO 算法

焦 巍, 刘光斌

(第二炮兵工程学院 303 室, 西安 710025)

摘 要: 通过两组搜索方向相反、相互协同的主、辅子群, 构造一种新的双子群粒子群优化算法. 该算法扩展了种群的搜索范围, 充分利用搜索域内的有用信息, 在感知到环境变化时能迅速、准确地跟踪动态变化的极值. 使用 DF1 (Dynamic Function 1) 生成的复杂动态环境对该算法进行了验证, 并与 Eberhart 提出的动态环境下的粒子群优化算法进行了比较分析. 仿真结果表明了该算法的有效性.

关键词: 粒子群优化; 动态环境; 子群

中图分类号: TP18 **文献标识码:** A

Two subpopulation swarm PSO algorithm in dynamic environment

JIAO Wei, LIU Guang-bin

(Unit 303, The Second Artillery Engineering College, Xi'an 710025, China. Correspondent: JIAO Wei, E-mail: jiaowei1981@yahoo.com.cn)

Abstract: Through main subpopulation particle swarm and assistant subpopulation particle swarm, whose searching direction are inversed completely, a two-subpopulation particle swarm optimization algorithm is proposed, which extends the searching rang. Changing extrema can be tracked promptly and accurately when the variable environment is detected. The simulative environment used in these experiments is generated by DF1 (Dynamic Function 1), and the results show that the improved PSO algorithm is more effective and adaptive than the PSO algorithm proposed by Eberhart.

Key words: PSO; Dynamic environment; Subpopulation

1 引 言

粒子群优化算法 (PSO) 是一种种群优化算法^[1]. PSO 算法通过模拟鸟群的捕食行为来达到优化问题的求解, 其简便的操作方法、较强的收敛能力使之自问世以来备受青睐, 并在很多领域的静态优化问题中得到成功应用.

然而, 真实世界中遇到的最优化问题往往是随时间变化的动态环境. 频繁变化的解空间使得最优解随时间的变化而变化, 在当前时刻得到的最优解不一定是下一时刻的最优解. 譬如在物流配送过程中, 受客户优先级、交通环境等因素变化的影响, 物流配送问题也相应发生变化. 所以, 近年来将 PSO 算法应用到动态环境中, 跟踪动态环境的变化已成为一个新的研究领域.

2 动态环境下的 PSO 算法

将 PSO 算法应用到动态环境中, 要求算法必须对复杂变化的动态环境具有较强的适应性. 为了能

跟踪到环境变化后的最优解, 动态环境下的 PSO 算法不仅能检测到环境的变化 (感知能力), 而且在环境变化后能紧密跟踪到解的变化直到获得最优解 (响应能力). 为此, 已有学者做了一些工作. Eberhart^[2] 提出通过监测全局最优解和次全局最优解的方法来检测环境的变化; 而 Carlisle^[3] 则提出随机选择一个或多个“sentry”粒子作为检测环境变化的感应粒子. 对于响应问题, 两者都是以种群为基本对象进行重新初始化处理. 这两种方法堪称动态环境下 PSO 算法的经典, 但是它们有共同的不足: 1) 不断地重新初始化会使粒子丧失“记忆”功能, 特别是在搜索的初始阶段, 频繁的初始化使得算法不能快速收敛于最优解; 2) 计算负担加重.

随后, 更多的学者在此领域开展研究, 比如 Blackwell^[4] 研究发现多量子群 PSO 算法, 在检测到环境变化时更能快速跟踪到变化的极值; Du 等^[5] 提出的多策略集合 PSO 算法, 同样通过多子群的协同

收稿日期: 2008-09-09; 修回日期: 2008-11-12.

作者简介: 焦巍 (1981 →), 男, 陕西临潼人, 博士生, 从事先进控制理论及应用、智能优化的研究; 刘光斌 (1963 →), 男, 四川射洪人, 教授, 博士生导师, 从事先进控制、智能仪器和系统可靠性与电磁环境控制等研究.

对环境的动态变化作出有效响应.单世民^[6]引入活性因数,提高了粒子群监测环境变化的能力;而胡静等^[7]则通过检测种群多样性,感知外部环境的变化,扩展了粒子群的分布.

其实,PSO算法之所以不能有效响应动态环境的变化,主要原因是该算法本身易陷入局部最优.因此,本文提出一种新的双子群PSO算法,扩展搜索范围,在检测到环境变化时能准确、快速地跟踪到极值点的变化.

3 双子群 PSO 算法

在引入本文改进算法之前,先给出标准PSO算法的速度-位置进化方程如下:

$$v_{id}(t+1) = v_{id}(t) + c_1 \text{rand}_1 [p_{id}(t) - x_{id}(t)] + c_2 \text{rand}_2 [p_{gd}(t) - x_{id}(t)], \quad (1)$$

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

其中: $i = 1, 2, \dots, m$; $d = 1, 2, \dots, D$; m, D 分别为粒子群规模和搜索空间的维数; c_1, c_2 是非负常数; $\text{rand}_1, \text{rand}_2$ 是介于 $[0, 1]$ 之间服从均匀分布的随机数; $P_i = (p_{i1}, p_{i2}, \dots, p_{id})$ 是粒子 i 当前所经历的最优位置,称为个体最优位置; $P_g = (p_{g1}, p_{g2}, \dots, p_{gd})$ 是群体中所有粒子所经历的最优位置,称为全局最优位置.

为方便书写,将本文给出的双子群PSO算法简记为TSPSO.其实现思想是:在随机初始化一组粒子群之后,将其等分为两个相互独立的子群,其中一组子群按照标准PSO算法迭代搜索,称为主子群;另一组子群沿主子群的相反方向迭代搜索,位置进化方程按下式更新:

$$x_{id}(t+1) = x_{id}(t) - v_{id}(t+1), \quad (3)$$

该子群称之为辅子群.在每次迭代结束时,比较两个子群个体最优位置所对应的适应值大小,适应值较差的个体最优粒子被更优者替代.即在进行下一次迭代时,两个子群从共同的当前最优位置出发,向两个相反方向搜索全局最优值.这样,通过主、辅两个子群相互补充、协同进化,在不增加粒子规模的情况下,充分扩展搜索范围,发掘搜索域内的有用信息,从而降低PSO算法陷入局部最优点的风险.正因如此,TSPSO算法能够在检测到环境变化时,迅速、准确地跟踪动态变化的极值,而不会导致种群因陷入局部最优而进化停滞.

TSPSO算法通过监测主子群的全局最优解和次全局最优解的方法来感知环境的变化,这一点和Eberhart提出的方法相同.当感知到环境发生变化时,TSPSO算法对辅子群重新初始化,而保持主子群的“记忆”性,只对主子群重新估计适应值,并更新两个子群的 P_i, P_g .同时,为了尽可能地扩展粒子

群在动态环境中的搜索范围,降低陷入局部最优点的风险,TSPSO算法对粒子最大速度不加限制.

该算法的主要计算流程如下:

Step1: 随机初始化粒子群中粒子的位置与速度.

Step2: 将随机初始化的粒子群等分为两组子群,计算每个粒子的适应值,设置 P_i 为初始群体的当前位置, P_g 为最优粒子位置.

Step3: 判断算法收敛准则是否满足,如果满足,则执行Step10;否则,转Step4.

Step4: 主子群根据式(1)和(2)更新粒子的位置与速度,辅子群按式(1)和(3)更新粒子的位置与速度.

Step5: 对主、辅子群进行位置限制.当 $X_i > X_{\max}$ 或 $X_i < X_{\min}$ 时,令 $X_i = X_{\max}$ 或 $X_i = X_{\min}$. X_{\max} 和 X_{\min} 分别表示粒子搜索范围的最大、最小位置限制.

Step6: 更新主、辅子群个体最优位置.如果粒子的适应值优于 P_i 的适应值,则 P_i 更新为新位置;反之, P_i 保持不变.

Step7: 更新主、辅子群全局最优位置.如果粒子的适应值优于 P_g 的适应值,则 P_g 更新为新位置;反之, P_g 保持不变.

Step8: 比较两组子群的个体最优位置所对应的适应值,优者更新为两组子群共同的个体最优位置.同样,比较两组子群的全局最优位置所对应的适应值,优者更新为整个粒子群的全局最优位置.

Step9: 每 n 次迭代监测一次全局最优解和次全局最优解是否发生变化,如果没有变化,则重新评估主子群的适应值、重新初始化辅子群,算法转向Step4;反之,算法继续执行.

Step10: 判断算法收敛准则是否满足,如果满足,则输出全局最优粒子 P_g 和最优适应值,算法运行结束;否则,转Step4.

4 实验设计与结果分析

为了测试上述算法在动态环境下追踪变化最优解的能力,在DF1产生的二维环境下进行了一系列实验测试.DF1函数是由Morrison等^[8]提出的,它通过利用一些锥体的组合产生复杂的环境.DF1函数定义如下:

$$f(X, Y) = \max_{i=1, \dots, N} (H_i - R_i \sqrt{(X - X_i)^2 + (Y - Y_i)^2}). \quad (4)$$

其中: $f(X, Y)$ 是在 (X, Y) 位置的适应值, N 是环境中锥体的个数, (X_i, Y_i) 是第 i 个锥体的顶点位置, H_i 是第 i 个锥体的高度, R_i 是第 i 个锥体的斜度参

数. 式(4) 说明, 在搜索空间的适应值曲面上任一点的取值可由一个最大化函数决定. 图 1 是由 DF1 函数随机生成的一个 3 锥体适应度曲面.

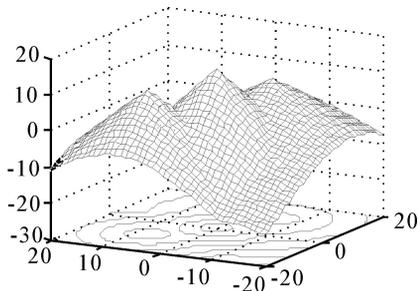


图 1 由 DF1 生成的 3 锥体适应度曲面

4.1 实验环境设置

为了直观说明 TSPSO 算法在复杂动态环境下的适应能力, 在由 DF1 函数生成的二维动态环境中进行了多个实验, 并在同样的实验条件下与 Eberhart 提出的 PSO 算法(记为 E-PSO 算法) 进行了比较. 实验中算法的目标是寻找环境中的最大值.

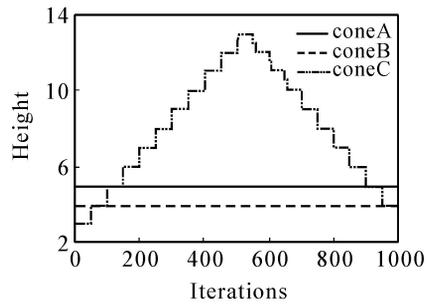
表 1 环境参数设置

名称	动态锥体个数	静态锥体个数	高度变化范围	锥体位置是否变化
实验 1	1	2	[3, 13]	是
实验 2	2	1	[- 6, 13]	否
实验 3	2	1	[- 6, 13]	是

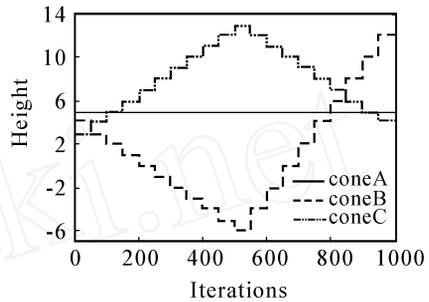
实验由 DF1 生成 3 个锥体的典型环境, 锥体分别记为 A, B, C. 在实验 1 的环境中, 锥体 A, B 的高度和位置保持不变, 锥体 C 的高度和位置则是动态变化的. 实验 2 的环境则设置为只有锥体 A 的高度和位置保持不变, 锥体 B, C 的高度是动态变化的, 但 3 个锥体的位置不变. 在实验 3 的环境下, 锥体 B, C 的高度和位置均是动态变化的. 实验环境参数设置见表 1, 3 组实验的锥体高度 H_i 变化如图 2 所示. 实验中锥体 A, B, C 的初始高度分别设定为 5, 4, 3. 在实验过程中, 动态变化的锥体每经过 50 次进行一次变化更新, 即在每次独立的进化过程中, 每经过 50 次迭代锥体变化一次. 每组实验中的锥体斜度参数 R_i 随机生成并保持不变.

4.2 算法参数设置

在所有实验中, 粒子群的搜索范围被限制在 $X, Y \in [-10, 10]$, $c_1 = c_2 = 1.494$, 最大迭代次数为 1000. 因为在动态环境中, 不能确定迭代过程中种群应该何时加强探索能力, 何时加强开发能力, 所以惯性权值因子采用随机取值, 即 $w = 0.5 + \text{rand}/2$. TSPSO 算法的每个子群规模为 10, 为保证比较的公平性, E-PSO 算法的种群规模取为 20. 两种算法都在每迭代 20 次时检测一次环境是否发生变化, 当环



(a) 实验1的结果



(b) 实验1和实验2的结果

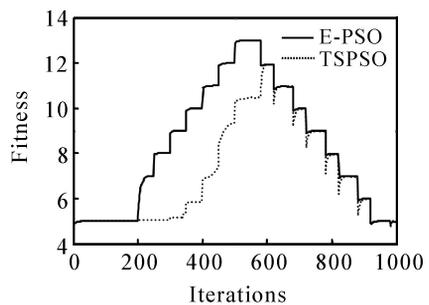
图 2 锥体高度变化

境发生变化时 E-PSO 算法的种群初始化规模取文献[2] 中给出的最优值 10%. 两种算法分别独立运算 50 次, 允许误差设定为 10^{-6} .

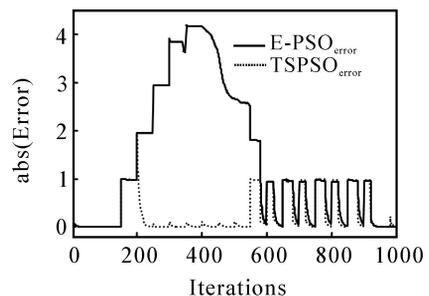
4.3 结果分析

将每组实验 50 次运算结果取平均值, 得到两种算法跟踪极值点的过程和种群所得最优值与锥体实际最大值的绝对误差, 分别如图 3 ~ 图 5 所示.

在实验 1 的 3 个锥体中, 只有锥体 C 的高度和位置发生动态变化. 从图 3 可以明显看出, 相形之下,

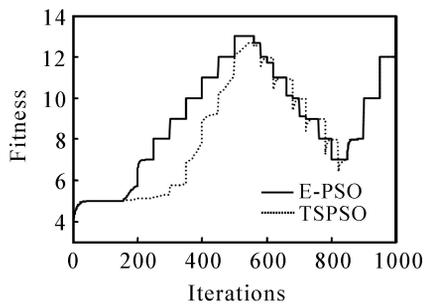


(a) 跟踪变化极值过程

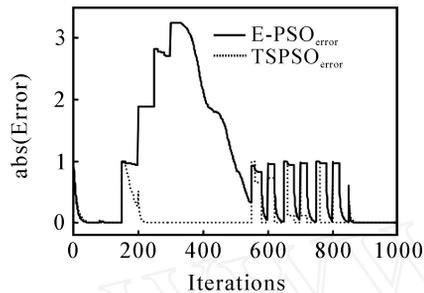


(b) 误差变化过程

图 3 实验 1 的仿真结果

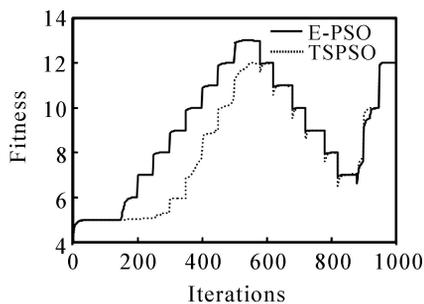


(a) 跟踪变化极值过程

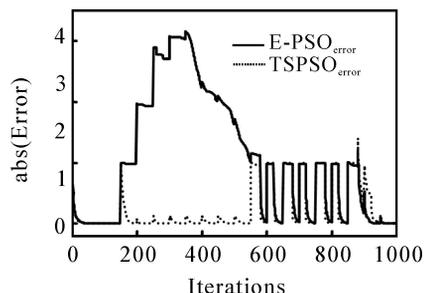


(b) 误差变化过程

图4 实验2的仿真结果



(a) 跟踪变化极值过程



(b) 误差变化过程

图5 实验3的仿真结果

传统的算法的跟踪响应速度慢、寻优精度差,尤其表现在前500次的迭代过程中.整个实验过程的第200次迭代是个分水岭,TSPSO算法在前200次迭代过程中与E-PSO算法效果相近;但在此之后,TSPSO算法便能迅速跟踪不断变化的极值点,误差也控制在一个较小的范围内.

实验2的动态环境设置为一个静态锥体A和两个高度变化的锥体B,C.实验初始时,锥体A的顶点

是极值点,从第200次迭代开始,锥体C的顶点为极值点并保持到第800次迭代才被幅值增长更大的锥体B的顶点所取代.同样,E-PSO算法在迭代的前半程始终无法及时跟踪到剧烈变化的极值点.而TSPSO算法在200次迭代之后就能准确跟踪上变化的极值,寻优结果误差为零,直到迭代后半期有一段最高锥体高度逐渐变小的过程中,两种算法的误差都有所增大.但TSPSO算法的响应速度比E-PSO算法更快,且随着迭代的进行,TSPSO算法的响应速度不断提高,而E-PSO算法则没有改善.

实验3是在实验2的基础上,使锥体B和C的高度动态变化的同时,其锥体位置也以一定的速度水平移动.在此过程中,TSPSO算法的性能同样好于E-PSO算法,只是此时的动态环境更为复杂.相比实验2,TSPSO算法的寻优精度和响应速度有所下降.

这里需要指出的是,由于本文提出的TSPSO算法在每代进化之后,要进行子群之间的最优适应值比较和信息交换,计算时间比E-PSO算法稍长,但相差不大.以文中实验3为例:在50次独立仿真运算中,TSPSO算法平均每次进化计算耗时0.7459s,而E-PSO算法耗时为0.7194s.这也是“无免费午餐”定理的又一佐证.

5 结论

本文提出了一种新的双子群粒子群(TSPSO)算法,该算法通过两组搜索方向相反的主、辅子群协同搜索,扩展了种群的搜索范围,充分挖掘了搜索域内的有用信息,在环境发生变化时能迅速、有效地跟踪到动态变化的极值.文中设置的几种动态实验环境的仿真结果表明,在复杂的动态环境下TSPSO算法是有效的,且响应速度和寻优精度均好于E-PSO算法.

参考文献(References)

- [1] Kennedy J, Eberhart R. Particle swarm optimization [C]. Proc of IEEE Int Conf on Neural Networks. Perth: IEEE Press, 1995, 4: 1942-1947.
- [2] Hu X H, Russell C Eberhart. Adaptive particle swarm optimization: Detection and response to dynamic systems[C]. Proc of the IEEE Int Conf on Evolutionary Computation. Honolulu: IEEE Press, 2002: 1666-1670.
- [3] Carlisle A, Dozier G. Tracking changing extrema with adaptive particle swarm optimizer [C]. World Automation Congress. Orlando, 2002: 265-270.

(下转第1091页)

依据,需要根据经验反复试验才能确定;将控制信号按有效数字划分节点后,收敛速度受到一定影响;对于算法的稳定性、收敛性等诸多问题还有待于进一步研究。

参考文献(References)

- [1] 宋莹,陈增强,袁著祉. 基于混沌优化的非线性预测控制器[J]. 控制理论与应用, 2007, 24(4): 562-564.
(Song Y, Chen Z Q, Yuan Z Z. A nonlinear predictive controller based on chaos optimization [J]. Control Theory & Applications, 2007, 24(4): 562-564.)
- [2] 关圣涛,楚纪正,邵帅. 粒子群算法在非线形模型预测控制中的应用[J]. 北京化工大学学报, 2007, 34(6): 653-656.
(Guan S T, Chu J Z, Shao S. Application of nonlinear model predictive control based on particle swarm optimization [J]. J of Beijing University of Chemical Technology, 2007, 34(6): 653-656.)
- [3] Vapnik V. The nature of statistical learning theory[M]. New York: Springer, 1999.
- [4] Vapnik V. An overview of statistical learning theory [J]. IEEE Trans on Neural Network, 1999, 10(5): 988-989.
- [5] Drucker H, Burges C J C, Kaufman L, et al. Support vector regression machines [J]. Advances in Neural Information Processing Systems, 1997: 999-1004.
- [6] 王定成,汪懋华. 基于 GA 的 SVMR 预测控制研究[J]. 控制与决策, 2004, 19(9): 1067-1070.
(Wang D C, Wang M H. On SVMR predictive control based on GA [J]. Control and Decision, 2004, 19(9): 1067-1070.)
- [7] Wang J, Sun S Y. Predictive control based on support vector machine model [C]. Proc 6th WCICA. Dalian, 2006: 1683-1687.
- [8] Suykens J A K, Van Gestel T, DE Brabanter J, et al. Least squares support vector machines [M]. Singapore: World Scientific Publishing Co Pte Lte, 2002.
- [9] 张日东,王树青. 基于支持向量机的一类非线性系统预测控制[J]. 控制与决策, 2007, 22(10): 1103-1107.
(Zhang R D, Wang S Q. Predictive control of a class of nonlinear systems based on support vector machine [J]. Control and Decision, 2007, 22(10): 1103-1107.)
- [10] Dorigo M, Maniezzo V, Colomi A. Ant system: Optimization by a colony of cooperating agents [J]. IEEE Trans on Systems, Man, and Cybernetics — Part B, 1996, 26(1): 29-41.
- [11] 高异,杨延西,刘军. 模糊遗传滚动优化的 LS-SVM 预测控制研究[J]. 系统仿真学报, 2003, 15(10): 1494-1496.
(Gao Y, Yang Y X, Liu J. Research on LS-SVM predictive control using fuzzy genetic algorithm rolling optimization [J]. J of System Simulation, 2003, 15(10): 1494-1496.)
- [12] 冯国楠. 现代伺服系统的分析与设计[M]. 北京: 机械工业出版社, 1990.
(Feng G N. Analyse and design of modern servo system [M]. Beijing: Mechanical Industry Press, 1990.)
- [4] Blackwell T, Branke J. Multi-swarm optimization in dynamic environments [C]. Proc of the 2004 Applications of Evolutionary Computing Workshops. Coimbra: IEEE Press, 2004: 489-500.
- [5] Du W L, Li B. Multi-strategy ensemble particle swarm optimization for dynamic optimization [J]. Information Sciences, 2008, 178(15): 3096-3109.
- [6] 单世民,邓贵仕. 动态环境下一种改进的自适应微粒群算法[J]. 系统工程理论与实践, 2006, 26(3): 39-44.
(Shan S M, Deng G S. Improved adaptive particle swarm optimization in dynamic environment [J]. The Theory and Practice of Systems Engineering, 2006, 26(3): 39-44.)
- [7] 胡静,曾建潮,谭瑛. 动态环境下基于种群多样性的微粒群算法[J]. 系统仿真学报, 2007, 19(21): 4932-4935.
(Hu J, Zeng J C, Tan Y. Particle swarm optimization based on diversity of particle in dynamic environments [J]. J of System Simulation, 2007, 19(21): 4932-4935.)
- [8] Morrison R W, De Jong K A. A test problem generator for non-stationary environments [C]. Proc of the Congress on Evolutionary Computation. Piscataway: IEEE Press, 1999: 2047-2053.

(上接第 1086 页)