第36卷 第19期 计 算 机 工 程 2010年10月
Vol.36 No.19 Computer Engineering October 2010
·博士论文· 文章编号：1000—3428(2010)19—0024—03 文献标识码：A 中图分类号：TP18

动态环境下的种群扩散粒子群优化算法

赵传信[1,2]，王汝传[1,3]，季一木[3]
(1. 苏州大学计算机科学与技术学院，江苏 苏州 215006；2. 安徽师范大学数学计算机科学学院，安徽 芜湖 241003；
3. 南京邮电大学计算机学院，南京 210003)

摘 要：传统的粒子群优化算法在优化过程中难以有效地监测环境的动态变化和响应。针对上述问题，通过增加外围监测粒子加强监测有效性，提出一种可以动态响应环境变化的种群多样性扩散函数，在此基础上设计一种扩散粒子群优化算法(DPSO)，在动态环境中与APSO、CPSO进行比较，实验结果表明，DPSO可以更有效地跟踪动态环境下极值的变化并快速收敛。
关键词：粒子群优化算法；多样性；动态环境；扩散

Population Diffuse PSO Algorithm in Dynamic Environment

ZHAO Chuan-xin[1,2], WANG Ru-chuan[1,3], JI Yi-mu[3]
(1. School of Computer Science and Technology, Soochow University, Suzhou 215006, China;
2. School of Mathematics and Computer Science, Anhui Normal University, Wuhu 241003, China;
3. College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

【Abstract】 It is difficult for PSO to detect dynamic change of environment and response in optimizing process. Aiming at the problems, by adding particles which are on the periphery for detecting the change of environment, this paper proposes a new diffuse population function to respond change, and designs an algorithm named Diffuse Particle Swarm Optimization(DPSO). Comparison with APSO and CPSO, it can detect changes of environment more effectively and track with optimum solution faster.
【Key words】 PSO algorithm; diversity; dynamic environment; diffuse

1 概述

粒子群优化算法(PSO)是一种基于群体智能的优化算法，在许多方面与进化算法表现相似。PSO在静态优化问题上表现优秀。但是在现实世界中很多问题往往是动态变化的，如网络环境中的资源调度问题，网络中的任务分配不是固定的，这需要算法能够跟踪最优解变化的轨迹。目前进化算法的一些变种比较适合解决此类动态问题，同样作为智能进化算法的PSO也有可能在这个问题上有较好的表现。近年来，将PSO应用到动态系统中追踪系统的极值变化成为一个新的研究领域。本文研究了如何改进PSO以适应在动态环境下跟踪变化的极值，并提出一种新的改进算法——扩散粒子群优化算法(Diffuse Particle Swarm Optimization, DPSO)。

2 相关工作

PSO具有实现简单、参数较少的优点，在静态优化方面取得许多重要的应用成果。基本的PSO并不能有效解决动态环境下的优化问题，在动态环境中需要解决两方面的问题：(1)如何快速感知当前环境是否改变；(2)在环境改变后如何跟踪变化后的极值。如文献[1]将调整的PSO用于动态环境。文献[2]提出的APSO通过设置sentry粒子监测环境变化来解决上述问题。在响应环境变化后的操作方面，文献[3]采用万有引力模型提出一种定量PSO，模仿电子围绕原子核的运动，把一部分粒子排斥在核心外围来保持粒子的多样性。多种群是保持多样性的有效方法[4]。在复杂动态环境中，不仅要能有效地监测环境的变化，更需要采取有效的措施来快速跟踪最优解。

3 动态环境下改进的PSO

3.1 动态环境下的种群多样性与监测

在动态环境中，算法需要感知环境的变化。Eberhart R C.提出，每次对全局最佳适应值进行重新评价，如果不同，则认为环境出现变化，这样的监测失去了分布式的优点。在复杂环境中，函数往往是多峰的，在粒子群进化一段时间后，种群集中在某一个地区搜索，这时环境的变化可能发生在种群搜索以外的范围，为了有效监测环境变化，必须保持监测粒子的多样性。而种群密度可以反映种群多样性。

定义1 种群密度：表征粒子群个体分散程度的度量值，定义为个体到群体中心的距离，计算公式如下：

$$D(t)=\frac{1}{|N|\times|L|}\times\sum_{i=1}^{|N|}\sqrt{\sum_{j=1}^{M}(x_{ij}(t)-\overline{x_j}(t))^2} \tag{1}$$

其中，$|N|$代表群体规模；$|L|$代表搜索空间的最大对角线长度；M代表搜索空间的维数；$\overline{x}_j(t)$代表第t次迭代时群体的平均中心的第j维分量。显然，这种多样性测度与种群规模及搜索空间的大小无关，能较好地描述个体在解空间中分布的疏密状况。

基金项目：国家自然科学基金资助项目(60773041)；江苏省自然科学基金资助项目(BK2008451)；安徽师范大学校青年基金资助项目(2008xqn48)
作者简介：赵传信(1977－)，男，博士研究生，主研方向：智能优化，无线自组织网络；王汝传，教授、博士生导师；季一木，讲师、博士
收稿日期：2010-03-25 **E-mail：**zcxonline@126.com

为了反映个体与全局最优个体的相对距离，定义个体的核心距离表示个体与全局最优个体的距离。

定义 2 核心距离：表征粒子与最优个体之间的距离，本文采用欧式距离计算：

$$S_j = \sqrt{\sum_{i=1}^{D}(x_{ij}(t) - P_{ig}(t))^2}, j = 1,2,\cdots,N \quad (2)$$

其中， $P_{ig}(t)$ 表示第 t 代全局最优个体第 i 维。监测节点选择核心距离较远的粒子有利于监测环境变化。

种群多样性反映粒子聚集的情况，如果粒子过于聚集，增加监测粒子也难以检测出环境变化，需要定义种群多样性的门限，算法种群的多样性低于门限时需要增加种群多样性。

定义 3 种群多样性门限：执行扩散操作的当前种群密度和初始化种群密度的最小比例 λ。当 $D(t)/D(0) > \lambda$ 时执行扩散操作，其中， $D(t)$ 表示第 t 代种群密度； $D(0)$ 表示初始时种群密度。当群体的多样性低于门限时进行保持多样性的扩散操作。

3.2 保持种群多样性的粒子扩散操作

文献[3]指出，进化算法在动态环境中保持种群的多样性更加重要。文献[5]提出了以染色体空间的最大距离的对偶运算增加多样性。进化计算的原对偶运算是为了解决 0-1 动态优化而定义的一种关键运算。借鉴遗传算法中的对偶运算，本文提出以 PSO 中的扩散操作来增加种群的多样性。PSO 易于在实数空间运算，扩散操作应使粒子尽可能地远离当前位置。本文设计了以种群均值为中心的扩散操作，为了简单起见，假设为一维环境，如图 1 所示，假设群体的中心为 *ave*，离中心较近的粒子向边界扩散，离中心较远的粒子向反向的边界扩散，如图 1 中 $x'_{(t)}$ 表示 $x_{(t)}$ 扩散后的位置。

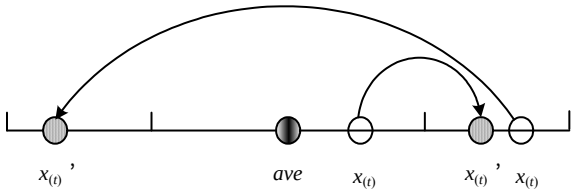

图 1 粒子扩散操作示意图

设粒子在 D 维空间中，为简单起见，对其进行第 i 维 $x \in [d_b, d_t]$ 映射，使其 $x' = dual(x)$，这里 $dual()$ 是扩散函数。

定义 4 粒子扩散操作：每一维以种群平均中心反向扩散粒子：

$$x'_{i(t+1)} = \begin{cases} x_{i(t)} - \dfrac{ave_i - d_b}{2} & ave_i - \dfrac{ave_i - d_b}{2} < x_{i(t)} < ave_i \\ -x_{i(t)} + (d_t - d_b) & d_b < x_{i(t)} < \dfrac{ave_i - (ave_i - d_b)}{2} \\ x_{i(t)} + \dfrac{d_t - ave_i}{2} & ave_i < x_i < ave_i + \dfrac{d_t - ave_i}{2} \\ -x_{i(t)} + (d_b + d_t) & ave_i + (d_t - ave_i) < x_i < d_t \end{cases} \quad (3)$$

$$x_{i(t+1)} = x'_{i(t+1)} + \eta\sigma \quad (4)$$

其中， ave_i 表示种群第 i 维位置的平均值； σ 为均值为 0、方差为 1 的高斯变量的扰动； η 表示限制幅度；对新位置按概率 p_m 变异，本文设变异概率为 50%。PSO 主要应用在实数域上。在一维上最大的距离是边界，但是直接取边界显然失去了多样性。所以，本文设计了一个接近边界的映射。这里的映射是在每一维对外扩散，使个体离开当前位置，反向扩散出去。

3.3 带扩散操作的 PSO

当粒子检测的环境变化或者种群多样性较差时，需要进行多样性操作。本文选择距离核心较远的粒子进行扩展操作。然后按标准 PSO 继续进化，这些粒子在后面的迭代中会很快集中。或者当种群多样性小于门限时，进行扩展操作来自适应地增加多样性，然后全体继续优化。DPSO 算法流程如下：

```
initialize
for i=1 to population size M
 pi=gi
 if f(pi)<f(pg) then g=i endif
 next i
do
if (pg or psentrv Changed)   then
 reevaluate all
 select particle diffuse by formula 3,4
 select   psentry
end if
for i=1 to Pso Number M
 for j=1 to dimension size N
  vij(t+1)= χ {vij(t)+ ξ1  φ1 [pij(t)-xij(t)]+ ξ2  φ2 [pgj(t)-xij(t)]}
  xij(t+1)=xij(t)+vij(t+1)
 next j
 if f(xj(t+1))<f(pj(t+1)) then pi(t+1)=xi(t+1)
 if f(pi(t+1)<f(pg) then g=i;
next i
if D(t)/D(0)< λ  then
 select particle diffuse by formula 3,4
 select psentry
end if
until termination criterion is met
```

算法的基本流程类似于基本 PSO，不同的是，在算法迭代过程中需要利用全局最优个体和监测粒子检测环境变化，一旦环境变化或者种群密度小于门限，需要对选择的粒子按式(2)、式(3)进行扩散操作，作为对环境变化后的响应。假设粒子规模为 M，选择距核心距离最大的 20%的粒子作为监测粒子 p_{sentry}。

3.4 算法性能分析

算法时间复杂度与基本 PSO 相似，设种群规模为 M，迭代次数为 N，忽略函数评价，算法计算的时间复杂度为 $O(MN)$，选择监测粒子的复杂度为 $O(M \log M)$，最坏情况下每次迭代都需要选择监测粒子，则算法的最坏时间复杂度为 $O(NM \log M + MN)$。动态环境下的 PSO 能否保持多样性是衡量动态环境下算法有效性的一个标准。本文引用的种群分布函数可以表示出多样性。进行多样性操作以后种群多样性会更好。

定理 DPSO 算法中扩散操作能够保持种群多样性。

证明：利用式(1)表示种群多样性，进行式(3)的扩散操作后，参与扩散运算的粒子的第 j 维有：

$$(x_{ij}(t+1) - \overline{x_j}(t+1))^2 \geq (x_{ij}(t) - \overline{x_j}(t))^2$$

等式成立当且仅当 $\left|x_{ij} - ave_i\right| > \left|\dfrac{ave_i - border}{2}\right|$，此时，种群多样性较好。

当多样性低于门限时，粒子需要执行扩散操作，则：

$$(x_{ij}(t+1) - \overline{x_j}(t+1))^2 > (x_{ij}(t) - \overline{x_j}(t))^2$$

扩散后的种群多样性为：

$$\frac{1}{|N|\times|L|}\times\sum_{i=1}^{|N|}\sqrt{\sum_{j=1}^{D}(x_{ij}(t+1)-\overline{X}_j(t+1))^2}>$$

$$\frac{1}{|N|\times|L|}\times\sum_{i=1}^{|N|}\sqrt{\sum_{j=1}^{D}(x_{ij}(t)-\overline{X}_j(t))^2} \quad (5)$$

多样性扩散可以重复进行，经过扩散操作后群体的多样性能够得到提高。

4 算法测试与分析

4.1 动态环境的模型

动态环境下的测试主要是通过一些测试函数检验算法性能，为了提供统一的测试环境，文献[6]提出了一种模仿动态环境的 DF 函数，利用锥体产生复杂的环境。静态的二维函数是：

$$f(X,Y)=\max_{i=1..N}[H_i-R_i\cdot\sqrt{(X-X_i)^2+(Y-Y_i)^2}] \quad (6)$$

其中，$f(X,Y)$ 表示适应值；(X_i,Y_i) 表示锥体的位置；R_i 表示斜率；H_i 表示锥体的高度；N 表示锥体的个数。锥体的高度在一定范围内变化，为了进行不同算法的测试，利用动态函数生成场景，动态锥体高度变化范围为[1,10]，其余参数如表 1 所示。

表 1 实验参数

动态环境	动态锥体个数	静态锥体个数	位置变化
1	1	3	是
2	2	2	是
3	4	0	否
4	1	1	否

4.2 实验结果及分析

为了便于比较，选取 2 种目前典型的在动态环境下改进的 PSO，分别为 APSO[1]、CPSO[3]。参数种群规模为 20，$\xi_1=\xi_2=2.05$，$\chi=0.729$，CPSO 的其他参数采用文献[3]中表 4 的设置。DPSO 的多样性阈值取为 0.3，取种群中 50%的粒子进行扩散。所有算法运行 20 次取平均，每次执行迭代 300 次，每隔 50 代变化环境。实验结果如图 2~图 5 所示，其中，横坐标为测试迭代次数；纵坐标为搜索到的最优解与实际最优解的误差。

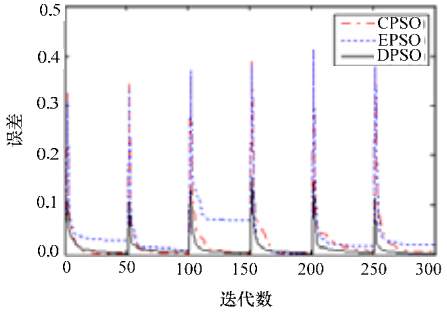

图 2 动态环境 1 下 3 种算法的性能比较

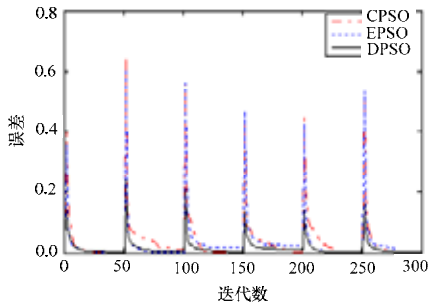

图 3 动态环境 2 下 3 种算法的性能比较

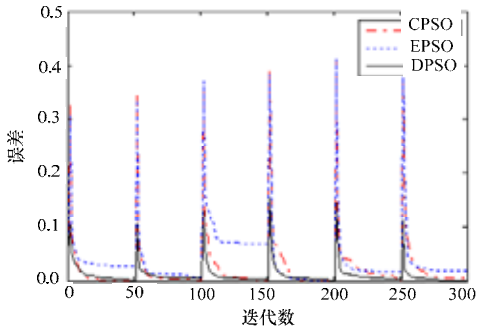

图 4 动态环境 3 下 3 种算法的性能比较

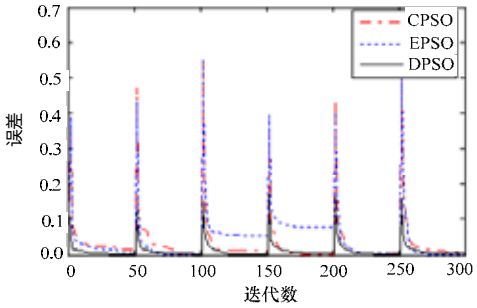

图 5 动态环境 4 下 3 种算法的性能比较

从图中可见，DPSO 的跟踪性能最稳定，CPSO 将一部分粒子置于外围，因此，在迭代过程中表现较为稳定，但处于外围的粒子使算法的收敛性受到影响，不容易快速收敛到环境变化后的解。APSO 在检测到环境变化后选取部分粒子重新初始化，这种初始化存在任意性，可以看出，在环境变化后 APSO 的跟踪性能变化较大。由于 DPSO 中增加了部分粒子检测环境变化，因此检测环境的变化更准确，另一方面，优化的种群规模不变，这样保证了算法的收敛性。另外，DPSO 在不同实验中都能更快速稳定地跟踪环境的变化。

5 结束语

本文设计了一种保持种群多样性的扩散粒子群优化算法，动态测试函数的实验结果表明本文算法是有效的，优于 CPSO 和 APSO。如何将动态环境下的 PSO 应用于实际环境是下一步的研究方向。

参考文献

[1] Hu Xiaohui, Eberhart R C. Adaptive Particle Swarm Optimization: Detection and Response to Dynamic Systems[C]//Proc. of 2001 Congress on Evolutionary Computation. Seoul, Korea: [s. n.], 2001.

[2] Carlisle A, Dozier G. Tracking Changing Extrema with Adaptive Particle Swarm Optimizer[C]//Proc. of WAC'02. Orlando, Florida, USA: [s. n.], 2002.

[3] Blackwell T, Branke J. Multi-swarm, Exclusion, and Anti-convergence in Dynamic Environments[J]. IEEE Trans. on Evolutionary Computation, 2006, 10(4): 459-472.

[4] 焦 巍, 刘光斌. 一种新的双子群 PSO 算法[J]. 计算机工程, 2009, 35(16): 173-175.

[5] Yang Shengxiang. Non-stationary Problem Optimization Using the Primal-dual Genetic Algorithm[C]//Proc. of 2003 Congress on Evolutionary Computation. Canberra, Australia: [s. n.], 2003.

[6] de Jong K A. A Test Problem Generator for Non-stationary Environments[C]//Proc. of 1999 Congress on Evolutionary Computation. Washington D. C., USA: [s. n.], 1999.

编辑 张 帆