

文章编号: 1000-8152(2010)08-1060-09

## 动态环境中的Memetic算法

王洪峰, 汪定伟, 黄 敏

(东北大学 信息科学与工程学院, 辽宁 沈阳 110004)

**摘要:** 针对近几年在进化计算领域被广泛关注的动态优化问题, 提出了一种基于粒子群优化(PSO)的Memetic算法。在一种环状拓扑结构的局部PSO模型中, 利用模糊认知局域搜索策略来改善部分粒子的质量, 同时引入一种自组织随机移民策略来保持算法的种群多样性。通过对一组标准动态测试问题的仿真实验, 能够证明所提出的算法在动态环境中的有效性和适应能力。

**关键词:** Memetic算法; 粒子群优化算法; 局域搜索; 动态优化问题

中图分类号: TP18 文献标识码: A

## Memetic algorithms in dynamic environments

WANG Hong-feng, WANG Ding-wei, HUANG Min

(Information Science and Engineering School, Northeastern University, Shenyang Liaoning 110004, China)

**Abstract:** Based on particle swarm optimization(PSO), we propose a memetic algorithm for solving dynamic optimization problems which are widely concerned from the evolutionary computation community. In this algorithm, a fuzzy cognition local search method is employed for improving the quality of individuals and a self-organized random immigrant scheme is used to further enhance the exploration capacity in a local version of PSO with a ring-shape topology structure. Experimental study over a series of dynamic test benchmark problems shows that the proposed PSO-based Memetic algorithm is robust and adaptable in the dynamic environments.

**Key words:** Memetic algorithm; PSO; local search; dynamic optimization problem

## 1 引言(Introduction)

在实际工程应用中, 很多优化问题通常是动态的, 比如新工件可能随时到达, 机器可能随时发生故障而降低加工速度等等。这些现实中的动态性往往会引起问题目标函数, 决策变量或者约束条件的变化, 从而给问题的优化带来很大的挑战。近几年, 一些研究者开始利用进化算法(evolutionary algorithm, EA)来解决这类动态优化问题, 提出各种策略来改善EA追踪变化中最优解的能力, 详细的相关综述读者可以参考文献[1, 2]。

Memetic算法(Memetic algorithm, MA)是近些年进化计算领域的一个研究热点<sup>[3]</sup>, 它是基于达尔文的自然进化理论和道金斯的文化进化思想而衍生的一种将EA和局域搜索(local search, LS)结合在一起的混合算法, 也常常被称为遗传局域搜索(genetic local search)。在MA的算法框架内, EA算子被用于执行广域搜索, LS策略被用于对某些个体执行局部改

进, 从而能够使得算法在探索和开发能力之间保持较好的平衡。值得注意的是, 尽管MA被成功地应用于解决很多复杂的静态优化问题<sup>[4,5]</sup>, 但对动态环境中优化问题的研究还很少。

作为一种模拟鱼群和鸟群社会行为的优化技术, 粒子群优化(particle swarm optimization, PSO)算法在最近十几年来已经快速发展成为进化计算领域十分活跃的算法, 而且也开始逐渐被应用于解决各种动态优化问题<sup>[6~8]</sup>。根据粒子不同的学习策略, PSO可以分为全局和局部两种版本。在全局版本的PSO中, 所有粒子都向种群中的最好粒子学习; 局部版本的PSO仅允许粒子向其邻域中最好粒子学习。显然, 局部PSO具有较慢的收敛速度, 有利于保持更好的种群多样性, 这对于追踪变化的最优解是很必要的。然而目前PSO的研究大多局限于全局版本, 利用局部PSO求解优化问题特别是动态优化问题的研究并不多。

收稿日期: 2009-07-25; 收修改稿日期: 2009-11-10。

基金项目: 国家自然科学基金资助项目(70931001, 70771021, 70671020); 国家创新研究群体科学基金资助项目(60821063); 中央高校基本科研业务费专项资金资助(N090404020); 教育部博士点新教师基金资助项目(200801451053)。

本文将一种模糊认知局部搜索(fuzzy cognition local search, FCLS)策略引入到局部PSO中, 提出一种新的求解动态优化问题的Memetic算法。此外, 一种自组织随机移民(self-organized random immigrants, SORI)策略也被扩展到提出的算法中用以进一步增强其在动态环境中的性能。通过对一系列由移动峰函数构造的动态测试环境的仿真实验, 检验所提出的基于PSO的Memetic算法性能。

论文的主要内容如下: 首先介绍相关的工作, 包括基本的PSO算法及相关文献综述, 然后详细介绍所提出的算法, 接着是实验结果和分析, 最后给出结论以及下一步的研究计划。

## 2 相关研究工作(Related works)

### 2.1 基本PSO(Principles of PSO)

PSO是一种基于种群的迭代优化技术, 与其他EA不同的是, PSO中个体(粒子)通过在搜索空间内的移动而不是进化来寻找最优解。每个粒子可以具有自己的位置和速度, 在每次迭代中, 粒子会根据自身当前的位置和速度, 自身曾经发现的最好解的位置以及被邻域中伙伴所发现的最好解的位置来进行更新。下面的公式描述了这样的迭代行为:

$$\begin{aligned} \vec{v}_i(t+1) &= \\ \omega \vec{v}_i(t) + c_1 \xi(\vec{p}_i(t) - \vec{x}_i(t)) + c_2 \eta(\vec{p}_{gi}(t) - \vec{x}_i(t)), & (1) \\ \vec{x}_i(t+1) &= \vec{x}_i(t) + \vec{v}_i(t+1), & (2) \end{aligned}$$

上式中:  $\vec{v}_i(t)$  和  $\vec{x}_i(t)$  分别表示粒子  $i$  在第  $t$  次迭代时的速度和位置,  $\vec{p}_i(t)$  是粒子  $i$  曾经发现的最好解(pbest)的位置,  $\vec{p}_{gi}$  是被粒子  $i$  伙伴所发现的最好解(gbest)的位置,  $\omega$  是惯性权重, 用于决定粒子对以前速度的继承大小,  $c_1$  和  $c_2$  分别表示认知和社会学习因子,  $\xi$  和  $\eta$  是两个在  $[0, 1]$  之间均匀分布的随机数。

根据选择gbest方法的不同, PSO能够分为全局和局部两种不同的版本。在全局版本PSO中, 每个粒子都会被整个种群中的最好粒子所影响, 从而导致所有粒子向解空间的某个极值点附近迅速地移动。相反, 在局部PSO中, 每个粒子仅被它邻域中的最好粒子所影响, 这就会使得整个粒子群会缓慢的收敛。考虑到一个缓慢收敛的种群能够更容易适应一个变化的环境, 这就使得利用局部PSO来求解动态优化问题成为一个值得进一步深入研究的工作。

### 2.2 动态环境中的PSO(PSO in dynamic environments)

利用PSO来求解动态优化问题的研究最早是由Eberhart和Shi<sup>[9]</sup>提出的, 他们利用基本PSO算法来追踪一个时变单峰函数, 实验结果表明简单的PSO很

难处理较为复杂的环境变化。Hu和Eberhart<sup>[10]</sup>提出了一种适应性的PSO, 利用各种环境探测和重新随机化策略来使PSO能够较为有效地适应不同强度的环境变化。在Carlisle和Dozier的工作<sup>[11]</sup>中, 每个粒子会重新设置其最好解pbest的位置, 以避免因环境变化而产生的过时信息误导粒子的搜索方向, 实验结果显示所提出的两种重设策略能够有效改善PSO在动态环境中的性能。

最近, 研究者们在利用PSO求解动态优化问题时使用了一些特殊的保持种群多样性方法。Blackwell和Bentley<sup>[12]</sup>将一种带电荷的粒子引入到PSO中, 与正常粒子相比, 电荷粒子更新时还要受到它们之间排斥力的影响, 以防止种群过于收敛。在文献[13]中一种复合粒子被构造来保持种群多样性, 复合粒子实际上是由若干个粒子组成的具有特定拓扑结构的粒子集合, 并采用一种特殊映射方法进行移动。

多粒子群策略也是PSO在动态环境中采用的一种常见策略。Blackwell和Branke<sup>[6]</sup>利用多个相互作用的粒子群同时追踪解空间内的多个极值点, 在他们的工作中, 两种特殊的算子(排斥算子和反收敛算子)分别被用于阻止多个粒子群收敛于同一个峰和保持对解空间内新峰的探索能力; 另外, 一些电荷粒子或者量子粒子也被用来保持各粒子群的种群多样性。Parott和Li<sup>[7]</sup>也提出了一种多粒子群算法, 粒子群的数量和大小会根据一种Species机制进行动态的调整, 实验结果表明这种基于Species的PSO能够有效地求解静态和动态环境中的多峰问题。Jansson和Middendorf<sup>[14]</sup>在一种求解动态优化问题的分级PSO算法中也采用了多粒子群策略, 当环境变化后一些子粒子群能够分离出来以适应环境的变化。

## 3 提出的算法(The proposed algorithm)

### 3.1 算法的基本流程(The general framework of algorithm)

本文研究的算法是一种基于PSO的Memetic算法, 也就是, 利用PSO模型来保持算法对解空间的探索, 利用LS策略对种群中某些粒子进行局部改进。这类Memetic算法的基本流程可由下面流程中所示伪码来描述:

```
Procedure General PSO-based MA:  

begin  

for  $i := 1$  to  $s\_size$  do  

    initialize particle  $i$  with random  $\vec{x}_i$  and  $\vec{v}_i$ ;  

    evaluate  $f(\vec{x}_i)$ ;  

     $\vec{p}_i := \vec{x}_i$ ;  

endfor
```

```

if LS is applied then
    select particles from the population for LS;
    apply LS upon each of selected particles;
endif
for i := 1 to s_size do
    update  $\vec{p}_{gi}$ ;
endfor
repeat
for i := 1 to s_size do
    update  $\vec{v}_i$  and  $\vec{x}_i$  according to Eqs.(1) and (2);
    evaluate  $f(\vec{x}_i)$ ;
    if  $f(\vec{p}_i) < f(\vec{x}_i)$  then  $\vec{p}_i := \vec{x}_i$ ;
endfor
if LS is applied then
    select particles from the population for LS;
    apply LS upon each of selected particles;
endif
for i := 1 to s_size do
    update  $\vec{p}_{gi}$ ;
endfor
until a stop condition is met
end

```

其中 $s\_size$ 表示粒子群的大小。如上面流程所示,这种Memetic算法也是从一个随机产生的初始种群(粒子的初始化包括随机产生其初始位置和速度)开始,接着利用某种方法从种群中选择部分粒子进行局部改进,并根据定义的邻域结构更新每个粒子的gbest。在接下来的迭代中,首先根据公式(1)和(2)更新粒子的位置和速度,然后对选择出来的部分粒子执行LS,最后更新每个粒子的gbest,这样的迭代过程持续到某个终止条件被满足为止。

根据2.1中的介绍,局部PSO具有缓慢的种群收敛速度,有助于算法长时间保持较好的探索能力,这显然有利于保证算法对环境变化的适应能力。于是,在本文研究中,一种特殊邻域结构的局部PSO模型被采用,所有的粒子都被安排在一个环状结构中,也就是说,粒子 $\vec{x}_1$ 与粒子 $\vec{x}_{s\_size}$ 相连。在这种环状拓扑的PSO模型中,粒子的邻域是通过它在种群中的编号来确定。设 $r_n$ 表示邻域半径,则粒子*i*的邻域中包含了从编号 $(i - r_n)$ 到 $(i + r_n)$ 的所有粒子。显然, $r_n$ 越小,粒子群收敛得越慢。为了最大程度保持种群的多样性,在本文 $r_n$ 被设为1,也就是说每个粒子只能与它最相邻的两个粒子进行“信息共享”。

### 3.2 LS策略(Local search)

尽管上述的PSO模型有助于保持算法对环境变化的适应能力,但值得注意的是局部PSO的开发能力往往不强,这就使得其很难迅速地追踪到变化后

的最优点。于是本文考虑利用某种LS策略来增强算法的开发能力。

通过对基本模型中公式(1)进行变形, Kennedy<sup>[15]</sup>提出了两种特殊的PSO模型:若不包含社会部分则称之为认知模型;若不包含认知部分则称之为社会模型。显然,认知模型是为了加强粒子对其自身发现的最好解 $p_{best}$ 的进一步开发。为了改善粒子在动态环境中对 $p_{best}$ 的追踪能力,这里提出一种模糊认知模型,如下式所示:

$$\vec{p}'(t) = N(\vec{p}(t), \sigma), \quad (3)$$

$$\vec{v}(t+1) = \omega \vec{v}(t) + c_1 \xi (\vec{p}'(t) - \vec{x}(t)), \quad (4)$$

其中:  $t$ 表示迭代次数,  $\vec{p}'(t)$ 表示粒子 $\vec{x}$ 的个人最好解( $p_{best}$ ) $p(t)$ 的一个模糊位置,能够通过一个正态分布函数 $N(\vec{p}(t), \sigma)$ 来确定,  $\sigma$ 是一个常数,用于控制模糊位置 $\vec{p}'(t)$ 生范围的大小。

根据这种模糊认知模型,本文提出了一种模糊认知局部搜索(fuzzy recognition local search, FCLS)策略,其伪码如下流程所示:

**Procedure** Algorithm of FCLS operator:

```

begin
     $\vec{x}' := \vec{x}$ ;
    initialize  $\vec{v}'$ ;
    for  $j := 1$  to  $ls\_size$  do
        generate  $\vec{p}'$  according to Eqs. (3);
        generate  $\vec{v}'$  according to Eqs. (4);
         $\vec{x}' := \vec{x} + \vec{v}'$ ;
        if  $f(\vec{x}') > f(\vec{x})$  then  $\vec{x} := \vec{x}'$ ;
        if  $f(\vec{x}') > f(\vec{p})$  then  $\vec{p} := \vec{x}'$ ;
    endfor
end;

```

Denotations:

$\vec{x}$ : the selected particle for local improvement

$ls\_size$ : the step size of LS operator

$\vec{v}'$ : the velocity of each step LS move

$\vec{p}$ :  $p_{best}$  of selected particle  $\vec{x}$

从上面流程能够看出,  $\sigma$ 是一个影响FCLS算子效果的参数,它的值越小,也就意味着模糊 $p_{best}$ 越接近它的原始位置。当 $\sigma = 0$ 时,则模糊认知模型变回认知模型,此时粒子的移动完全被它的 $p_{best}$ 所决定。此外,为了减小随机因素的影响,  $\vec{v}'$ 的初始化被限制在一个很小的范围之内(本文中 $\vec{v}'$ 总是在 $[0, 5]^n$ 内进行随机初始化)。

接下来要解决的问题是怎样从当前种群中挑选出合适的粒子进行局部改进。在很多相关研究中,LS策略被应用到每一个新产生的个体上,这无疑会大大增加算法运行的时间成本,使得MA很难对外界

环境的变化迅速做出反应。当然,如果解空间中只有一个最优解,那么仅对当前适值最好的粒子进行改进是一个不错的选择<sup>[16]</sup>。然而考虑到很多实际问题往往是动态多峰的,本文提出了一种更为适应的选择算法,如下流程所示:

```
Procedure Algorithm of selecting particles for LS:
begin
   $S := \emptyset;$ 
   $count := 0;$ 
   $P' := P;$ 
  while ( $count \leq max\_ls\_count$ ) do
    get best fitness particle  $\vec{x} \in P'$ ;
     $S := S \cup \{\vec{x}\};$ 
     $count := count + 1;$ 
    remove  $\vec{x}$  from  $P'$ ;
    for each  $\vec{y} \in P'$  do
      if  $d(\vec{x}, \vec{y}) < r_s$  then remove  $\vec{y}$  from  $P'$ ;
    endfor
    if  $P' = \emptyset$  then break;
  endwhile
end;
```

Dominations:

$S$ : a list of all selected particles to be improved by LS

$P$ : the current population

$max\_ls\_count$ : the allowable maximal value of  $|S|$

$d(\vec{x}, \vec{y})$ : the Euclidean distance between  $\vec{x}$  and  $\vec{y}$

$r_s$ : a parameter of controlling the crowded degree in  $S$

在上面流程中,  $S$ 表示局域更新集, 它总初始化为空集。首先将临时选择池 $P'$ 中当前最好的粒子 $\vec{x}$ 移入 $S$ , 然后 $P'$ 中剩余粒子依次被检查与 $\vec{x}$ 的距离, 如果某个粒子与 $\vec{x}$ 之间的距离小于 $r_s$ , 则这个粒子就会被从 $P'$ 中移出。这样的选择过程一直持续到 $P'$ 变为空集或者 $S$ 的长度( $|S|$ )达到最大允许值。显然这种选择算法的目的是为了挑选出多个高质量且不聚集的粒子进行LS改进。参数 $r_s$ 用于控制 $S$ 集的拥挤程度, 保证LS能够开发不同的峰。另外, 由于每一代 $|S|$ 的大小可能是不同的, 所以对 $S$ 中每个粒子进行局域搜索的步数 $ls\_size$ 需要重新计算。如果每一代总的LS步数 $total\_ls\_size$ 是固定不变的, 那么 $ls\_size = \lceil total\_ls\_size / |S| \rceil$ 。

### 3.3 增加种群多样性(Increasing population diversity)

尽管局部PSO模型能够保持一定程度的种群多样性, 然而随着种群的缓慢收敛, 算法会逐渐失去对新峰的探索能力。这在动态环境下是需要避免的, 因为新峰可能会随时出现在适值函数曲线上。

为了保证算法始终具有足够的探索能力, 一种自组织随机移民策略(self-organized random immigrants, SORI)<sup>[17]</sup>被扩展到提出的Memetic算法中。在这种SORI方法中, 每经过一定的迭代次数( $t\_ri$ ), 一段连续的粒子(其长度为 $r_r$ )被随机选择进行初始化, 并从整个粒子群环中分离出来, 而分离出来的子粒子群独立地搜索解空间, 直到当一个新的子粒子群产生后它会重新加入主粒子群中。此外, 子粒子群采用全局模式进行更新, 并且其最好粒子也会被执行LS改进。

综上, 通过将FCLS算子和SORI策略引入到一种环状拓扑结构的局部PSO模型中, 本文提出了一种新的求解动态优化问题的Memetic算法, 其流程如下:

**Procedure** Proposed PSO-based MA:

```
begin
   $t = 1;$ 
  for  $i := 1$  to  $s\_size$  do
    initialize particle  $i$  with random  $\vec{x}_i$  and  $\vec{v}_i$ ;
    evaluate  $f(\vec{x}_i)$ ;
     $\vec{p}_i := \vec{x}_i$ ;
  endfor
  if SORI scheme is used then
     $k := rand(1, s\_size);$ 
    set the status of particles with indices from  $k$  to  $(k+r_r-1)$  as preserved;
  endif
  if FCLS operator is used then
    apply FCLS operator upon the best preserved particle;
    select a set  $|S|$  of unpreserved particles for LS;
    apply FCLS operator upon each particle in  $S$ ;
  endif
  for  $i := 1$  to  $s\_size$  do
    update  $\vec{p}_{gi}$ ;
  endfor
  repeat
     $t = t + 1;$ 
    for  $i := 1$  to  $s\_size$  do
      update  $\vec{v}_i$  and  $\vec{x}_i$  according to Eqs.(1) and (2);
      evaluate  $f(\vec{x}_i)$ ;
      if  $f(\vec{p}_i) < f(\vec{x}_i)$  then  $\vec{p}_i := \vec{x}_i$ ;
    endfor
    if SORI scheme is used and  $t \% t\_ri == 0$  then
       $k := rand(1, s\_size);$ 
      re-initialize particles with indices from  $k$  to  $(k+r_r-1)$  randomly and set their statuses as preserved;
    endif
    if FCLS operator is used then
```

```

apply FCLS operator upon the best preserved particle;
select a set  $|S|$  of unpreserved particles for LS;
apply FCLS operator upon each particle in  $S$ ;
endif
for  $i := 1$  to  $s\_size$  do
    update  $\vec{p}_{gi}$ ;
endfor
until a stop condition is met
end

```

## 4 实验研究(Experimental study)

### 4.1 算法测试环境(Algorithm test environments)

在本文的实验中, 利用一类标准动态测试函数移动峰问题(moving peaks benchmarks, MPB)构造出一系列动态环境用于测试算法性能. MPB问题可以描述为在 $n$ 维实空间内存在 $m$ 个可以改变位置, 高度和宽度的峰, 其函数曲线描述如下:

$$F(\vec{x}, t) = \max_{i=1, \dots, m} \frac{H_i(t)}{1 + W_i(t) \sum_{j=1}^n (x_j(t) - X_{ij}(t))^2}, \quad (5)$$

其中:  $\vec{X}_i(t) = \{X_{i0}(t), X_{i1}(t), \dots, X_{in}(t)\}$ ,  $H_i(t)$  和  $W_i(t)$  分别表示 $t$ 时刻峰 $i$ 的位置, 高度和宽度.

下文实验中采用的MPB参数设置对应其标准测试网站上的Scenario1<sup>[18]</sup>, 这种MPB函数定义在一个含有5个移动峰的5维空间中. 每经过 $\tau$ 次迭代, 每个峰的高度和宽度都会被加上一个随机的高斯变量, 而位置会根据一个固定长度为 $\rho$ 的随机向量 $\vec{v}$ 进行移动. 这样的变化可以通过如下公式进行描述:

$$\begin{cases} \delta \in N(0, 1), \\ H_i(t) = H_i(t-1) + 7 \cdot \delta, \\ W_i(t) = W_i(t-1) + 0.01 \cdot \delta, \\ \vec{X}_i(t) = \vec{X}_i(t-1) + \vec{v}_i(t), \end{cases} \quad (6)$$

$$\vec{v}_i(t) = \frac{\rho}{|\vec{r} + \vec{v}_i(t-1)|} ((1-\lambda)\vec{r} + \lambda\vec{v}_i(t-1)), \quad (7)$$

显然, 参数 $\tau$ 可以用来控制环境变化的快慢, 而参数 $\rho$ 可以用来控制一个环境变化的强度.

为了检验算法在不同性质动态环境中的性能, 本文利用上述MPB问题构造了一系列动态测试环境. 为了检验算法在不同变化强度环境中的性能, 参数 $\rho$ 分别被设为0.1, 1.0, 2.0和5.0: 从非常弱的变化( $\rho = 0.1$ ), 到中等强度的变化( $\rho = 1.0$ 和2.0), 再到很强的变化( $\rho = 5.0$ ). 变化速度参数 $\tau$ 分别被设为50, 100和200, 分别表示环境变化很快, 中等速度和很慢. 这样可以构造12个不同的动态优化问题如表1所示.

表1 构造的动态环境

Table 1 The constructed dynamic environments

$\tau$	动态环境指标				
	50	1	2	3	4
100	5	6	7	8	
200	9	10	11	12	
$\rho$	0.1	1.0	2.0	5.0	

### 4.2 实验设置(Experimental design)

在接下来的实验中, 一方面检验所提出算法的主要性质, 另一方面通过与已有算法的比较检验其求解动态优化问题的能力. 为便于描述和分析, 下文实验中所涉及的一些算法的缩写表述如下: LPSO(3.1中所描述的局部PSO), GPSO(全局PSO), LFCMA(基于LPSO的采用FCLS算子的MA), GFCMA(基于GPSO的采用FCLS算子的MA), LRWMA(基于LPSO的采用RWDE算子的MA, RWDE算子是文献[19]中提出的一种随机迭代LS策略), multiCPSO(文献[6]中提出的一种带电荷的多粒子群算法).

为了保证比较的公平性, 所有算法在每次迭代的估值次数被固定为100次, 惯性权重 $\omega$ 设为0.72948, 学习因子 $c_1$ 和 $c_2$ 都设为1.4962, 每个粒子的速度都被限制在 $[-V_{\max}, +V_{\max}]$ 内(这里 $V_{\max} = 100$ ). 另外, 本文提出的Memetic算法中的一些特殊参数分别被设为:  $r_s = 1.0$ ,  $r_r = 5$ ,  $t\_ri = 25$ . 算法每次运行都要经历10个变化周期, 所有的实验结果都是采用相同随机种子运行30次后所获得的平均值.

由于动态环境中并不存在一个唯一的最优解, 因此本文利用一种新的离线性能指标 $\bar{E}_{BG}$ 来评价算法的性能,  $\bar{E}_{BG}$ 的计算公式如下:

$$\bar{E}_{BG} = \frac{1}{G} \sum_{i=1}^G \left( \frac{1}{N} \sum_{j=1}^N (F_{ij}^* - F_{BG_{ij}}) \right), \quad (8)$$

其中:  $G$ 是算法迭代次数( $G = 10 * \tau$ ),  $N = 30$ 是算法运行次数,  $F_{BG_{ij}}$ 表示算法在第 $j$ 次运行的第 $i$ 代所获得最好解的适值,  $F_{ij}^*$ 表示全局最优解的适值. 显然, 在最优解适值已知的情况下, 这种离线性能指标能够很好地评估算法对它的追踪能力.

### 4.3 LS算子效果的实验研究(Experimental study on the effect of LS operators)

首先检验FCLS算子中参数 $\sigma$ 对其效果的影响. 在具体的实验中, 3种不同的LFCMA算法( $\sigma$ 分别设为0.05, 0.5和5.0)仅在MPB问题的静态阶段上运行, 粒子群大小 $s\_size = 90$ , 每次迭代总对当前最好粒子执行 $ls\_size = 10$ 次FCLS操作. 对于算法每次运行, 最大的迭代次数设为100, 图1中的实验结果是采

用相同随机种子运行30次后所获得的平均值。

从图1中能够看出,采用 $\sigma = 5.0$ 方案的LFCMA算法在早期的搜索阶段表现不错,然而很快被 $\sigma = 0.5$ 方案所超过。当 $\sigma$ 值很小( $\sigma = 0.05$ )时,LFCMA算法直到后期搜索阶段才表现得较为理想。考虑到一个中等程度的 $\sigma$ 值能够使FCLS算子在测试函数中表现出较好的性能,因此在后面的实验中 $\sigma$ 将被固定为0.5。

类似的实验被用于检验不同的LS算子对算法性能的影响。在具体的实验中,同样在MPB函数的静态阶段执行5种不同的算法,包括LPSO, GPSO, LFCMA, LRWMA和GFCMA。LPSO和GPSO中粒子群大小 $s\_size = 100$ ,而其他3种MA中 $s\_size = 90$ ,这是因为Memetic算法每次迭代都要对当前的最好粒子执行 $ls\_size = 10$ 次局域改进。同样也将各算法采用相同随机种子运行30次后所获得的平均值作为实验结果,如图2所示。

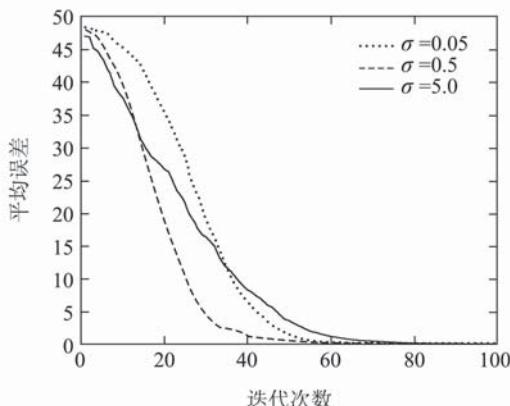


图1 不同LFCMA算法在静态MPB问题中的实验结果  
Fig. 1 Experimental results of different LFCMAs on the stationary MPB problem

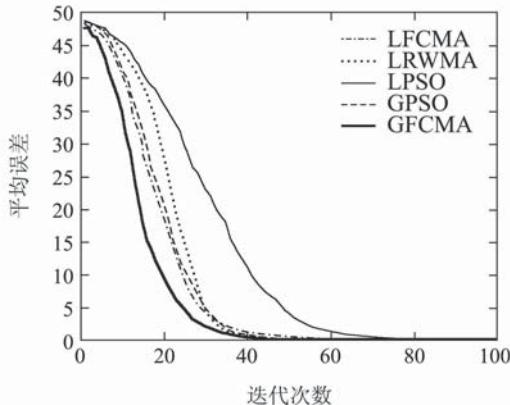


图2 采用不同LS策略的MAs在静态MPB问题中的性能  
Fig. 2 Experimental results of MAs with different LS operators on the stationary test problem

从图2中能够看出,首先,LFCMA和LRWMA都显著优于LPSO,而GFCMA的表现也明显好于GP-

SO。这说明LS策略能够有效地改善算法的性能。其次,在两种基于LPSO的Memetic算法中,FCLS算子能够帮助算法执行比RWDE算子更为鲁棒的局域改进。这是由于RWDE算子总是执行一种随机移动,而FCLS算子同时具有随机移动和有偏移动的特点,这就使粒子能够更容易的接近最优解。显然,这样的结论能够表明本文提出的FCLS算子的有效性。

FCLS算子的另一个特征是怎样从当前的粒子群中选择合适的粒子进行局域改进。在3.2中提出了一种选择算法(如流程图所示)来保证能够从粒子群中选择若干个高适值且不聚集的粒子。在接下来的实验中,将通过其他两种策略的比较来检验这种选择方法的效果:一种策略是仅仅改善当前的最好粒子,另一种策略是随机选择若干个粒子进行局域更新。为了便于描述,LFCMA1, LFCMA2和LFCMA3分别表示采用3.2中选择算法,选择最好粒子和随机选择粒子的LFCMA。

在具体的实验中,LFCMA2中粒子群大小 $s\_size = 90$ ,而LFCMA1和LFCMA3中 $s\_size = 70$ ; LFCMA2中, $ls\_size = 10$ , LFCMA3中随机选择的粒子个数为5,每个粒子被局域改进的次数 $ls\_size = 6$ ,LFCMA1中允许选择的最大粒子数量(max\_ls\_count)被设为5,而每次迭代最大局域改进次数(total\_ls\_size)设为30。3种LFCMA算法被用于求解表1中所构造的动态优化问题,其实验结果如表2所示。为了更好的评价算法的性能,表3给出了不同算法在置信水平为0.05自由度为58的单侧t-检验结果,其中,Alg.1–Alg.2被标注为“+”,“-”,“+”和“-”分别表示Alg.1的性能显著优于,显著劣于,不显著优于和不显著劣于Alg.2的性能。从表2和表3中,可以获得如下结论。

总体上说,LFCMA1在大多数动态问题中总是表现得最好,这表明笔者提出的选择算法能够有效改善LPSO-FCMA在动态环境中的性能。这是因为LFCMA1能够利用FCLS算子同时对适值曲线上不同峰上的粒子进行开发,这显然非常适用于动态多峰问题。

LFCMA2优于LFCMA3,甚至在一些变化强度很小的动态环境中优于LFCMA1。这是因为当环境变化强度较弱时,新的最优解往往与旧的最优解很接近。在这种情况下,仅对最好粒子执行充分的FCLS操作可能是更有利的。然而一旦环境变化得较为强烈,LFCMA2中的选择策略就可能会误导FCLS算子,这就是为什么在所有 $\rho = 5.0$ 的动态问题上LFCMA2显著劣于LFCMA1的原因。LFCMA3总是表现得最差,说明随机选择的策略是令人失望的。因此,LFC-

MA1中的选择算法在后面的实验中将被一直采用.

表2 不同LFCMA在动态测试问题中的实验结果  
Table 2 Experimental results of different LFCMAs on dynamic test problems

动态测试		算法		
$\tau$	$\rho$	LFCMA1	LFCMA2	LFCMA3
50	0.1	1.97±0.90	2.30±1.83	3.24±1.47
50	1.0	2.63±1.06	2.85±1.14	3.59±1.32
50	2.0	3.42±1.04	3.94±1.88	5.17±1.49
50	5.0	5.52±1.04	6.27±1.42	9.26±1.87
100	0.1	1.04±0.91	1.00±0.26	1.56±0.66
100	1.0	1.38±0.68	1.27±0.58	2.01±0.95
100	2.0	2.13±1.47	2.46±1.61	2.79±1.03
100	5.0	3.37±0.50	4.01±1.24	5.24±0.78
200	0.1	0.46±0.25	0.53±0.51	0.87±0.67
200	1.0	0.77±0.61	0.82±0.62	1.03±0.51
200	2.0	1.12±0.62	1.36±0.76	1.51±0.51
200	5.0	1.88±0.43	2.78±1.15	3.04±0.87

表3 不同LFCMA在动态测试问题中的统计比较结果

Table 3 The  $t$ -test results of comparing different LFCMAs on dynamic test problems

$t$ 检验结果		MPB问题		
$\tau = 50, \rho \Rightarrow$		0.1	1.0	2.0
LFCMA1–LFCMA2	+	+	+	s+
LFCMA1–LFCMA3	s+	s+	s+	s+
LFCMA2–LFCMA3	s+	s+	s+	s+
$\tau = 100, \rho \Rightarrow$		0.1	1.0	2.0
LFCMA1–LFCMA2	–	–	+	s+
LFCMA1–LFCMA3	s+	s+	s+	s+
LFCMA2–LFCMA3	s+	s+	+	s+
$\tau = 200, \rho \Rightarrow$		0.1	1.0	2.0
LFCMA1–LFCMA2	+	+	+	s+
LFCMA1–LFCMA3	s+	+	s+	s+
LFCMA2–LFCMA3	s+	+	+	+

#### 4.4 多样性策略效果的实验研究(Experimental study on the effect of diversity schemes)

上述实验中LFCMA算法在动态环境中依然会面临因粒子群收敛而逐渐失去对解空间中新峰探测能力的问题. 在3.3中, 提出一种SORI策略来解决这个问题. 接下来的实验将检验这种SORI策略对LFCMA算法性能的影响. 在具体的实验中, LFCMA算法和引入SORI策略的LFCMA算法(用LSOFCMA

表示)被用于求解表1中所构造的动态问题, 在LSOFCMA算法中每次迭代还需对最好的受保护粒子执行5次LS操作, 而其他的参数与4.3中LFCMA的设置方法相同. 实验结果如表4所示, 其中 $t$ -检验列表示LSOFCMA–LFCMA在置信水平为0.05自由度为58的单侧 $t$ -检验结果.

表4 LSOFCMA和LFCMA在动态测试问题中的实验结果

Table 4 Experimental results of LSOFCMA and LFCMA on dynamic test problems

动态测试		算法		
$\tau$	$\rho$	LSOFCMA	LFCMA	$t$ 检验结果
50	0.1	1.81±0.33	2.77±2.61	s+
50	1.0	2.68±1.24	4.17±5.36	s+
50	2.0	3.50±1.12	3.68±1.83	+
50	5.0	6.32±1.54	7.05±4.24	s+
100	0.1	1.29±0.99	1.32±1.44	+
100	1.0	1.25±0.39	1.32±0.70	+
100	2.0	2.03±1.46	2.07±1.31	+
100	5.0	3.36±0.72	3.47±0.97	+
200	0.1	0.60±0.48	1.22±2.04	+
200	1.0	0.79±0.57	0.72±0.38	–
200	2.0	1.00±0.82	1.44±1.96	+
200	5.0	2.16±2.16	2.11±1.08	–

从表4中能够看出, LSOFCMA在大多数动态环境中总是优于LFCMA. LFCMA能够保持一定程度的种群多样性主要是依赖于它较小的邻域半径, 然而随着迭代的进行, 粒子将会逐渐收敛于它的邻域最好解 $g_{best}$ . 也就是说, 经过一定的迭代次数之后, LFCMA将失去对新峰的探测能力. 然而SORI策略一方面能够通过随机初始化若干个连续的粒子来始终保持充分的种群多样性, 另一方面新初始化的具有较低适值的粒子能够被保护并独立的进化一定得迭代次数, 从而能够使LSOFCMA始终具有对新峰的探测能力. 显然, 实验结果也表明这种SORI策略能够有效改善算法在动态环境中的性能.

#### 4.5 与其他算法比较的实验研究(Experimental study on comparing with the peer algorithms)

最后, 将LSOFCMA算法与文献中一种多粒子群的PSO算法进行比较来进一步检验其性能. 在具体的实验中, 两种multiCPSO算法被采用: 一种含有5个粒子群, 而另一种含有10个粒子群(分别用multiCPSO1和multiCPSO2来表示). 显然仅对本文研究的MPB问题来说, multiCPSO1具有最合适的粒子群数,

而multiCPSO2具有不恰当的粒子群数。对于每个粒子群中的正常粒子和带电荷粒子数, multiCPSO1分别被设为10和10, 而multiCPSO2分别设为5和5, 其他的相关参数总保持其初始设置。LSOFCMA算法的相关参数与4.4中实验的设置保持一致。实验结果如表5和表6所示, 从中能够发现如下结论。

表 5 LSOFCMA与multiCPSO在动态测试问题中的实验结果

Table 5 Experimental results of LSOFCMA and multiCPSO on dynamic test problems

动态测试		算法		
$\tau$	$\rho$	LSOFCMA	multiCPSO1	multiCPSO2
50	0.1	1.83±0.38	2.31±0.37	2.58±0.34
50	1.0	2.48±0.84	2.81±0.58	3.14±0.35
50	2.0	3.16±0.65	3.50±0.49	4.52±0.51
50	5.0	6.50±2.05	7.13±1.68	10.22±1.94
100	0.1	1.06±0.63	1.11±0.14	1.29±0.15
100	1.0	1.25±0.39	1.47±0.26	2.93±0.59
100	2.0	1.95±0.89	1.68±0.21	2.27±0.33
100	5.0	3.34±0.70	3.50±0.47	5.26±0.68
200	0.1	0.59±0.35	0.60±0.07	0.68±0.07
200	1.0	0.68±0.33	0.66±0.07	0.83±0.09
200	2.0	0.88±0.17	0.85±0.12	1.19±0.19
200	5.0	1.66±0.37	1.70±0.22	2.61±0.39

表 6 LSOFCMA与multiCPSO在动态测试问题中的统计比较结果

Table 6 The *t*-test results of comparing LFCMAs and multiCPSO on dynamic test problems

<i>t</i> 检验结果		MPB问题			
$\tau = 50, \rho \Rightarrow$		0.1	1.0	2.0	5.0
LSOFCMA–multiCPSO1	<i>s+</i>	+	<i>s+</i>	+	
LSOFCMA–multiCPSO2	<i>s+</i>	<i>s+</i>	<i>s+</i>	<i>s+</i>	
multiCPSO1–multiCPSO2	<i>s+</i>	<i>s+</i>	<i>s+</i>	<i>s+</i>	
$\tau = 100, \rho \Rightarrow$		0.1	1.0	2.0	5.0
LSOFCMA–multiCPSO1	+	<i>s+</i>	–	+	
LSOFCMA–multiCPSO2	<i>s+</i>	<i>s+</i>	+	<i>s+</i>	
multiCPSO1–multiCPSO2	<i>s+</i>	<i>s+</i>	<i>s+</i>	<i>s+</i>	
$\tau = 200, \rho \Rightarrow$		0.1	1.0	2.0	5.0
LSOFCMA–multiCPSO1	+	–	–	+	
LSOFCMA–multiCPSO2	+	<i>s+</i>	<i>s+</i>	<i>s+</i>	
multiCPSO1–multiCPSO2	<i>s+</i>	<i>s+</i>	<i>s+</i>	<i>s+</i>	

首先, 当环境变化得较快的时候, LSOFCMA总是优于multiCPSO1。当 $\tau=100$ 和200时, 情况变得有点不同。尽管LSOFCMA在 $\rho=0.1$ 和5.0时表现较好,

而multiCPSO1在 $\rho=1.0$ 和2.0时表现较好, 但总体上两种算法的表现相差不多。这是因为FCLS算子能够有效增强算法的开发能力, 这就使得LSOFCMA能够较快的追踪到最优解。

其次, multiCPSO2在所有环境中总是表现得最差。这是因为multiCPSO算法的性能很大程度取决于粒子群的数量。如果粒子群数设定得很恰当时, multiCPSO能够获得很好的性能, 一旦粒子群数设定得不恰当, 其性能将会受很大影响。由于实际环境往往总是预先未知的, 对于multiCPSO中粒子群数量的最优设置通常是很难获得的, 这样看来LSOFCMA是一种更为鲁棒的优化方法。

最后, 环境参数能够影响算法的性能, 所有算法的性能会随着 $\tau$ 的增加或 $\rho$ 的减小而变得越来越好。这很容易理解, 因为 $\tau$ 越大, 算法就会在下一次变化之前拥有更多的时间去获得更好的解; 而 $\rho$ 越小, 新的最优解就越接近旧的最优解, 越容易被发现。

## 5 结论(Conclusions)

本文提出了一种有效的求解动态优化问题的Memetic算法, 该算法采用一种环状拓扑结构的局部PSO模型来保持其探索能力, 并利用一种模糊认知局域搜索(FCLS)算子来增强其开发能力。此外, 一种自组织随机移民(SORI)策略被用于增加其种群多样性, 以进一步改善所提出算法适应动态环境的能力。通过对一组标准动态测试问题的实验, 能够获得如下结论。

首先, 被适当策略增强的局部PSO算法在动态环境中能够展现出很好的性能。在本文的实验中, LSOFCMA在大多数动态环境中总是优于其他算法。

其次, FCLS算子在静态和动态环境中都能帮助Memetic算法执行一种鲁棒的局域改进。在本文的实验中, LFCMA在静态问题上优于LRWMA, 而LSOFCMA在很多动态问题上优于multiCPSO。

第三, 当Memetic算法被应用到动态多峰问题时, 对多个高适值而且分布在不同峰上的粒子同时进行局域改进总是很有益的, 本文所提出的选择算法被证明能够很好的解决这样的问题。

第四, 多样性策略能够有效地改善算法求解动态优化问题的能力。相对多粒子群策略来说, SORI策略看起来是一个更为鲁棒的多样性策略。

最后, 环境的动态性质会影响算法的性能。在本文的实验中, 算法的性能会随着环境变化频率的增加或者变化强度的减小而变得更好。

总体上说, 实验结果表明本文所提出的算法是一种不错的求解动态优化问题的方法。当然, 还有很多工作有待进一步研究。比如, 本文没有对算法的收敛

速度进行理论分析,这也是目前动态进化计算领域中一个亟待解决的问题。此外,更多保持种群多样性的策略,例如记忆策略或者Speciation方法,也可以考虑引入到提出的Memetic算法中,以进一步改善其求解动态优化问题的能力。

### 参考文献(References):

- [1] JIN Y, BRANKE J. Evolutionary optimization in uncertain environments-a survey[J]. *IEEE Transactions on Evolutionary Computation*, 2005, 9(3): 303–317.
- [2] 王洪峰, 汪定伟, 杨圣祥. 动态环境中的进化算法[J]. 控制与决策, 2007, 22(2): 127–131.  
(WANG Hongfeng, WANG Dingwei, YANG Shengxiang. Evolutionary algorithms in dynamic environments[J]. *Control and Decision*, 2007, 22(2): 127–131.)
- [3] KRASNOGOR N, SMITH J E. A tutorial for competent memetic algorithms: model, taxonomy and design issues[J]. *IEEE Transactions on Evolutionary Computation*, 2005, 9(5): 474–488.
- [4] GALLARDO J E, COTTA C, FERANDEZ A J. On the hybridization of memetic algorithms with branch-and-bound techniques[J]. *IEEE Trans on Systems, Man, and Cybernetics-Part B: Cybernetics*, 2007, 37(1): 77–83.
- [5] ISHIBUCHI H, YOSHIDA T, MURATA T. Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling[J]. *IEEE Transactions on Evolutionary Computation*, 2003, 7(2): 204–223.
- [6] BLACKWELL T M, BRANKE J. Multiswarms, exclusion, and anti-convergence in dynamic environments[J]. *IEEE Transactions on Evolutionary Computation*, 2006, 10(4): 459–472.
- [7] PARROTT D, LI X. Locating and tracking multiple dynamic optima by a particle swarm model using speciation[J]. *IEEE Transactions on Evolutionary Computation*, 2006, 10(4): 440–458.
- [8] 王洪峰, 汪定伟. 一种动态环境下带有记忆的三岛粒子群算法[J]. 系统工程学报, 2008, 23(2): 252–256.  
(WANG Hongfeng, WANG Dingwei. Tri-Island particle swarm optimization with memory in dynamic environment[J]. *Journal of Systems Engineering*, 2008, 23(2): 252–256.)
- [9] EBERHART R C, SHI Y. Tracking and optimizing dynamic systems with particle swarms[C] //Proceedings of IEEE Congress on Evolutionary Computation. New York: IEEE, 2001, 94–97.
- [10] Hu X, Eberhart R. Adaptive particle swarm optimization: Detection and response to dynamic systems[C] //Proceedings of IEEE Congress on Evolutionary Computation. New York: IEEE, 2002, 1666–1670.
- [11] CARLISLE A, DOZIER G. Adapting particle swarm optimization to dynamic environments[C] //Proceedings of International Conference on Artificial Intelligence. 2000, 1: 429–434.
- [12] BLACKWELL T M, BENTLEY P. Dynamic search with charged swarms[C] //Proceedings of the 2002 Genetic and Evolutionary Computation Conference. [S.I.]: [s.n.], 2002: 19–26.
- [13] LIU L, YANG S, WANG D. Compound particle swarm optimization in dynamic environments[C] //Proceedings of EvoWorkshops 2008. [S.I.]: [s.n.], 2008: 616–625.
- [14] JANSON S, MIDDENDORF M. A hierarchical particle swarm optimizer for noisy and dynamic environments[J]. *Genet Program Evolvable Mach*, 2006, 7(3): 329–354.
- [15] KENNEDY J. The particle swarm: social adaptation of knowledge[C] //Proceedings of IEEE International Conference on Evolutionary Computation. [S.I.]: [s.n.], 1997: 303–308.
- [16] WANG H, YANG S, WANG D. A memetic algorithm with adaptive hill climbing strategy for dynamic optimization problems[J]. *Soft Computing*, 2009, 13(8/9): 763–780.
- [17] TINOS R, YANG S. A self-organizing random immigrants genetic algorithm for dynamic optimization problems[J]. *Genet Program Evolvable Mach*, 2007, 8(3): 255–286.
- [18] BRANKE J. The moving peaks benchmark website[EB/OL]. <http://www.aifb.uni-karlsruhe.de/jbr/MovPeaks>.
- [19] PETALAS Y G, PARSOPOULOS K E, VRAHATIS M N. Memetic particle swarm optimization[J]. *Annals of Operations Research*, 2007, 156(1): 99–127.

### 作者简介:

**王洪峰** (1979—), 男, 博士, 讲师, 目前研究方向为进化计算与动态进化计算、生产计划及调度、供应链管理, E-mail: hfwang@mail.neu.edu.cn;

**汪定伟** (1948—), 男, 博士, 教授, 博士生导师, 目前研究方向为生产计划与调度、人工生命计算、电子商务建模与优化等研究, E-mail: dwwang@mail.neu.edu.cn.

**黄敏** (1968—), 女, 博士, 教授, 博士生导师, 目前研究方向为虚拟企业风险管理、生产计划及存储理论、第四方物流管理以及智能优化等, E-mail: mhuang@mail.neu.edu.cn.