

# 动量交叉粒子群算法

禹云<sup>1</sup>, 陈熙<sup>2†</sup>

(1. 娄底职业技术学院 电子信息工程系, 湖南 娄底 417000; 2. 昆明理工大学 信息工程与自动化学院, 昆明 650504)

**摘要:** 针对传统粒子群算法收敛速度慢、全局搜索能力差的缺点, 提出了一种新的最优交叉动量粒子群算法。该算法通过在粒子群算法中引入一种新的二进制交叉策略来加快粒子群算法的收敛速度, 通过设置新的惯性权重来改善新算法的全局搜索和局部搜索能力, 并且在粒子搜索路径中引入变系数低通滤波器构成的动量算法来平滑粒子搜索路径。几个典型的测试函数仿真表明, 新算法在收敛速度和搜索精度上均得到了明显改善。

**关键词:** 粒子群算法; 最优模拟二进制交叉策略; 变系数低通滤波器

**中图分类号:** TP391; F251      **文献标志码:** A      **文章编号:** 1001-3695(2012)12-4459-04

doi:10.3969/j.issn.1001-3695.2012.12.013

## Momentum particle swarm optimization with optimal crossover

YU Yun<sup>1</sup>, CHEN Xi<sup>2†</sup>

(1. Dept. of Electronic & Information Engineering, Loudi Occupational & Technical College, Loudi Hunan 417000, China; 2. School of Information Engineering & Automation, Kunming University of Science & Technology, Kunming 650504, China)

**Abstract:** Aiming at the PSO's shortcoming about slow convergence rate and badly global searching ability, this paper presented a new particle swarm optimization with optimal crossover (OCPSO). By introducing a new simulated binary-crossover strategy (SBX) and a new strategy of inertia weight setting, it improved the ability of global and local searching. Furthermore, it utilized variable coefficient low-pass filters to update particles' positions of OCPSO, called momentum algorithm, which could enhance the speed and accuracy of convergence. Experimental results on several classical functions indicate that the new algorithm can greatly improve the searching speed and accuracy.

**Key words:** particle swarm optimization; optimal simulated binary-crossover strategy; variable coefficient low-pass filters

## 0 引言

粒子群算法(PSO)是 Kennedy 和 Eberhart<sup>[1,2]</sup> 于 1995 年提出的一种新的群智能算法, 它源于对鸟群捕食行为的研究。通过对简单鸟群系统的模拟, 在多维坐标空间中构造所谓的粒子群, 粒子群中每个个体通过跟踪自己和群体的历史最优值来引导自己的前进方向和速度, 从而实现寻优。由于 PSO 算法概念简单、实现容易、参数易调整、能有效解决复杂优化任务, 近年来已引起了优化及演化计算等领域的学者们的广泛关注, 且在系统辨识、图像分割等领域<sup>[3,4]</sup> 都有了广泛的运用。但粒子群算法与其他智能优化算法一样, 也同样存在收敛速度慢、全局搜索能力差等缺点, 尤其是在比较复杂的多峰函数优化和病态函数优化问题中。为改善 PSO 算法的上述缺点, 近年来国内外专家学者也对其作了很多相应的改进。

Lophbjerg 等人<sup>[5]</sup> 最早将遗传算法中的复制和重组引入到粒子群算法来提高粒子群算法的优化性能。该方法是对按概率  $p_i$  选出的粒子进行交叉操作, 产生的子代代替父代。由于后代选择并不基于适应度, 防止了基于适应度选择对那些多局部极值的函数优化时带来的潜在问题, 从理论上讲该方法可以更好地搜索粒子空间, 增强了多模态函数的寻优能力, 但对于单模态问题优化效果差。文献<sup>[6]</sup> 通过对微粒速度或位置引

入了一个小概率随机变异操作来增强种群多样性 (dissipative PSO, DPSO), 使算法能够有效地进行全局搜索, 但过大的变异概率在增加粒子多样性的同时导致了粒子位置的混乱, 延缓了算法的收敛速度。文献<sup>[7]</sup> 在自组织算法的基础上给出了一种变异操作随时间变化的自适应层次 PSO (self-organizing hierarchical PSO, HPSO) 以进一步提高搜索性能, 但由于算法消除了惯性权重, 算法的全局寻优能力降低。文献<sup>[8]</sup> 提出了通过逃逸运动使微粒能够有效地进行全局和局部搜索的一种改进的自适应逃逸微粒群算法, 但该方法的变异时机难以把握。总的来说, 上述改进都通过引入遗传算法中的相关算子来提高粒子群的优化能力, 虽然一定程度上提高了算法的优化性能, 但都还是存在相应的一些问题。

为此, 本文在分析交叉策略和惯性权重的设置对算法搜索能力影响的基础上, 提出了一种新的最优交叉粒子群算法。新算法引入了模拟二进制交叉来模拟遗传算法中的交叉和变异算子, 采用一种全局最优交叉策略来提高算法收敛速度, 并且通过一种新的惯性权重设置方法来平衡算法的全局和局部搜索能力。另外, 为了避免粒子在搜索过程中出现振荡现象, 采用一个变系数低通滤波器来平滑粒子路径。在搜索早期采用较大的滤波器系数, 从而对粒子路径具有较大的平滑作业; 在搜索后期, 为了尽可能保持粒子的多样性, 防止粒子陷入局部

收稿日期: 2012-04-15; 修回日期: 2012-05-22

作者简介: 禹云 (1978-), 女, 湖南双峰人, 讲师, 硕士, 主要研究方向为计算机应用; 陈熙 (1976-), 男 (通信作者), 助教, 博士, 主要研究方向为生物特征识别、智能优化计算 (chenximing2@163.com)。

极值,采用了较小的滤波器系数。文中采用六个函数优化问题进行测试,并与 DPSO、HPSO、自适应逃逸微粒群算法<sup>[8]</sup>(AEP SO1)和自适应逃逸微粒群算法<sup>[8]</sup>(AEP SO2)进行比较,仿真结果表明了算法的优越性。

### 1 动量粒子群优化算法

Reynolds 通过对鸟群的捕食研究发现,每只鸟仅仅是通过追踪它有限数量的邻居来寻找食物,但最终的整体结果是整个鸟群好像在一个中心的控制之下,即复杂的全局行为是由简单规则的相互作用引起的。PSO 即源于对鸟群捕食行为的研究:一群鸟在随机搜寻食物,如果这个区域里只有一块食物,那么找到食物最简单有效的策略就是搜寻目前离食物最近的鸟的周围区域。PSO 就是从这种模型中得到启示而产生的,并用于解决优化问题。研究以下全局优化问题:

$$\min |f(x) : x \in \Omega \subset R^d|, f: \Omega \subset R^d \rightarrow R^1 \quad (1)$$

所有满足  $\{x|x \in \Omega\}$  的  $x$  叫做问题的可行解,多个可行解的集合叫做一个种群。粒子群算法把群体中的每一个个体看做  $d$  维搜索空间中一个没有体积、质量的微粒点,在搜索空间中以一定的速度飞行。算法用  $d$  维向量  $x_i = (x_{i1}, x_{i2}, \dots, x_{id})^T \in \Omega$  表示第  $i$  个粒子的位置,  $v_i = (v_{i1}, v_{i2}, \dots, v_{id})^T \in \Omega$  表示第  $i$  个粒子的速度。微粒在搜索空间飞行过程中,用  $p_i = (p_{i1}, p_{i2}, \dots, p_{id})^T$  表示粒子  $i$  所经历的历史最优位置,称为 pbest,用  $p_g = (p_{g1}, p_{g2}, \dots, p_{gd})^T$  来表示所有粒子所经历过的最优位置,称为 gbest。对每一代,标准粒子群算法中第  $i$  个粒子的速度和位置采用如下公式进行更新<sup>[9]</sup>:

$$v_{id} = wv_{id} + c_1r_1(p_{id} - x_{id}) + c_2r_2(p_{gd} - x_{id}) \quad (2)$$

$$x_{id} = x_{id} + v_{id} \quad (3)$$

其中:  $w$  为惯性权重;  $c_1$  和  $c_2$  为加速常数;  $r_1$  和  $r_2$  为两个在  $[0, 1]$  内服从均匀分布的随机变量。式(2)的第一部分为惯性(inertia)部分,表示微粒先前速度对当前速度的引导;第二部分为认知(cognition)部分,表示粒子本身对它的引导;第三部分为社会(social)部分,表示粒子间的信息共享。这三个部分之间的相互平衡和制约决定了算法的主要性能。

为了进一步改善粒子的搜索能力,将式(3)作如下改动:

$$x_{id} = r_c \times x_{id} + (1 - r_c)v_{id} \quad (4)$$

其中:  $0 \leq r_c < 1$ 。式(4)实际上是对粒子的运动轨迹采用低通滤波器进行平滑,即所谓的动量算法<sup>[9]</sup>。考虑如下低通滤波器:

$$y(k) = r_c \times y(k-1) + (1 - r_c) \times w(k) \quad (5)$$

假设输入  $w(k)$  为  $1 + \sin(2\pi k/16)$ , 输出为  $y(k)$ 。图 1 为低通滤波器的平滑效果。图中虚线表示输入  $w(k)$ , 黑实线表示输出  $y(k)$ 。

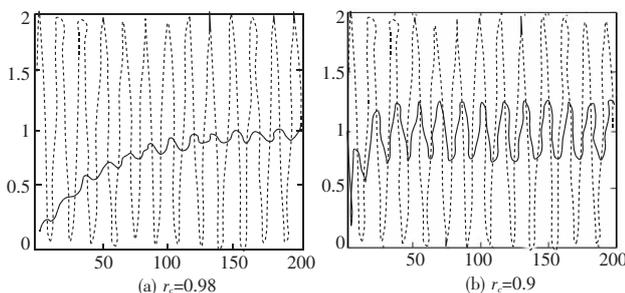


图 1 低通滤波器的平滑效果

可以从图 1 中看出,  $r_c$  越大,平滑效果越好。为了在粒子群算法初期防止粒子出现振荡现象,同时在后期保留粒子的多

样性,粒子能够进行精细的局部搜索,可以在早期设置较大的  $r_c$ ,搜索后期设置较小的  $r_c$ 。一般可设置  $r_c$  的最大值为 0.9,最小值为 0.4,在迭代过程中,  $r_c$  从最大值到最小值随步长线性变化。

### 2 最优交叉粒子群算法

PSO 是依靠群体之间的相互合作和竞争在问题空间中搜索最优解。由于算法本身没有交叉、变异等机制,所以当粒子群集中在某一局部最优解时,算法便很难再寻找到全局最优值,易出现早熟现象。于是很多学者通过将遗传算法中的交叉和变异算子引入粒子群算法来增加种群的多样性<sup>[8]</sup>,以提高算法的寻优性能。其中传统的实数交叉<sup>[10]</sup>公式为

$$\text{child1}(i, d) = r \times \text{parent1}(i, d) + (1 - r) \times \text{parent2}(i, d) \quad (6)$$

$$\text{child2}(i, d) = (1 - r) \times \text{parent1}(i, d) + r \times \text{parent2}(i, d) \quad (7)$$

其中:  $r$  为  $[0, 1]$  之间的随机数。当父代粒子  $\text{parent1}(i, d)$  和  $\text{parent2}(i, d)$  进行交叉时,子代个体  $\text{child1}(i, d)$  和  $\text{child2}(i, d)$  是被控制在  $\text{parent1}(i, d)$  与  $\text{parent2}(i, d)$  之间的范围内。当群体中粒子比较集中,需要增加其种群多样性来避免其陷入局部最优时,采用这种交叉方法并不能得到很好的效果。这表明算法引入交叉算子来提高群体多样性的策略受到了一定的限制,且交叉破坏了粒子向全局最优收敛的趋势,从而使得粒子群算法收敛速度变慢。因此,进一步改进粒子群的交叉方法来提高算法收敛性能显得尤为必要。

#### 2.1 模拟二进制交叉及一种新的交叉策略

遗传算法是通过模拟自然界中生物进化时的优胜劣汰规律来进行函数寻优<sup>[11]</sup>。在遗传算法中,种群通过交叉、变异和选择来搜索最优解。从种群中随机寻找两个个体作为父代个体,以一较大概率进行交叉操作;对每一个个体以一个较小概率进行变异操作,然后选择较优个体进入下一代,通过这样的迭代过程来寻找最优解。分析得出模拟二进制交叉在一定的参数设置条件下能同时起到交叉和变异的作用,因此本文将模拟二进制交叉<sup>[12]</sup>引入到粒子群算法中来提高算法的优化能力。

同时,为了减小交叉的盲目性,更加充分利用全局最优粒子对粒子群的引导作用,本文提出了一种新的交叉策略:将每一代中的每一个粒子和全局最优粒子进行模拟二进制交叉,并将交叉后的个体与父代个体进行比较,选择较好的个体进入下一代循环,使粒子以更快的速度收敛到当前全局最优解。这样就大大地提高了算法的收敛速度,随之也出现了相应的问题:由于粒子群快速向当前全局最优粒子收敛,使得算法的全局搜索能力下降。为了避免粒子收敛至局部最优,新算法在第 1 章采用一种变系数低通滤波器调节粒子搜索路径,然后在 2.2 节对惯性权重的选择作了一些要求。

模拟二进制交叉公式如下:

$$\text{child}(i, d) = \frac{1}{2} [(1 + r_u)P_{gd} + (1 - r_u)x(i, d)] \quad (8)$$

其中  $r_u$  为随机变量,由式(4)生成。

$$r_u = \begin{cases} (2u)^{\frac{1}{\eta+1}} & u < 0.5 \\ \frac{1}{[2(1-u)]^{\frac{1}{\eta+1}}} & u \geq 0.5 \end{cases} \quad (9)$$

在式(9)中,  $u$  是  $[0, 1]$  之间的均匀分布随机数;  $\eta$  是交叉参数(本文取为 2),为一常数。

对式(8)进行分析:当  $r_u$  取值在 1 的附近时,基于最优交叉策略的模拟二进制交叉使得粒子产生一个强加速,迅速地当前全局最优粒子靠拢,算法将快速收敛至当前最优解;当  $r_u$  取值较大时,模拟二进制交叉对粒子产生一个变异作用,从而保持了粒子的多样性,减小了算法早熟的可能性。图 2 为  $\eta = 2$  时  $r_u$  的取值范围。从图 2 中可以看出  $r_u$  以一个较大的概率取值在 1 的附近,以一个较小的概率取值偏离 1 的附近。结合上面最优模拟二进制交叉的分析得出:新算法在迭代过程中,种群中的每一个粒子将以一较大概率与当前最优粒子进行交叉,加速了粒子的收敛;以一较小概率进行变异,保持了粒子的多样性。这样,新算法通过模拟二进制交叉同时具有了遗传算法中的遗传与变异的作用。

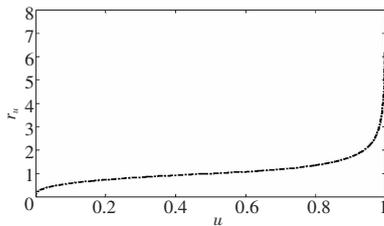


图 2  $\eta=2$  时  $r_u$  的取值范围

### 2.2 最优交叉粒子群算法惯性权重参数的设置

由速度更新公式(2)易得知,惯性权重  $w$  描述了微粒先前速度对当前速度的影响,控制其取值大小可调节 PSO 的全局与局部寻优能力<sup>[13]</sup>。当  $w$  较大时,粒子前一时速度对当前粒子的运动有很大的影响,粒子将保持原来的速度在全局范围内进行最优搜索,使得算法将更有利于全局搜索,但粒子群的局部搜索能力差;当权重参数较小时,粒子向个体最优和全局最优趋近时,速度更新公式中的认知和社会部分会迅速减小,直至降为 0,而由于惯性权重较小,速度更新公式中的惯性部分也会迅速趋近于 0,从而导致粒子速度快速降至 0,这样将使得算法迅速收敛至当前全局最优,算法的全局搜索能力降低。

Shi 等人<sup>[14]</sup>提出:当惯性权重从 0.9 线性下降至 0.4 时,可以加快算法的收敛速度,提高算法的收敛性能;但是当惯性权重  $w$  线性减小时, $w$  最大最小所经历的时间较短,全局搜索和局部搜索的时间均不够,从而限制了算法的寻优能力。动量最优交叉粒子群算法(OCPSO)使用模拟二进制交叉将每一个粒子和全局最优进行交叉,将使得算法以更快的速度收敛至当前搜索到的全局最优。为了避免粒子群收敛至局部最优,必须采取策略使算法迭代前期具有更强的全局搜索能力,后期具有进行细致局部搜索的能力。为此,改进的新算法采用一种新的权重参数设置策略:在前一半迭代过程中,采用较大的权重系数  $w_{max}$ ,使粒子群能在全局范围内进行搜索;在后一半迭代过程中,采用较小的权重系数  $w_{min}$ ,使粒子群能进行更加细致的局部搜索。

### 2.3 OCPSO 对粒子速度和位置的限制

在 OCPSO 中,使用  $v_{max}$  来限制粒子的飞行速度。如果对微粒的加速导致它在某一维的速度超过该维的最大速度  $v_{max,d}$ ,则该维的速度被限制为该维最大速度  $v_{max,d}$ 。当粒子速度变为 0 时,重新随机初始化速度:

$$\begin{aligned} &\text{if } v_{id} > v_{max,d} \text{ then } v_{id} = v_{max,d} \\ &\text{else if } v_{id} < -v_{max,d} \text{ then } v_{id} = -v_{max,d} \\ &\text{else if } v_{id} = 0 \text{ then } v_{id} = \text{rand}() \times x_{max,d} \end{aligned} \quad (10)$$

此外,当粒子飞出边界区域时,通常将边界位置赋予该粒子,粒子向反方向上飞行寻找最优位置:

$$\begin{aligned} &\text{if } x_{id} > x_{max,d} \text{ then } x_{id} = x_{max,d}, v_{id} = -v_{id} \\ &\text{else if } x_{id} < -x_{max,d} \\ &\text{then } x_{id} = -x_{max,d}, v_{id} = -v_{id} \end{aligned} \quad (11)$$

通过这种操作,既使得粒子保持在可行域内,又克服了粒子容易在边界堆积的情况,有利于算法的全局搜索。

### 2.4 最优交叉粒子群算法流程

针对上面的分析,给出完整的最优交叉粒子群算法流程:

- a) 初始化参数及粒子群(包括位置和速度),并将 pbest 初始化为粒子本身,gbest 初始化为第一代微粒群中适应度最高的粒子。
- b) 评价粒子的适应度。
- c) 更新粒子的 pbest、gbest。
- d) 根据式(2)(4)更新粒子的速度和位置。
- e) 判断粒子的速度、位置是否在限定范围内,否则根据式(10)(11)进行更新。
- f) 利用式(11)将更新后的粒子和全局最优粒子进行模拟二进制交叉,选择父代和交叉产生的子代中较好的粒子进入下一次循环。
- g) 若达到结束条件(通常规定一定的迭代次数或计算精度),则输出最优值;否则返回 b),继续循环迭代。

## 3 算法性能仿真分析

为了分析新算法的优化能力,本文选择近年来提出来的具有较强优化能力的改进 PSO 进行对比实验<sup>[6-8]</sup>,引入六个 Benchmark 优化问题(见附录)进行分析。其中  $f_3$  是很难优化的病态函数, $f_5$ 、 $f_6$  是具有大量局部最优的多峰函数<sup>[15]</sup>。为了能与文献[8]提出的 AEPSO1、AEPSO2,以及文献[6]提出的 DPSO,文献[7]提出的 HPSO 进行比较,将新算法放到与其相同的实验环境中,主要是指:种群规模为 20,采用附录中函数  $f_1 \sim f_6$ ,函数维度设为 30,进化代数设为 6 000 代,每个函数独立运行 50 次。其比较结果见表 1、2,其中 DPSO、HPSO、AEPSO1 和 AEPSO2 算法的结果数据来源于文献[8]。算法实验结果中的数值 0 表示 MATLAB 精度值  $1e-308$ (当 MATLAB 小于这个值时显示 0)。

表 1 OCPSO 与 DPSO、HPSO 的性能比较

函数	理论最小值	最小值			平均值			标准偏差		
		OCPSO	DPSO	HPSO	OCPSO	DPSO	HPSO	OCPSO	DPSO	HPSO
$f_1$	0	0(100%)	1.9e-11	2.4e-26	0	1.3e-10	3.9e-24	0	9.7e-11	4.9e-24
$f_2$	0	0(100%)	7.8e-04	0.2550	0	3.8e-03	1.41220	0	2.0e-03	0.8020
$f_3$	0	2.15e-05	8.0e-02	5.8e-02	3.7295	34.020	1.1e+02	30.62508	35.100	1.5e+2
$f_4$	0	0(100%)	4.4e-12	1.1e-16	0	2.5e-02	1.5e-02	0	3.0e-02	2.1e-2
$f_5$	0	0(100%)	7.2e-09	12.930	0	0.2010	29.070	0	0.4980	8.9630
$f_6$	0	0(42%)	82.350	1.9e+02	0.676078	1.5e+02	2.4e+02	2.802474	36.670	22.420

表 2 OCPSO 与 AEPSO1、AEPSO2 的性能比较

函数	理论最小值	最小值			平均值			标准偏差		
		OCPSO	AEPSO1	AEPSO2	OCPSO	AEPSO1	AEPSO2	OCPSO	AEPSO1	AEPSO2
$f_1$	0	0(100%)	4.1e-115	1.5e-128	0	1.4e-101	2.0e-122	0	9.1e-101	8.8e-122
$f_2$	0	0(100%)	1.4e-04	4.4e-11	0	2.1e-03	1.2e-09	0	2.3e-03	1.4e-009
$f_3$	0	2.15e-05	9.4e-03	2.8e-03	3.7295	40.130	14.000	30.62508	41.780	19.9100
$f_4$	0	0(100%)	0(20%)	0(40%)	0	1.2e-02	1.2e-02	0	1.3e-02	1.6e-002
$f_5$	0	0(100%)	1.2e-13	0(20%)	0	1.2340	0.5770	0	1.5770	1.08400
$f_6$	0	0(42%)	54.83	44.5700	0.676078	1.4e+02	1.2e+02	2.802474	41.630	39.3900

### 3.1 参数设置

在 OCPSO 中,首先,对于所有测试函数, $c_1 = 1.4$ , $c_2 = 1.4$ ,交叉参数  $\eta = 2$ , $w_{min} = 0.9$ ;其次,对于函数  $f_1$ 、 $f_2$ 、 $f_4$ 、 $f_5$ 、 $f_6$ , $w_{max} = 2$ ,对于病态函数  $f_3$ , $w_{max}$  取为较小的值 0.9,加强其局部

搜索能力。其他算法中参数设置如表 3 所示。

表 3 其他各粒子群算法的参数设置

算法	w	c <sub>1</sub>	c <sub>2</sub>	粒子数	其他
DPSO	0.7	1.4	1.4	20	c <sub>v</sub> =0.001
HPSO	0	[2.5,0.5]	[0.5,2.5]	20	
AEPSO1	[0.95,4]	1.4	1.4	20	K <sub>1</sub> =0.5, K <sub>2</sub> =10
AEPSO2	0.7	1.4	1.4	20	K <sub>1</sub> =0.5, K <sub>2</sub> =10

### 3.2 算法收敛结果对比及分析

表 1、2 显示了新算法 OCPSO 与 DPSO、HPSO、AEPSO1 和 AEPSO2 的对比实验。仿真结果表明:无论在最小值指标、均值指标还是标准差指标,本文提出的新算法在 f<sub>1</sub> ~ f<sub>6</sub> 的优化中的优化结果都得到了明显的改善。其中对于函数 f<sub>1</sub>、f<sub>2</sub>、f<sub>4</sub>、f<sub>5</sub> 和 f<sub>6</sub>,新算法均获得了理论最优值。在 50 次运行中,函数 f<sub>1</sub>、f<sub>2</sub>、f<sub>4</sub>、f<sub>5</sub> 收敛到理论最优的概率达到了 100%,函数 f<sub>6</sub> 收敛到理论最优的概率也达到了 42%。这说明新算法无论是在一般函数优化问题中,还是在多峰函数优化问题中都具有很好的全局收敛性和鲁棒性,而且搜索精度也大幅度提高。新算法对于很难极小化的病态函数 f<sub>3</sub> 的优化也取得了很好的效果,这表明了新算法在处理病态函数时也有很好的收敛性能。

### 3.3 算法收敛速度对比与分析

图 3 是迭代过程中各函数的收敛轨迹图。

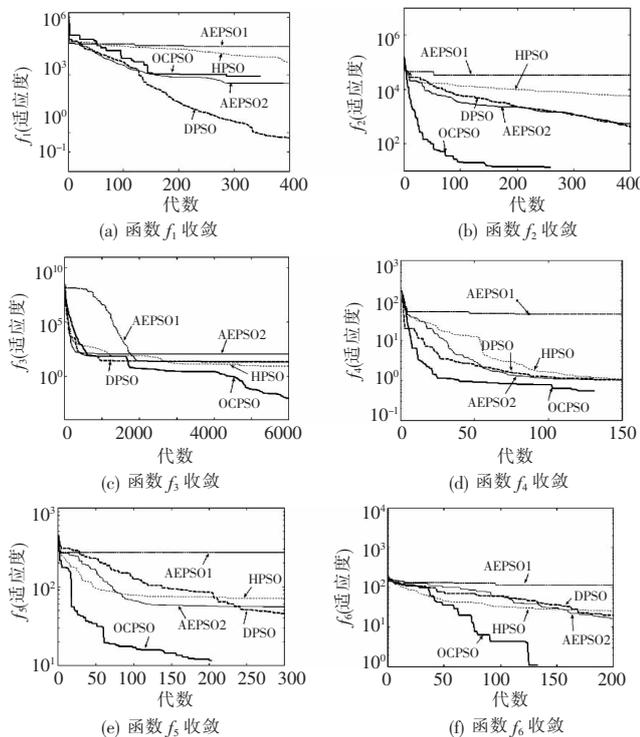


图 3 各函数收敛轨迹图

由图 3(a) ~ (f) 得到:以前的一些改进算法 DPSO、HPSO、AEPSO1、AEPSO2 在后期收敛速度非常慢,而新算法具有更快的收敛速度和更好的全局寻优能力。函数 f<sub>1</sub> 经过 360 代、f<sub>2</sub> 经过 260 代、f<sub>4</sub> 经过 580 代、f<sub>5</sub> 经过 200 代、f<sub>6</sub> 经过 130 代均已收敛至 0 (理论最优值),病态函数 f<sub>3</sub> 也取得了较好的优化结果。由此说明新算法具有很快的收敛速度和很强的收敛性,能有效地避免遗传算法和粒子群算法的早熟问题。

另外,3.2 和 3.3 节的对比实验还表明:

a) DPSO 对微粒速度或位置引入了一个小概率随机变异操作来增强种群多样性,使算法能够有效地进行全局搜索,但是必须适当选择变异概率才能获得较好的收敛速度,实验中通

过选择适当的变异概率使 DPSO 算法获得了较好的性能。

b) HPSO 消除了惯性权重,算法的全局寻优能力降低,不能有效地逃离局部极小点,图 3 中(a)(b)(c)(e)表明 HPSO 容易过早收敛到局部极值。

c) 实验结果表明,AEPSO2 的性能要好于 AEPSO1。这是因为 AEPSO2 中逃逸速度的引入改变了微粒的搜索行为,使其具有一个较快的搜索速度;而 AEPSO1 开始时权重参数过大,使微粒不能详细搜索,而后期权重参数过小,使粒子速度趋近于 0,从而不能以较大步长进行局部搜索,该结论与文献[8]中结论一致。

d) OCPSO 算法采用交叉策略增强粒子的多样性,而不同迭代阶段设置不同的权重参数和迭代过程所采用的低通滤波器参数由大变小都使得粒子群从全局范围内进行搜索逐步过渡到精细的局部搜索,从而使 OCPSO 具有更快的收敛速度和更高的精度。实验中 f<sub>3</sub>、f<sub>5</sub>、f<sub>6</sub> 等一般算法难以优化的函数仿真中,OCPSO 都展示了其强收敛特性,验证了 OCPSO 的优越性。

## 4 结束语

本文针对传统粒子群算法收敛速度慢、全局搜索能力差的缺点,提出了一种新的粒子交叉策略,将每一代微粒群中的粒子与全局最优进行模拟二进制交叉,在分析权重参数对算法收敛性影响的前提下提出了新算法权重参数的设置策略,并且采用动量方式调节粒子搜索路径。通过实验仿真发现,新算法具有很强的收敛特性和全局收敛能力,是一种有效的优化算法。

### 附录

#### Benchmark 函数

1) f<sub>1</sub> (Tablet)  $f_1 = 10^6 x_1^2 + \sum_{i=2}^n x_i^2$ , 搜索空间:  $-100 \leq x_i \leq 100, d = 30$ ; 全局最小值  $f_1(x^*) = 0$ , 其中  $x^* = 0, 0, \dots, 0$ 。

2) f<sub>2</sub> (Quadratic)  $f_2 = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$ , 搜索空间:  $-100 \leq x_i \leq 100, d = 30$ ; 全局最小值  $f_2(x^*) = 0$ , 其中  $x^* = 0, 0, \dots, 0$ 。

3) f<sub>3</sub> (Rosenbrock)  $f_3 = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$ , 搜索空间:  $-50 \leq x_i \leq 50, d = 30$ ; 全局最小值  $f_3(x^*) = 0$ , 其中  $x^* = 1, 1, \dots, 1$ 。

4) f<sub>4</sub> (Griewank)  $f_4 = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(x_i/\sqrt{i}) + 1$ , 搜索空间:  $-300 \leq x_i \leq 300, d = 30$ ; 全局最小值  $f_4(x^*) = 0$ , 其中  $x^* = 0, 0, \dots, 0$ 。

5) f<sub>5</sub> (Rastrigin)  $f_5 = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)$ , 搜索空间:  $-5.12 \leq x_i \leq 5.12, d = 30$ ; 全局最小值  $f_5(x^*) = 0$ , 其中  $x^* = 0, 0, \dots, 0$ 。

6) f<sub>6</sub> (Schaffer)  $f_6 = \sum_{i=1}^{n-1} (x_i^2 + x_{i+1}^2)^{0.25} \times [\sin(50 \times (x_i^2 + x_{i+1}^2)^{0.1}) + 1.0]$ , 搜索空间:  $-100 \leq x_i \leq 100, d = 30$ ; 全局最小值  $f_6(x^*) = 0$ , 其中  $x^* = 0, 0, \dots, 0$ 。

### 参考文献:

[1] KENNEDY J, EBERHART R C. Particle swarm optimization [C] // Proc of IEEE International Conference on Neural Network. [S. l.]: IEEE Press, 1995:1942-1948.

[2] EBERHART R C, KENNEDY J. A new optimizer using particle swarm theory [C] // Proc of the 6th International Symposium on Micro Machine and Human Science. [S. l.]: IEEE Press, 1995:39-43.

[3] 张小红, 宁红梅. 基于混沌粒子群和模糊聚类的图像分割算法 [J]. 计算机应用研究, 2011, 28(12): 4786-4789.

X、Y 方向相对误差如图 6、7 所示。

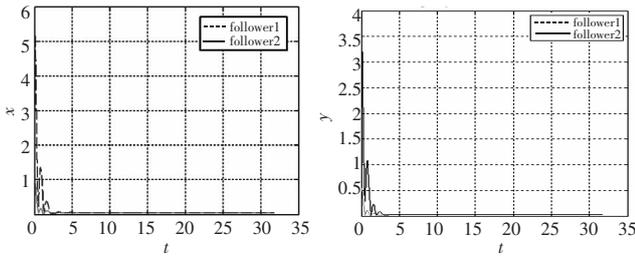


图 6 Agent x 方向上位置相对误差 图 7 Agent y 方向上位置相对误差

由上面的仿真结果可以看出,该队形能在较短时间内 ( $t = 2\text{ s}$ ) 达到预期的队形,在形成队形之后,能够维持队形运行,稳定时 X 坐标相对误差为 4.977%, Y 坐标相对误差为 5.188%。

### 2.2 避障机动过程

以三角队形为例,队形数据中  $l_{1,x} = l_{2,x} = 20, l_{1,y} = -30, l_{2,y} = 30$ 。仿真结果如图 8~11 所示。

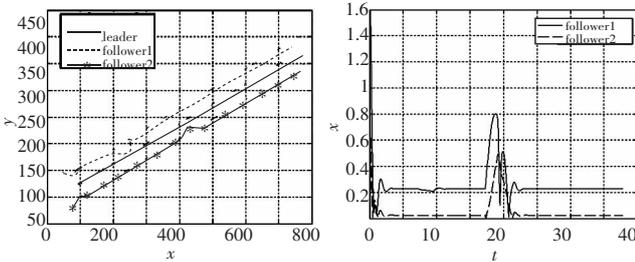


图 8 Leader-follower 位置变化 图 9 Agent x 方向上位置相对误差

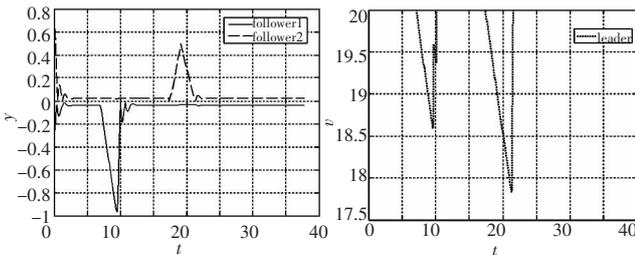


图 10 Agent y 方向上位置相对误差 图 11 避障时 leader 速度大小变化

由仿真实验可以看出,agent 能够及时躲避障碍,在避障过程中,由于在目标方向上 leader 与 follower 之间距离增大,为了防止掉队的现象,leader 会相应地降低自身的速度,避障结束之后,又能较快地恢复到以前的队形。

### 3 结束语

本文通过对已有的一种基本队形的控制方法进行改进,将 leader-follower 算法思想应用其中,并结合基于行为的方法,在其中加入避障行为。为了防止掉队现象,增加了 follower 状态对 leader 的反馈,使 leader 能够根据当前 follower 与自身的状态来动态地改变自身的属性,以确保系统能够稳定地运行。最后对改进后的队形收敛及避障行为进行了仿真实验验证和分析。

在实验中发现,在采取避障行为之后,队形中 follower 的位置误差会增加,而 leader 速度的改变不平滑。在下一步的工作中需要对这两个问题进行研究,使其能更加适应战术仿真的需要。

#### 参考文献:

- [1] 任德华,卢桂章. 对队形控制的思考[J]. 控制与决策,2005,20(6):601-604.
- [2] LAWTON J R, YOUNG B J, BEARD R W. A decentralized approach to elementary formation maneuvers[C]//Proc of IEEE International Conference on Robotics & Automation. Piscataway, NJ: IEEE Press, 2000:2728-2733.
- [3] KHATIB O. Real-time obstacle avoidance for manipulators and mobile robots[J]. Int J of Robotics Research,1986,5(1):90-98.
- [4] LUO Xiao-yuan, HAN Na-ni, GUAN Xin-ping. Leader-following consensus protocols for formation control of multi-agent network[J]. Journal of Systems Engineering and Electronics,2011,22(6):991-997.
- [5] HENGSTER-MOVRI C K, BOGDAN S, DRAGANJAC I. Multi-agent formation control based on bell-shaped potential functions[J]. Intell Robot System,2011,58(2):165-189.
- [6] 李强,刘国栋. 多移动机器人的队形控制[J]. 计算机系统应用,2011,20(4):80-85.
- [7] 李强,刘国栋. 基于行为的机器人自适应队形控制[J]. 计算机系统应用,2011,20(4):177-181.
- [8] 宋艳萍,王兴平,张术东,等. 基于势函数的护卫队形控制[J]. 海军航空工程学院学报,2011,26(6):622-627.
- [9] 刘新科,李大鹏,谭乐祖,等. 基于制导与控制一体化的导弹编队队形控制[J]. 火力与指挥控制,2011,36(8):208-210.
- [10] 张峰,王金超,陈雪波. 机器人队形控制中的二叉树方法[J]. 辽宁科技大学学报,2010,33(5):456-459.
- [11] 黄天云,王晓楠,陈雪波. 基于队形控制的多机器人时间最优搬运方法[J]. 系统仿真学报,2010,22(6):1442-1446,1465.
- [12] 李波,王祥凤. 基于动态 leader 多机器人队形控制[J]. 长春工业大学学报:自然科学版,2009,30(2):210-214.

(上接第 4462 页)

- [4] 曾宇容,王林,富庆亮. 基于 DE 和 PSO 的混合智能算法及其在模糊 EOQ 模型中的应用[J]. 计算机应用研究,2012,29(2):438-441.
- [5] LØVBJERG M, RASMUSSEN T K, KRINK T. Hybrid particle swarm optimizer with breeding and subpopulations[C]//Proc of International Conference on Genetic and Evolutionary Computation Conference. San Francisco: Morgan Kaufmann, 2001:101-106.
- [6] XIE Xiao-fang, ZHANG Wen-jun, YANG Zhi-lian. A dissipative particle swarm optimization[C]//Proc of IEEE International Conference on Evolutionary Computation. [S. l.]: IEEE Press, 2002:1456-1461.
- [7] RATNAWEERA A, HALGAMUGE S K, WATSON H C. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients[J]. IEEE Trans on Evolutionary Computation, 2004,8(3):240-255.
- [8] 赫然,王永吉,王青,等. 一种改进的自适应逃逸微粒群算法及实验分析[J]. 软件学报,2005,16(12):2036-2044.

- [9] 马瑞新,刘宇,覃征,等. 求解约束优化问题的动量粒子群算法[J]. 系统仿真学报,2010,22(11):2485-2488.
- [10] MICHALEWICZ Z. Genetic algorithms + data structures = evolution programs[M]. Berlin: Springer, 1996:117-119.
- [11] GOLDBERG D E. Genetic algorithms in search, optimization and machine learning[M]. Boston: Addison-Wesley, 1989:212-214.
- [12] DEB K, AGRAWAL R B. Simulated binary crossover for continuous search space[J]. Complex Systems, 1995,9(4):115-148.
- [13] KRINK T, VESTERSTRON J S, RIGET J. Particle swarm optimization with spatial particle extension[C]//Proc of Congress on Evolutionary Computation. Washington DC: IEEE Computer Society, 2002:1474-1479.
- [14] SHI Yu-hui, EBERHART R C. A modified particle swarm optimizer[C]//Proc of IEEE International Conference on Evolutionary Computation. [S. l.]: IEEE Press, 1998:69-73.
- [15] 王小平,曹立明. 遗传算法——理论、算法与软件实现[M]. 西安: 西安交通大学出版社, 2002:105-107.