五种粒子群优化模型效率的研究

王娟勤,何东健,孙建敏
WANG Juan-qin,HE Dong-jian,SUN Jian-min

西北农林科技大学 信息工程学院,陕西 杨凌 712100
College of Information Engineering,Northwest Agriculture & Forestry University,Yangling,Shaanxi 712100,China
E-mail:wangjq@nwsuaf.edu.cn

WANG Juan-qin,HE Dong-jian,SUN Jian-min.Research of effectiveness of five particle swarm optimization models. Computer Engineering and Applications,2008,44(33):62-65.

Abstract: The basic particle swarm optimization algorithm is identified as five types of PSO according to its cognition component and social component value,such as PSO Full-Model,PSO Cognitive-Only Model,PSO Social-Only Model,PSO Selfless Model and PSO Selfless Full-model.Compare five PSO models' effectiveness and efficiency according to their success rate,average function evaluation and their best fitness by applying parameter set and using five benchmark functions.The result is that Full-Model and Selfless Full Model with *K* are effective in solving the functions with high dimension,Social Model and Selfless Model without *K* are also effective in solving the functions with less dimension such as Schaffer function.
Key words: Particle Swarm Optimization(PSO);effectiveness;benchmark functions;best fitness

摘 要:粒子群优化算法按照认知部分和社会部分被区分为5种模型(完全模型、自认知模型、社交模型、非自身社交模型和非自身完全模型)。为了明确5种粒子群优化模型的效率,选用进化计算领域中常用的5种基准函数,分别对5种粒子群优化算法模型设置不同的参数,分析了它们在求解5种基准函数时的成功率、平均函数求值数、最佳适应度等。结果表明:PSO完全模型和非自身完全模型使用收缩系数 *K* 在某些参数设置下求解高维问题时即搜索问题的解时效率较高,社交模型和非自身社交模型在一些参数设置下求解Schaffer函数等二维问题的效率最好。
关键词:粒子群优化算法;效率;基准函数;最佳适应度
DOI:10.3778/j.issn.1002-8331.2008.33.020 **文章编号**:1002-8331(2008)33-0062-04 **文献标识码**:A **中图分类号**:TP18

1 引言

粒子群优化算法(Particle Swarm Optimization,简写为PSO)由Russ Eberhart和James Kennedy于1995年提出[1-2],它是一个基于群体的、随机技巧的优化算法,现在已经演化出很多种版本,并已得到广泛应用,它不仅被用来解决非线性问题,还用在解不可微、多峰值等复杂优化问题中。

PSO属于进化计算的一种,因此,它与其他的进化算法如遗传算法有很多相似之处[3],与遗传算法一样,PSO用随机的方法对于选定的人口进行初始化、赋值,以待优化问题目标函数的适应值为依据,但它利用社会群体中的历史认知影响向解的方向收缩。粒子在整个过程中不会死亡(即淘汰)。在PSO中,每一个粒子的位置实际就是一个潜在的问题的解,当粒子被创建时赋予随机值,并赋予粒子飞行的速率,这些粒子将在飞行空间中利用自身的历史飞行认知和邻居最佳的认知调整自身的位置和飞行速率,一步一步向解的方向收缩。

粒子群优化算法中参数众多,有外在的和内在的参数,一些参数或值的选择会影响PSO的效率和可靠性,一些则没有影响,那么,什么样的参数或值的组合会形成一个好的、高效的PSO呢?在一些文献[4-5]里已经做了一些研究,如"一个通用的PSO"[6]。但对于不同类的问题,不同的PSO表现出的效率是不同的。在本文中,选用进化计算领域中常用的5种基准函数,对5种粒子群优化算法模型设置不同的参数,分析了它们在求解5种基准函数时的成功率、平均函数求值数、最佳适应度等,目的是为使用PSO求解问题提供一种参考模型。

2 粒子群优化算法

在PSO中,每个粒子由以下五个部分组成:

(1)$\boldsymbol{x}$ 表示粒子当前的位置,是待解问题或函数潜在的解。如第 i 个粒子的位置用 $X_i=(x_{i1},x_{i2},x_{i3},\cdots,x_{id})$ 来表示。d 表示待解问题或函数有 d 维,例如函数 $f(x,y,z)$,即该函数有三维。

(2)$\boldsymbol{v}$ 表示粒子的飞行速率。如第 i 个粒子的飞行速率用 $V_i=(v_{i1},v_{i2},v_{i3},\cdots,v_{id})$ 来表示。

(3)Fitness(适应度)是对 $\boldsymbol{x}$ 的一个评价,是将得到 $\boldsymbol{x}$ 的代到函数中产生的值。

作者简介:王娟勤(1973-),女,讲师,主要研究方向:智能信息处理;何东健(1957-),男,博士生导师,教授,主要研究方向:智能化检测与控制,计算机应用技术;孙建敏(1969-),男,讲师,主要研究方向:计算机应用技术。
收稿日期:2007-12-12 **修回日期**:2008-05-17

（4）*Pbest* 为最佳适应度。记录粒子在飞行过程中找到的最佳位置所产生的适应度。

（5）**p** 用来存储粒子在前面飞行中产生 *Pbest* 的 **x** 值。

按照 Kennedy 最早的版本[2]，粒子群优化算法的基本步骤总结如下：

（1）对于选定粒子群中的每个粒子 $X_i=(x_{i1},x_{i2},x_{i3},\cdots,x_{id})$，用随机的方法赋予初始值，并赋予粒子初始的飞行速率 $V_i=(v_{i1},v_{i2},v_{i3},\cdots,v_{id})$。

（2）粒子在搜索空间飞行，每步飞行速率用下式来更新：

$$v_{id}=v_{id}+\phi_1*rnd_1()*(p_{id}-x_{id})+\phi_2*rnd_2()*(p_{gd}-x_{id}) \quad (1)$$

式中：ϕ_1、ϕ_2 为两个正的常数，由这两个参数决定 $rnd_1()*(p_{id}-x_{id})$ 和 $rnd_2()*(p_{gd}-x_{id})$ 在此公式中的权重；$rnd_1()$、$rnd_2()$ 为两个得到（0，1）区间随机数的函数；p_{id} 为 i 粒子的历史最佳位置；p_{gd} 为所有粒子中有最佳适应度的粒子的位置。

（3）用粒子的速率计算粒子下一步将要移动的位置，即

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

（4）返回到第 2 步重复计算，直到达到所设置的极限值或函数求值数超过设置的最大函数求值数。

Shi 和 Eberhart 的研究表明[7]，使用式（1）的 PSO 在搜索大范围空间时是高效的，但对局部搜索缺乏精细，他们在论文中引入了惯量因子 ω，用其来动态调整粒子的速率，让 PSO 逐渐地集中于局部搜索，引入 ω 的公式为：

$$v_{id}=\omega*v_{id}+\phi_1*rnd_1()*(p_{id}-x_{id})+\phi_2*rnd_2()*(p_{gd}-x_{id}) \quad (3)$$

一般 ω 初始取 0.9，并使其随迭代次数的增加而线性递减至 0.4，这样便可先侧重全局搜索，使搜索空间快速收敛于某一区域，然后采用局部精细搜索以获得高精度的解。

Maurice Clerc[8]定义了另外一种收缩因子 K，有

$$K=\frac{2}{|2-\phi-\sqrt{\phi^2-4\phi}\,|},\phi=\phi_1+\phi_2>4.0 \quad (4)$$

当在算法中应用 K 时，粒子群优化算法的公式（1）可变换为：

$$V_{id}=K*(V_{id}+\phi_1*rnd_1()*(P_{id}-X_{id})+\phi_2*rnd_2()*(P_{gd}-X_{id})) \quad (5)$$

PSO 参数众多，如 ϕ_1、ϕ_2 的取值、粒子的人口数、粒子与相邻粒子的交流方式及所有粒子对自身位置的更新采用同步更新还是异步更新等等。设置何种参数或值会使 PSO 在求解一类问题时表现出较稳定和较高的效率，是本文研究的目的。

3　研究对象与方法

3.1　五种 PSO 模型

从公式（1）可以看出，粒子新的速率被 3 个因素所影响：一个是它前一次的速率 v_{id}；一个是粒子的认知部分：$\phi_1*rnd_1()*(p_{id}-x_{id})$；另一个是粒子的社交部分：$\phi_2*rnd_2()*(p_{gd}-x_{id})$。

$(p_{id}-x_{id})$ 为粒子当前的位置与粒子历史最优位置之间的距离，这可以看作是粒子自己的认知、思考，这也表示了粒子的前一个历史最佳位置将要影响多少粒子下一步移动的距离。

社交部分表示粒子之间的相互影响、相互协作，$(p_{gd}-x_{id})$ 表示粒子第 d 维当前位置与所有粒子中有最佳适应度的粒子第 d 维位置之间的距离，它表示所有粒子中有最佳适应度的粒子的位置将要影响多少粒子下一步的移动距离。

调整 ϕ_1、ϕ_2 就可以调整认知部分和社交部分对粒子移动的影响成份，Kennedy[9]已经说明了基于等式（1）的由于 ϕ_1 和 ϕ_2 不同而区分的 4 种不同的 PSO 模型，即完全模型、自认知模型、社交模型和非自身社交模型。

（1）完全模型（Full Model）：$\phi_1,\phi_2>0$，粒子的速率计算如式（1）所示。

（2）自认知模型（Cognitive Only Model）：$\phi_1>0,\phi_2=0$，去除社交的认知部分，就象每个粒子被隔绝而靠它自己的思维去寻找解。速率计算公式如下式所示：

$$v_{id}=v_{id}+\phi_1*rnd_1()*(p_{id}-x_{id}) \quad (6)$$

（3）社交模型（Social Only Model）：$\phi_1=0,\phi_2>0$，省略了自认知因子，每个粒子在寻找解时只参照所有粒子中有最佳适应度的粒子的位置去调整自己下一步的位置。

$$v_{id}=v_{id}+\phi_2*rnd_2()*(p_{gd}-x_{id}) \quad (7)$$

（4）非自身社交模型（Selfless Model）：这种模式与第（3）种模式类似，不同的是在本模式下 $g\neq i$。

$$v_{id}=v_{id}+\phi_2*rnd_2()*(p_{gd}-x_{id})\text{ 且 }g\neq i \quad (8)$$

提出一种与完全模型相似的非自身完全模型（Selfless Full Model），在本模式下 $g\neq i$，其公式为：

$$v_{id}=v_{id}+\phi_1*rnd_1()*(p_{id}-x_{id})+\phi_2*rnd_2()*(p_{gd}-x_{id})\text{ 且 }g\neq i \quad (9)$$

3.2　基准函数

为了比较 5 种 PSO 算法的效率，选用如下 5 种基准函数，这些基准函数经常被应用在进化计算领域中。

（1）Sphere function

$$f(x)=\sum_{i=1}^{n}x_i^2 \quad (10)$$

（2）Rosenbrock function

$$f(x)=\sum_{i=1}^{n}(100(x_{i+1}-x_i^2)^2+(x_i-1)^2) \quad (11)$$

（3）Generalized Rastrigrin function

$$f(x)=\sum_{i=1}^{n}(x_i^2-10\cos(2\pi x_i)+10) \quad (12)$$

（4）Generalized Griewank function

$$f(x)=\frac{1}{4\,000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})+1 \quad (13)$$

（5）Schaffer's F6 function

$$f(x,y)=0.5+\frac{(\sin\sqrt{x^2+y^2})^2-0.5}{(1.0+0.001(x^2+y^2))^2} \quad (14)$$

为上述 5 种函数定制常用的参数[4,10]及可接受的误差值，如表 1 所示。

表 1　函数参数

函数	函数维数	X_i 的取值范围	可接受的误差值
Sphere	30	[−100，100]	<0.01
Rosenbrock	30	[−30，30]	<100
Rastrigrin	30	[−5.12，5.12]	<100
Griewank	30	[−600，600]	<0.1
Schaffer F6	2	[−100，100]	<0.000 01

3.3　研究方法

设置粒子群优化算法公式中的参数如表 2 所示。

在试验中，5 种 PSO 模型的粒子拓扑结构为星形，每个粒子数据的更新采用异步更新法，当粒子的速率或 X_i 的值超过最大值时，设置两个量的值为边缘值[10]，设置 $V_{max}=X_{max}$，人口数

表 2 PSO 模型参数

模式	(ϕ_1,ϕ_2)	粒子数
Full Model and Selfless Full Model	(2.05,2.05);(2.8,1.3)	
Cognitive Only Model	(2.05,0);(2.8,0);(4.1,0)	100;50;30;15;5
Social Only Model and Selfless Model	(0,2.05);(0,4.1)	

设为子集{100,50,30,15,5}，观察人口数对 PSO 五种模型效率的影响。

对完全模型和非自身完全模型，设置 ϕ_1=2.8，ϕ_2=1.3[1]和 ϕ_1=2.05，ϕ_2=2.05[9]。在自认知模型中，因为 ϕ_2=0，所以测试 ϕ_1=2.8、2.05 和 4.1 的情况；在社交型模型和非自身社交模型中 ϕ_1=0，设置 ϕ_2=2.05 和 4.1。

为了证明收缩系数 K 在 PSO 中的作用，对 5 种 PSO 使用收缩系数去求解 5 种基准函数，设置 $\phi=\phi_1+\phi_2$，ϕ=4.1，观察它们的效率。

对每一组参数进行一次试验，用 5 种 PSO 模型一一对每个基准函数进行求解，每次求解过程运行 50 次，对基准函数平均求值数上限设为 100 000 次。对于每一组参数设置，在 50 次运行过程中，记录每次运行得到的成功率、函数平均求值数、最佳适应度和函数求值标准方差等数据。

4 结果与分析

本文将成功率作为一个重要指标去衡量算法或算法在某些参数设置下的效率，在成功率相同的情况下，以平均函数求值数来衡量，平均函数求值数小，说明效率高。本文只列出了成功率大于 50%的结果，因为较小的成功率说明这种参数设置在这种模式中或此模式本身不适合解这类问题。

设 Sphere、Rosenbrock、Rastrigrin 和 Griewank 函数的维数为 30 维，如表 1 所示，目的是找出该函数的最小值。试验发现，用 PSO 解这 4 种方程时产生非常相近的结果：当不用收缩系数 K 时，5 种 PSO 模型在所有参数设置下得到的成功率都为0。

而在解这 4 种基准函数式(10)~(13)时使用收缩系数 K，完全模型和非自身完全模型都有较高的效率，在一些参数设置下可以得到几乎 100%的成功率。社交模型和非自身社交模型在解这 4 种基准方程时较好于自认知模型，且两种模型结果类似，社交模型在一些参数设置下解 Griewank 函数(13)时可以得到大于 80%的成功率，但用该模型在所有参数设置下解另外三种函数(10)~(12)时效率较差，所得到的最好成功率为38%。自认知模型在求解此 4 种基准函数时都未能求得解，即成功率都为 0。

表 3~表 7 分别给出了 PSO 在解基准方程时成功率在 50%以上的算法和参数设置，在各表中，"SR"表示"成功率"，"Arg-F"表示"平均函数求值数"，"SD"表示函数计算中的标准偏差，"Arg-BF"表示平均最佳适应度。

从试验结果可以得出，带收缩系数 K 的完全模型和非自身完全模型都可以得到大于 50%的成功率，且这两种模型优于其他 3 种模型。比较这两种模型在 ϕ_1、ϕ_2 设置不同参数时所表现的效率，结果表明：当 ϕ_1=2.8，ϕ_2=1.3 时它们所表现出的效率优于其他参数设置，此时的平均函数求值数较小。同时也得出，如果设置算法中的粒子数为 5，在解函数(10)~(13)时成功率较低。所以对于完全模型和非自身完全模型，粒子数也是影响这两种算法效率的一个很重要的因素。另外，本研究还发现，这两种完全模型在其他参数设置下表现出相似的效率。

表 3 用 PSO 模式解 Sphere 函数的结果(成功率>50%)

模式	Sphere 函数						
	PN	ϕ_1	ϕ_2	SR/(%)	Avg-F	SD	Avg-BF
	15	2.8	1.3	100	6 273.26	639.44	0.009 617
Full Model with K	30	2.8	1.3	100	7 695.92	609.46	0.009 244
	15	2.05	2.05	100	9 755.10	1 123.00	0.009 287
	50	2.8	1.3	100	10 781.72	462.04	0.009 265
	100	2.8	1.3	100	17 732.42	787.34	0.009 190
	50	2.05	2.05	100	22 701.48	1 749.11	0.009 430
	100	2.05	2.05	100	39 838.06	3 267.55	0.009 076
	30	2.05	2.05	96	18 807.76	16 791.90	400.008 800
	5	2.05	2.05	56	57 005.52	38 745.60	5 207.701 000
Selfless Full Model with K	15	2.8	1.3	100	6 283.92	903.77	0.009 503
	30	2.8	1.3	100	7 859.32	574.25	0.009 235
	50	2.8	1.3	100	11 048.28	577.69	0.009 210
	100	2.8	1.3	100	18 237.94	1006.38	0.009 208
	50	2.05	2.05	100	23 465.16	1 781.23	0.009 195
	100	2.05	2.05	100	40 347.36	3 310.94	0.009 171
	30	2.05	2.05	96	19 412.24	16 680.10	400.008 900
	15	2.05	2.05	90	19 116.78	27 281.80	1 000.008 000
	5	2.05	2.05	58	52 635.14	40 909.20	5 400.006 000

表 4 用 PSO 模式解 Rosenbrock 函数的结果(成功率>50%)

模式	Rosenbrock 函数						
	PN	ϕ_1	ϕ_2	SR/(%)	Avg-F	SD	Avg-BF
Full Model with K	100	2.8	1.3	98	24 793.32	21 257.50	96.44
	50	2.8	1.3	92	21 007.02	25 489.80	217.92
	15	2.8	1.3	88	24 244.22	32 463.30	238.59
	30	2.8	1.3	84	25 019.74	33 707.60	3 825.11
	30	2.05	2.05	72	40 583.30	38 424.30	7 525.24
	50	2.05	2.05	68	51 269.38	36 890.80	14 690.40
	100	2.05	2.05	66	60 689.52	31 416.00	5 700.98
	15	2.05	2.05	58	49 391.86	43 885.40	18 285.20
Selfless Full Model with K	100	2.8	1.3	92	24 986.28	23 175.80	99.78
	50	2.8	1.3	92	22 428.60	26 054.20	274.33
	30	2.8	1.3	92	18 818.28	26 201.70	167.62
	15	2.8	1.3	82	29 493.56	37 336.80	2 038.31
	15	2.05	2.05	72	37 949.10	40 053.10	9 281.01
	30	2.05	2.05	70	42 731.34	38 410.40	12 987.34
	50	2.05	2.05	68	48 663.10	36 539.70	5 989.13

表 5 用 PSO 模式解 Rastrigrin 函数的结果(成功率>50%)

模式	Rastrigrin 函数						
	PN	ϕ_1	ϕ_2	SR/(%)	Avg-F	SD	Avg-BF
Full Model with K	30	2.8	1.3	100	7 015.80	2 180.84	95.876 65
	50	2.8	1.3	100	10 543.62	2 603.66	96.063 70
	100	2.8	1.3	100	17 878.62	4 850.71	94.450 89
	100	2.05	2.05	100	21 561.10	3 973.53	96.731 39
	50	2.05	2.05	98	13 567.58	12 646.30	96.747 07
	15	2.8	1.3	96	7 623.08	18 878.90	99.212 14
	30	2.05	2.05	92	14 837.42	25 151.20	98.690 85
	15	2.05	2.05	70	32 966.56	43 905.00	106.865 20
Selfless Full Model with K	15	2.8	1.3	100	4 625.52	1 089.84	96.882 30
	30	2.8	1.3	100	8 618.54	2 544.23	95.354 30
	50	2.8	1.3	100	11 444.12	2 682.03	95.563 95
	50	2.05	2.05	100	13 512.48	3 308.18	95.738 86
	100	2.05	2.05	100	24 770.06	6 804.48	96.159 48
	100	2.8	1.3	100	24 886.02	6 645.66	95.588 08
	30	2.05	2.05	96	11 947.14	18 083.40	97.184 14
	15	2.05	2.05	76	27 690.12	40 655.30	102.003 50

表 6　用 PSO 模式解 Griewank 函数的结果(成功率>50%)

模式	Griewank 函数						
	PN	ϕ_1	ϕ_2	SR/(%)	Avg-F	SD	Avg-BF
Full Model with K	50	2.8	1.3	100	10 207.62	6 485.97	0.000 005
	100	2.8	1.3	100	13 712.14	6 727.15	0.000 005
	100	2.05	2.05	98	15 101.76	15 277.80	0.000 199
	30	2.8	1.3	92	16 897.94	25 775.00	0.000 782
	50	2.05	2.05	90	20 564.06	28 732.40	0.000 977
	15	2.8	1.3	74	37 899.20	42 318.80	0.002 337
	30	2.05	2.05	72	33 167.06	42 255.60	0.002 724
Selfless full Model with K	100	2.05	2.05	100	14 283.52	10 372.00	0.000 005
	50	2.8	1.3	98	12 424.44	13 861.00	0.000 199
	100	2.8	1.3	98	15 578.18	13 064.00	0.000 200
	50	2.05	2.05	96	11 535.10	18 719.00	0.000 234
	30	2.8	1.3	88	23 213.96	30 851.00	0.001 170
	30	2.05	2.05	84	24 557.54	34 470.00	0.001 559
	15	2.8	1.3	68	38 049.06	44 572.00	0.003 113
	15	2.05	2.05	64	44 634.30	45 087.00	0.003 501
Social Only Model with K	100	0	4.1	84	27 958.78	34 933.90	0.001 557
	50	0	4.1	70	37 751.60	42 439.90	0.002 917
	30	0	4.1	58	45 653.94	47 009.20	0.004 083
Selfless Model with K	100	0	4.1	82	28 112.16	34 613.70	0.001 752
	50	0	4.1	72	34 847.68	41 593.60	0.002 723
	30	0	4.1	62	43 856.76	45 658.90	0.003 694
	15	0	4.1	50	52 322.76	48 274.60	0.004 860

Schaffer 函数(14)为二维,当用 5 种 PSO 模型去求解该方程时,得到与解其他 4 种基准函数不同的有趣的结果。当使用收缩系数 K 时,得到与解其他 4 种基准函数类似的结论:完全模型和非自身完全模型优于其他 3 种模型,自认知模型在求解问题时效率是最差的,从来没有找到过解。社交型和非自身社交模型在解 Schaffer 函数时比解其他 4 种方程表现出较高的效率。

然而,当不用收缩系数 K 时,得到完全不同的结果,所有的 PSO 模型都得到 100%的成功率,甚至自认知模型,而它在解其他 4 种基准方程时的成功率为 0。表 7 给出了成功率为 100%的结果,对于求解 Schaffer 函数,不用收缩系数 K 比用收缩系数 K 的 PSO 有更高的效率。表 7 也表明,在成功率为 100%的情况下,社交模型和非自身社交模型在函数平均求值数和标准偏差方面优于其他 3 种,完全模型和非自身完全模型在不同参数设置的情况下表现出相似的结果,对于社交模型和非自身社交模型,设置参数 ϕ_2=4.1 比设置其他参数得到较好的结果。

5　结论

本研究结果表明,在 5 种 PSO 模型中,完全模型和非自身完全模型在合适的参数设置下总是表现出较高的效率,即粒子用较少的飞行搜索次数就可以发现问题的解,在平均函数求值数和函数标准偏差方面,完全模型微优于非自身完全模型,这是因为完全模型 PSO 利用了所有粒子(包括自己)发现的最好位置向解的方向收缩。而自认知模型在 5 种模型中效率是最差的,这说明粒子群优化算法的认知部分和社交部分对于指导 PSO 向解的方向收缩起到了很重要的作用。完全模型和非自身完全模型在解高维问题如基准函数(10)~(13)时推荐使用收缩系数 K,且对于这两种模型,设置参数:人口数为 50,ϕ_1=2.8,ϕ_2=1.3 可以使这两种粒子群优化算法在搜索问题的解时发挥较高的效率。

对于维数较小的问题,如 Schaffer 函数(14),社交模型和非自身社交模型在成功率相同的情况下平均函数求值数较小,

表 7　用不带收缩系数 K 的 PSO 模式 Schaffer 函数的结果(成功率 100%)

模式	Schaffer F6 函数						
	PN	ϕ_1	ϕ_2	SR/(%)	Avg-F	SD	Avg-BF
Full Model without K	15	2.05	2.05	100	435.66	454.26	0
	30	2.05	2.05	100	498.10	420.95	0
	50	2.05	2.05	100	569.78	533.88	0
	15	2.8	1.3	100	572.66	710.19	0
	30	2.8	1.3	100	605.30	396.11	0
	50	2.8	1.3	100	695.62	508.76	0
	100	2.05	2.05	100	774.92	458.21	0
	100	2.8	1.3	100	1 089.54	620.70	0
Selfless Full Model without K	15	2.8	1.3	100	455.46	366.33	0
	30	2.05	2.05	100	534.88	514.50	0
	15	2.05	2.05	100	548.08	627.20	0
	100	2.05	2.05	100	647.88	406.06	0
	50	2.8	1.3	100	660.00	521.01	0
	30	2.8	1.3	100	693.26	549.38	0
	50	2.05	2.05	100	714.62	619.83	0
	100	2.8	1.3	100	1 094.76	593.02	0
Cognitive Only Model without K	15	4.1	0	100	565.02	448.01	0
	30	4.1	0	100	637.88	402.69	0
	5	2.8	0	100	678.82	470.92	0
	50	4.1	0	100	858.66	424.72	0
	100	4.1	0	100	1 029.18	549.23	0
	15	2.8	0	100	1 076.96	606.07	0
	30	2.8	0	100	1 139.34	694.72	0
	50	2.8	0	100	1 647.48	848.63	0
	100	2.8	0	100	2 345.72	1 299.28	0
	15	2.05	0	100	2 460.18	1 416.18	0
	5	2.05	0	100	2 657.58	2 782.67	0
	30	2.05	0	100	3 676.82	2 092.12	0
	50	2.05	0	100	4 314.30	3 079.97	0
	100	2.05	0	100	6 726.20	3 513.74	0
Social Only Model without K	15	0	4.1	100	145.76	96.77	0
	30	0	4.1	100	211.46	133.31	0
	50	0	4.1	100	246.90	97.34	0
	100	0	4.1	100	419.62	129.56	0
	15	0	2.05	100	767.46	710.06	0
	30	0	2.05	100	768.66	724.03	0
	50	0	2.05	100	890.04	746.62	0
	100	0	2.05	100	1 196.56	877.74	0
Selfless Model without K	15	0	4.1	100	182.62	234.14	0
	30	0	4.1	100	190.28	109.90	0
	50	0	4.1	100	274.70	101.47	0
	100	0	4.1	100	416.74	104.44	0
	15	0	2.05	100	748.88	864.80	0
	50	0	2.05	100	814.82	621.83	0
	100	0	2.05	100	963.56	646.18	0
	30	0	2.05	100	1 058.84	794.54	0

而在解二维问题中,收缩系数 K 是完全没有必要的,相反会影响 PSO 的效率。所以对于维数较小的问题,可以采用不使用收缩系数而最大可能地利用相邻粒子发现的最优位置向问题的解逼近。

参考文献:

[1] Eberhart R,Kennedy J.A new optimizer using particle swarm theory[C]//Proc of the Sixth International Symposium on Micro Machine

(下转 68 页)