

基于混合迁移行为的自组织迁移算法^{*})

林志毅 李元香 王玲玲

(武汉大学软件工程国家重点实验室 武汉 430072)

摘要 自组织迁移算法(Self-organizing migrating algorithm, SOMA)是一种新型的进化算法。在对基本的自组织迁移算法分析的基础上提出了基于混合迁移行为的自组织迁移算法(Hybrid migrating behavior based self-organizing migrating algorithm, HBSOMA)。该算法通过在个体迁移过程中引入了多种迁移方式,形成混合迁移行为,使得个体的行为变得多样化,增加了种群多样性,加速了群体在多峰复杂空间中的寻优进程。仿真结果显示,该算法优于原自组织迁移算法。

关键词 进化算法,自组织迁移算法,混合迁移行为, HBSOMA

Hybrid Migrating Behavior Based Self-organizing Migrating Algorithm

LIN Zhi-yi LI Yuan-xiang WANG Ling-ling

(State Key Lab. of Software Engineering, Wuhan University, Wuhan 430072, China)

Abstract Self-organizing migrating algorithm (SOMA) is a new evolutionary algorithm. This paper proposed a hybrid migrating behavior based self-organizing migrating algorithm(HBSOMA) based on the analysis of the basic SOMA, which employs the multiple migrating modes in the individual migration procedure. It increases the diversity of the population and speeds up the population-based search process in the multi-modal complex space. Simulation results reveal that the proposed algorithm is better than the original self-organizing migrating algorithm.

Keywords Evolutionary algorithm, Self-organizing migrating algorithm, Hybrid migrating behavior, HBSOMA

1 引言

Zelinka 和 Lampinen 于 2000 年发展了一个算法,即自组织迁移算法(Self-Organizing Migrating Algorithm,简称 SOMA)^[1]的新型进化算法。与多数进化算法一样,SOMA 也是一种基于群体的随机优化算法,但与传统的进化算法不同,SOMA 在演化过程中并不产生新的代,也就是说新的代的产生并不是基于进化原理,而是基于社会环境下群体的自组织行为,如社会性动物的合作觅食行为。从这一意义上说,SOMA 和粒子群算法^[2]、蚁群算法^[3]也可归于集群智能的范畴。

SOMA 算法具有描述简单,易于应用的特点。SOMA 能应用到所有整数、离散和连续变量函数的优化,同时也能优化带有多个非平凡约束的非线性目标函数。此外,SOMA 算法所需要调整的参数也比较少,易于并行化^[4],是一类较有潜力的新型进化算法。SOMA 在寻优过程中,很大程度上受早期发现的较好解的影响,这些较好解以极大的概率引导 SOMA 走向局部最优解。为了避免陷入局部最优解,同时又能加快算法的寻优速度,本文研究了一种基于混合迁移行为的自组织迁移算法(hybrid migrating behavior based self-organizing migrating algorithm, HBSOMA)。该算法通过在个体迁移过程中引入多种迁移方式,使得个体的寻优行为变得多样化。这样一来,在较好保持种群的多样性的同时,可以加速群体在多峰复杂空间中的寻优进程。结果表明,该算法性能优于基本 SOMA。

本文的结构如下:第 2 节介绍基本的 SOMA 算法;第 3

节介绍了 HBSOMA 算法,并分析了它的特点;第 4 节用实例对 HBSOMA 进行了验证;最后是结论和展望。

2 自组织迁移算法

SOMA 是一个基于合作竞争策略的优化算法。在生物群体中,合作与竞争往往是并存的。如一群动物在寻找食物时,若某一个体率先发现食物而成为群体中的领先者,群体中其它个体得到此信息后,往往会改变其运动方向,向领先者所在位置前进。如在搜索过程中,某个体比先前的领先者更为成功(如发现更多或更好的食物),群体中其它个体往往会再次改变其运动方向,转而向新的领先者所在位置前进^[5]。SOMA 从上述“合作-竞争”行为中得到启发,通过寻优群体在问题空间中的自组织迁移运动,逐步达到或接近最优解。

2.1 问题描述与编码

为了简单起见,本文只考虑如下函数优化问题:

$$\text{Minf}(X) \quad (1)$$

其中 $X = (X_1, X_2, \dots, X_n)^T \in R^n$

$D = \{X \in R^n \mid l_i \leq X_i \leq u_i, i=1, 2, \dots, n\}$, $f(X)$ 为目标函数。

SOMA 采用浮点编码,迭代寻优。

2.2 SOMA 的运行过程

SOMA 在不停的循环中工作,这个循环叫做迁移循环(Migration loops,简称 ML)。在 SOMA 运行过程进行之前,先产生一群个体。在每一次迁移的循环中,最好的个体被选择出来,叫做领导者(Leader),在群中的其他个体都在解空间

^{*}基金项目:863 计划项目 2007AA01Z290,国家自然科学基金项目 60773009,湖北省自然科学基金 2007ABA009。林志毅 博士生,主要研究方向为自然计算、无线传感器网络。

中朝向这个领导者运动,运动采用小步跳跃(Step)的形式,步长参数 Step 向领先者跳跃,每跳跃一步都根据目标函数对当前位置进行评价并重新生成 PRTVector 向量,从而形成一条由个体 i 当前位置指向领先者位置的搜索轨迹。跳跃一直持续到 PathLength(最大迁移路径长度)所限定的位置。在整个运动结束后,个体把在运动中遇到的最好的位置选择出来,如果这个最好的位置比个体的原来位置好,则个体就迁移到新位置,否则个体原地不动。在所有的个体都移动以后,新的领导者被选出来,然后进行下一次的迁移。

迁移过程的描述如下

Procedure 个体迁移

```
begin
    t ← Step;
    while t ≤ PathLength do
        生成 PRTVector;
        for j ← 1 to n do

$$X_{i,start}^{temp} = X_{i,start}^{ML} + (X_i^{ML} - X_{i,start}^{ML}) * t * PRTVector$$

            end
            计算  $f(X_{temp})$ ;
            if  $f(X_{temp}) > f(X_i^{ML})$  then
                 $X_i^{ML+1} ← X_{temp}$ 
                 $f(X_i^{ML+1}) ← f(X_{temp})$ 
            end
            t ← t + Step
        end
end
```

其中 X_i^{ML+1} 表示群体中在第 $ML+1$ 次迁移过程后的第 i 个个体, $X_{i,start}^{ML}$ 是第 i 个个体迁移开始时的位置。 X_i^{ML} 表示群体中在第 ML 次迁移过程结束后选择的最佳个体。 $t ∈ (0, PathLength)$, t 以步长 Step 为间隔长度,最大取 PathLength。PRT 参数通常定义为 $[0, 1]$ 区间内的实数, PRTVector 是一个依赖于 PRT 的向量。如果在 $(0, 1)$ 的随机数小于 PRT, 那 PRTVector 就设为 1, 否则为 0。

2.3 SOMA 的基本流程

基本 SOMA 的步骤如下:

(1)参数定义。算法运行前,先要定义运行所必要的参数,例如:Step, PathLength, NP(个体数), ML(最大迁移循环次数), PRT 等。

(2)粒子初始化。在解空间内初始化产生 NP 个粒子, $M=0$ 。

(3)根据目标函数评价每一个体所对应的目标函数值。

(4)确定当前领先个体 L(通常采用目标函数值最小的个体)。

(5)迁移过程。按照上面介绍的过程对整个种群(当前领先个体 L 除外)进行迁移操作。

(6)判断结束条件,如领先者与最差个体之间目标函数值的差的绝对值大于预定的 MinDiv 且未达到最大迁移代数 ML,则转到步骤(3),否则向下执行步骤(7)。

(7)算法结束。返回最好个体的值,或者根据情况返回所有个体的值。

SOMA 基本参数的分析和设置可以见参考文献[6,7]。

3 HBSOMA

与其它全局优化算法一样, SOMA 也会出现过早收敛的

情况,即在算法运行过程中,如果某个个体发现一个当前最优位置,其它个体将迅速向其靠拢,如果该最优位置为一局部最优解, SOMA 就无法在解空间内重新搜索,算法将陷入局部最优。从多态性角度分析,主要原因是基本 SOMA 没有考虑个体的行为差异,使所有个体都朝目前的最优解靠近。如能在现有的 SOMA 中,引入混合迁移行为,即采用多种迁移方式,使得个体在迁移过程并不总是朝着最优解的方向前进,也向着其它可能的方向前进的话,算法就更有可能跳出局部解。为此,本文提出了基于混合迁移行为的自组织迁移算法 HB-SOMA,其迁移过程形式化描述如下:

Procedure 个体迁移

```
begin
    t ← Step;
    while t ≤ PathLength do
        生成 PRTVector;
        CR = rand(0, 1); //生成  $[0, 1]$  区间内的随机数
        if rand(0, 1) > CR then
            for j ← 1 to n do

$$X_{temp} = X_{i,start}^{ML} + (X_i^{ML} - X_{i,start}^{ML}) * t * PRTVector$$

                end
            end
            else
                找出与当前个体  $i$  不同的两个个体  $v1, v2$ ;
                for j ← 1 to n do

$$X_{temp} = X_{i,start}^{ML} + (X_{v1,start}^{ML} - X_{v2,start}^{ML}) * t * PRTVector$$

                end
            end
            计算  $f(X_{temp})$ ;
            if  $f(X_{temp}) > f(X_i^{ML})$  then
                 $X_i^{ML+1} ← X_{temp}$ 
                 $f(X_i^{ML+1}) ← f(X_{temp})$ 
            end
            t ← t + Step
        end
end
```

由于个体在迁移过程中引入了多种迁移方式,个体迁移的行为多样化,不再总是向最优个体方向迁移,而是朝着各种可能的方向前进,这样一来,增加了种群的多样性,使得个体更有可能并且更快跳出局部最优,从而加速了群体在多峰复杂空间中的寻优进程。

4 实验结果与分析

为了验证和比较算法的性能,选择了一组标准测试函数^[8]进行仿真实验,其函数基本特征见表 1,基本 SOMA 的策略参数设置如下: PRT = 0.10, PathLength = 2.1, Step = 0.21, NP = 50, ML = 500, MinDiv = 0.000001。HBSOMA 的参数设置为迁移方式选择概率 CR 在 $(0, 1)$ 之间动态变化,其余同基本 SOMA。终止条件都设为评价次数不超过 300000 次。所有实验均在一台 Pentium4 3.40G/1G 内存的计算机上运行,由于算法具有随机性,对每个函数均运行 20 次,其平均结果见表 2。图 1 和图 2 绘出了基本 SOMA 和 HBSOMA 在函数 2 和函数 3 上的寻优动态曲线(20 次独立运行的平均结果)。

表1 测试函数 F1—F5

函数名称	函数表达式(求最小值)	维数	变量范围	最优值
Sphere 函数	$f(X) = \sum_{i=1}^n x_i^2$	30	$[-100, 100]^n$	0
Griewank 函数	$f(X) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	$[-600, 600]^n$	0
Rastrigin 函数	$f(X) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)$ $f(X) = 0.5 +$	30	$[-5.12, 5.12]^n$	0
Schaffer 函数	$\frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2}$	2	$[-100, 100]^n$	0
Ackley 函数	$f(X) = 20 + e - \frac{1}{20e} \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} - \frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)$	100	$[-30, 30]^n$	0

表2 HBSOMA 与 SOMA 的比较

f	f _{min}	最优解的平均值		达到最优解的次数		平均函数评价次数	
		HBSOMA	SOMA	HBSOMA	SOMA	HBSOMA	SOMA
F1	0	0.0000	0.0000	20	20	39639	49503
F2	0	0.0000	0.0070	20	9	57777	152722
F3	0	0.0995	2.6870	18	1	124802	261141
F4	0	0.0000	0.0011	20	18	27485	44396
F5	0	0.0440	3.8805	19	0	198402	238881

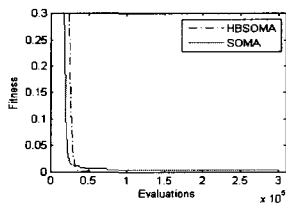


图1 函数 F2 算法的平均收敛曲线

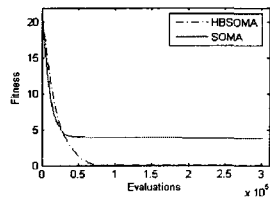


图2 函数 F5 上算法的平均收敛曲线

从上述实验数据可以得出以下结论:

(1)表1清晰地展示了 HBSOMA 的稳定性。HBSOMA 的收敛比率远远高于 SOMA 算法, HBSOMA 在 5 个标准测试函数上的 20 次运行几乎每次都能成功地达到最优解。SOMA 在单峰函数或低维函数表现稳定,但在振荡强烈的多峰函数上则容易陷入局部最优解。

(2)图1和图2显示 HBSOMA 的收敛速度也相当快。在前期 HBSOMA 的收敛速度与 SOMA 接近,而在后期明显要优于 SOMA。这主要是因为 HBSOMA 能够一直较好地保持多样性,从而为迁移操作提供更有效的操作。而 SOMA 前

期虽然收敛很快,但到后期由于多样性的缺失,收敛速度就大大降低。

(3)从表1同样可以看出,除函数 F1 外, HBSOMA 所需要的代价(平均评价次数)要比 SOMA 算法少得多。这说明 HBSOMA 能以较小的代价获得较好的结果,主要原因是它采用了多种迁移方式,较好地维持了多样性。

从以上分析可以看出,基于混合迁移行为的自组织迁移算法要优于基本的自组织迁移算法。

结束语 本文提出了基于混合迁移行为的自组织迁移算法 HBSOMA,多种迁移方式的运用使得个体的行为具有多样性,在保证种群多样性的同时,也加速了算法在复杂函数下的寻优过程。通过对经典数值优化问题的测试,显示出优异的性能,能以较大概率搜索到全局最优解。我们下一步的研究目标是研究算法的收敛性并将它应用在其他领域的优化;此外,算法参数的设置和其它迁移方式对本算法性能的影响也是下一步实验的重点。

参考文献

- [1] Ivan Z, Jouni L. SOMA-Self-Organizing Migrating Algorithm// Mendel, 6th International Conference on Soft Computing. Brno, Czech Republic, 2000
- [2] Clerc M, Kennedy J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space [J]. IEEE Trans on Evolutionary Computation, 2002, 6(1): 58-73
- [3] Dorigo M, Maniezzo V, Colnari A. Ant system: optimization by a colony of cooperating agents. IEEE Transaction on Systems, Man and Cybernetics-Part B, 1996, 26(1): 29-41
- [4] Červinka M, Ivan Z. Parallel computation for SOMA[R]// Proceeding 19th European Conference on Modeling and Simulation. Riga, Latvia, 2005
- [5] Zelinka I. SOMA-Self-organizing migrating algorithm [M]. New Optimization Techniques in Engineering. Berlin, Springer, 2004: 167-217
- [6] Oplatková Z, Zelinka I. Investigation On Shannon-Kotelnik Theorem Impact On Soma Algorithm Performance [A]// Proceedings 19th European Conference on Modelling and Simulation. Yuri Merkurjev, Richard zobel, Eugène Kerckhoffs. ECMS [C], 2005
- [7] 柯晶, 李歧强, 乔谊正. 采用随机变异步长的改进自组织迁移算法. 计算机工程与应用[J], 2006, 35: 41-44
- [8] Eberhart R C, Shi Y H. Evolving Artificial Neural Networks[R] // Proceedings of Int'l Conference on Neural Networks and Brain. Beijing, 1998

(上接第 141 页)

- [9] Lamperti G, Zanella M. Flexible diagnosis of discrete-event systems by similarity-based reasoning techniques. Artificial Intelligence, 2006, 170(3): 232-297
- [10] Cordier M-O, Grastien A. Exploiting independence in a decentralised and incremental approach of diagnosis//Proc. 20th International Joint Conference on Artificial Intelligence (IJCAI-07). Hyderabad, India, 2007: 292-297
- [11] Schumann A, Pencolély Y. Scalable diagnosability checking of event-driven systems//Proc. 20th International Joint Conference on Artificial Intelligence (IJCAI-07). Hyderabad, India, 2007: 575-580
- [12] Cerutti S, Lamperti G, Scaroni M, et al. A diagnostic environment for automaton networks. Software-Practice & Experience, 2007, 37(4): 365-415

- [13] Lamperti G, Zanella M. On monotonic monitoring of discrete-event systems//Proc. 18th International Workshop on Principles of Diagnosis (DX-07). Nashville, TN, USA, 2007: 130-137
- [14] Alur R, Dill D L. A theory of timed automata. Theoretical Computer Science, 1994, 126(2): 183-235
- [15] Tripakis Z S. Fault diagnosis for timed automata // Lecture Notes in Computer Science, 2469. 205-221
- [16] Brusoni V, Console L, Terenziani P, et al. A spectrum of definitions for temporal model-based diagnosis. Artificial Intelligence, 1998, 102 (1): 39-79
- [17] Allen J. Maintaining knowledge about temporal intervals. Communications of the ACM, 1983, 26(11): 832-843
- [18] Panati A, DupréD T. State-based vs. simulation-based diagnosis of dynamic systems//Proc. 14th European Conference on Artificial Intelligence (ECAI-00). Berlin, Germany, 2000: 176-180