

基于反向学习的自组织迁移算法

林志毅¹ 王玲玲²

(广东工业大学计算机学院 广州 510006)¹ (武汉大学软件工程国家重点实验室 武汉 430072)²

摘 要 为了解决自组织迁移算法存在的早期收敛问题,提出了基于反向学习的自组织迁移算法(Opposition-based Self-organizing Migrating Algorithm, OSOMA)。该算法利用反向学习机制扩展了个体的搜索方向,获得了更优秀的采样个体,使得算法在保持多样性的同时提高了收敛速度。此外,该算法还对步长进行自适应调整,进一步平衡了算法的勘探和开采能力。通过典型函数的测试证实了 OSOMA 的有效性。

关键词 自组织迁移算法,反向学习,OSOMA

中图法分类号 TP18 **文献标识码** A

Opposition-based Self-organizing Migrating Algorithm

LIN Zhi-yi¹ WANG Ling-ling²

(Faculty of Computer, Guangdong University of Technology, Guangzhou 510006, China)¹

(State Key Lab. of Software Engineering, Wuhan University, Wuhan 430072, China)²

Abstract A new opposition-based self-organizing migrating algorithm(OSOMA) was proposed to deal with premature convergence of self-organizing migrating algorithm. The key points of OSOMA lie in: 1) the opposition-based learning is applied to extend the migrating direction and obtain better individual, which maintains diversity of population and improves the convergence speed. 2) the algorithm adaptively adjusts the step to further balance between the ability of exploration and exploitation capacity. Then, OSOMA is used to solve typical problems and numerical results show the effectiveness of OSOMA.

Keywords Self-organizing migrating algorithm, Opposition-based learning, OSOMA

1 引言

2000 年, Zelinka 和 Lampinen 共同发展了一类称为自组织迁移算法(Self-Organizing Migrating Algorithm, 简称 SOMA)^[1]的新型演化算法。SOMA 模拟动物的觅食过程,是一种基于群体的随机优化算法,可用于非线性、可微和多峰等复杂问题的优化。SOMA 具有较快的收敛速度,但是它在寻优的过程中,很大程度上受早期发现的较好解的影响,这些较好解将以较大的概率引导 SOMA 走向局部最优解,从而导致早熟收敛^[2]。

为克服 SOMA 易于早熟收敛的问题,提高算法的性能,一些研究者对其进行了不同的尝试。如柯晶等在个体迁移过程中引入了随机变异步长^[3]; Leandro dos Santos Coelho 利用高斯算子来增加算法的局部微调能力^[4]; 笔者提出了基于混合迁移行为的自组织迁移算法 HBSOMA^[2]。这些都一定程度地提高了 SOMA 算法的性能,但它们主要考虑 SOMA 的多样性或收敛速度的某一方面。

为避免算法过早陷入局部最优解的同时加快算法的收敛速度,本文提出了一种基于反向学习的自组织迁移算法(Opposition-based Self-organizing Migrating Algorithm, OSO-

MA)。该算法在个体迁移过程中对种群中的每个个体进行反向学习,不仅扩展了个体的搜索方向,提高了种群的多样性,而且能促使演化生成过程的跃变,加快算法的收敛速度。此外,对步长进行自适应调整,可在前期加快算法的收敛速度,后期加强算法的局部搜索能力。结果表明,这种算法性能优于基本的 SOMA 和 HBSOMA。

本文第 2 节详细介绍反向学习、OSOMA 的主要迁移过程和算法流程;第 3 节用典型测试函数验证 OSOMA;最后为结束语。

2 OSOMA

2.1 反向学习

反向学习(Opposition-based Learning, OBL)的概念最初是由 Hamid R. Tizhoosh 提出的^[5],之后被用于强化学习^[6]、多目标遗传算法^[7]等,取得了较好的效果。

OBL 的主要思想是:在考虑每个候选个体的同时考虑其反向个体,从而得到一个更接近最优解的个体。事实上,根据概率学原理,每个随机产生的候选个体相比它的反向个体有 50% 的概率远离或靠近问题最优解。如果采样时引入基于反向学习的机制,为每个采样所得的个体产生相对应的反向个

到稿日期:2011-06-11 返修日期:2011-09-04 本文受国家自然科学基金项目(60773009),广东工业大学校博士基金(093058)资助。

林志毅(1979-),男,博士,主要研究方向为自然计算、模式识别, E-mail: lzy2911@hotmail.com; 王玲玲(1979-),女,博士,主要研究方向为自然计算、图像处理。

体,然后从采样个体及其反向个体中挑选适应度最优的个体产生下一代个体,将有助于改进优化过程的收敛速率。

首先,给出了两个定义^[5]:

定义 1(反向数, Opposite Number) 令实数 $x \in [a, b]$, 其反向数为 $\bar{x} = a + b - x$ 。

定义 2(反向点, Opposite Point) 令 $X = (x_1, x_2, \dots, x_n)$ 为 n 维空间中的一点,其反向点 \bar{X} 定义为 $\bar{X} = (\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)$, 其中 $\bar{x}_i = a_i + b_i - x_i$, $x_i \in R$ 且 $x_i \in [a_i, b_i]$, $i = 1, \dots, n$ 。当 X 为自组织迁移算法中的个体时, \bar{X} 也称为反向个体。

由以上定义,基于 OBL 的优化过程如下:令 $X = (x_1, x_2, \dots, x_n)$ 为 n 维空间中的一点, $f(\cdot)$ 为评价函数。根据反向点的定义, $\bar{X} = (\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)$ 为 $X = (x_1, x_2, \dots, x_n)$ 的反向点。如果 $f(\bar{X}) < f(X)$ (求最小值问题), 则用 \bar{X} 代替 X , 并将 X 从候选点中删除。

2.2 OSOMA 的主要迁移过程

在迁移过程中, SOMA 中的个体只朝着当前最优解的正方向跳跃, 即跳跃的范围只能在 $(0, Pathlength)$ 。这种迁移操作可能导致算法朝着当前最优解的正方向收缩, 而忽略了负方向上的一些搜索空间, 不利于全局优化。根据概率学原理, 每个随机产生的候选解相比它的反向解有 50% 的概率机会远离或靠近问题最优解。也就是说, 反方向也有可能就是最优解的所在。那么, 如果在现有的 SOMA 中, 通过个体的反向学习获得反向个体, 不仅可能获得更好的解, 进而加快算法的收敛速度, 而且将使个体的跳跃方向发生改变, 使算法的迁移范围更广, 一定程度上扩大了种群的多样性, 从而提高了算法的全局搜索能力。

此外, 原始 SOMA 将步长 $step$ 设置为一个固定的值, 实验显示, 一般取 0.11 可以使算法的效果令人较为满意。但是将 $step$ 设置为一个固定值, 会使得在每次迁移循环过程中搜索的路径较为单一, 粒子前进步长都为 $step$ 的整数倍, 不利于更精细的局部搜索。实验表明, 较大的 $step$ 具有较强的勘探能力, 但是会导致算法早熟收敛; 较小的 $step$ 具有较强的开采能力, 但会导致收敛速度下降。因此, 本文中对步长采取自适应变化, 初期用稍大的步长, 随着进化代数的增加, 步长逐渐减少到固定值。具体见式(1)。

为此, 本文提出了基于反向学习的自组织迁移算法, 其迁移过程形式化描述如下:

```

Procedure 个体迁移
Begin
    t ← 0; // 控制参数, t ∈ (0, PathLength), 步长为 step
    if rand(0, 1) < p0 then // 反向学习, p0 为反向学习概率
        for i = 1 to n do // n 为种群规模
            根据 OBL(定义 2) 得到每个个体的反向个体 opi, 并计算其适应值。
            if (f(opi) < f(Xi))
                Xi ← opi;
                f(Xi) ← f(opi);
            end
        end
    else // SOMA
        if step > 0.1 then // step 为迁移的步长
            step = step * 0.99;
        end
        for i = 1 to n
            while t ≤ PathLength do // PathLength 为迁移的最大步长值

```

```

生成 PRTVector;
for j ← 1 to dim do // dim 为所求问题的维数
    Xtemp = Xi, startML + (XLML - Xi, startML) * t * PRTVector (2)
end
计算 f(Xtemp);
if f(Xtemp) > f(XiML) then
    XiML+1 ← Xtemp
    f(XiML+1) ← f(Xtemp)
end
t ← t + step
end // while
end // for
end // else
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。 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 $(0, 1)$ 范围的随机数小于 PRT , 那 $PRTVector$ 就设为 1, 否则为 0。

在上述个体的迁移过程中, 引入了反向学习机制。该机制不仅扩大了个体搜索的方向, 增加了种群多样性, 而且有利于获得更优秀的解, 加快算法的收敛速度。在种群中采用步长的自适应调整, 则有助于算法进一步保持勘探和开采能力的平衡。

2.3 OSOMA 的流程

在上述迁移过程的基础上, OSOMA 的流程如下:

- (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 、 ML 、 PRT 等。
- (2) 粒子初始化。在解空间内初始化产生 NP 个粒子。
- (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 。
- (5) 迁移过程。按照上面介绍的个体迁移过程对整个种群中的每个个体进行迁移操作。
- (6) 判断结束条件, 如果满足则转到步骤(4), 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。

3 实验结果与分析

为了验证 OSOMA 的有效性, 选择了一组标准测试函数进行实验, 其函数基本特征见表 1, 并与基本 SOMA、HBSOMA 的求解结果进行了对比。其中基本 SOMA 的策略参数设置如下: $PRT=0.10$, $PathLength=2.1$, $step=0.21$, $NP=20$, $ML=5000$ 。HBSOMA 的参数设置为迁移方式, 选择概率 $CR=0.5$, 其余同基本 SOMA。OSOMA 的参数设置为 $p_0=0.25$, 其余参数同基本 SOMA。终止条件都设置为评价次数不超过 $1e6$ 次或精度达到 $1e-18$ 。所有实验均在一台 Intel Core2 2.10G/2G 内存的计算机上运行。由于算法具有随机性, 因此对每个函数均运行 20 次, 其平均结果见表 2(黑体数字表示 3 种算法的最优值)。图 1 和图 2 绘出了基本 SOMA 和 HBSOMA 在函数 2 和函数 5 上的寻优动态曲线(20 次独立运行的平均结果。为了方便比较, 函数 2 的评价次数只到 $3e5$ 为止)。

(下转第 233 页)

tion with structured syntactic knowledge[C]//COLING-ACL, 2006:41-48

Measuring the Relatedness of Concepts[EB/OL]. www.aaai.org, 2004

[11] Lesk M. Automatic sense disambiguation using machine readable dictionaries; How to tell a pine cone from a ice cream cone [C]//Proceedings of SIGDOC '86, 1986

[12] Banerjee S, Pedersen T. An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet [C] // CILing 2002. LNCS 2276, 2002; 136-145

[13] Pedersen T, Patwardhan S, Michelizzi J. WordNet::Similarity-

[14] Collins M, Duffy N. Covolution kernels for natural language [C]// NIPS. 2001; 625-632

[15] Zhou Guo-dong, Kong Fang, Zhu Qiao-ming. Context-sensitive convolution tree kernel for pronoun resolution [C] // IJCNLP. 2008

[16] 孔芳, 朱巧明, 周国栋, 等. 基于中心理论的指代消解研究[J]. 计算机科学, 2009, 36(6)

(上接第 218 页)

表 1 测试函数 F1-F6

函数表达式(求最小值)	维数	变量范围	最优值
$F_1(x) = \sum_{i=1}^n x_i^2$	30	[-5, 12, 5, 12]	0
$F_2(x) = \sum_{i=1}^n i * x_i^2$	30	[-5, 12, 5, 12]	0
$F_3(x) = \sum_{i=1}^n i * x_i^4 + \text{random}[0, 1]$	30	[-1, 28, 1, 28]	0
$F_4(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	30	[-5, 12, 5, 12]	0
$F_5(x) = 20 + e^{-20e^{-\frac{1}{5} \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}}} - e^{-\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)}$	30	[-32, 32]	0
$F_6(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$	30	[-600, 600]	0

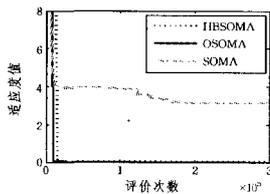


图 1 函数 F2 算法的平均收敛曲线

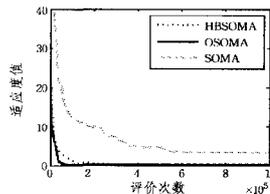


图 2 函数 F5 算法的平均收敛曲线

从表 2 的实验结果比较可以看出:

(1)与 SOMA 的优化结果相比较,用 OSOMA 算法搜索到的表 1 中的 6 个测试函数的最优解和平均解的质量明显优于 SOMA;与 HBSOMA 算法的优化结果比较,利用 OSOMA 算法得到的 F1、F2、F3、F5、F6 的平均解和最优解质量也比 HBSOMA 有了一定的提高,虽然 F4 上 OSOMA 不如 HBSOMA,但明显可以看出 OSOMA 比 HBSOMA 更有可能寻优,从而得到期望的全局最小值。

(2)从独立运行 20 次得到优化质量的稳定性(方差)方面比较,除了 F4、F5 外,OSOMA 算法得到解的适应度值的方差也都小于 SOMA 和 HBSOMA 算法优化解的方差值,这说明在求解质量上 OSOMA 比 SOMA 和 HBSOMA 算法更具有稳定性。

从图 1 和图 2 可以看出,OSOMA 不但具有很强的全局搜索能力,而且具有快速的收敛能力。其收敛速度和精度均优于 SOMA,也比 HBSOMA 有了一定的提高。

综合分析,OSOMA 在求解精度和稳定性上较 SOMA 有了明显提高,也要优于 HBSOMA。其原因在于它将反向学习和自适应步长引入了迁移过程中,增强了种群的寻优能力,减小了其陷入局部极值的可能性,使算法具有较好的寻优能力和较快的搜索速度。

表 2 实验结果

Function	SOMA			HBSOMA			OSOMA		
	Mean	Std Dev	Best	Mean	Std Dev	Best	Mean	Std Dev	Best
F1	5.72e-5	5.15e-5	7.73e-6	3.56e-7	8.36e-7	2.44e-8	1.72e-7	5.36e-7	0
F2	3.9563	1.8299	1.4821	1.75e-15	7.77e-15	0	0	0	0
F3	8.83e-4	3.87e-4	4.68e-4	1.25e-3	3.99e-4	7.15e-4	3.22e-4	9.54e-5	1.82e-4
F4	3.5099	2.45855	0.02265	0.9030	0.9045	0.0081	3.5561	5.6731	0
F5	0.1635	0.3325	0.0081	1.37e-3	9.70e-4	7.15e-4	9.97e-4	2.53e-3	1.10e-12
F6	0.0493	0.0337	0.0097	0.0062	0.0093	1.68e-4	1.83e-3	4.71e-3	0

结束语 本文提出了一种基于反向学习的自组织迁移算法 OSOMA。算法利用反向学习来降低算法陷入局部最优的可能性并加快了算法的搜索速度;反向学习引入了个体的负向性,增大了算法找到最优解的概率;反向学习提高了个体的跳跃性,加快了算法获得最优解的速度。此外,算法还对步长进行自适应变化,即在前期以较大的步长提高算法的速度,后期则以较小的步长进行局部搜索,以提高算法的精度。仿真结果表明,本文算法在函数优化问题上显示出了优异的性能,取得了较好的结果。下一步将研究算法的参数设置并将算法应用于其他领域的优化。

参考文献

[1] Ivan Z, Jouni L. SOMA-Self-organizing Migrating Algorithm[C]// 6th International Conference on Soft Computing. Brno, Czech Republic, 2000

[2] 林志毅, 李元香, 王玲玲. 基于混合迁移的自组织迁移算法[J]. 计算机科学, 2008, 35(12): 175-177

[3] dos Santos Coelho L. Self-organizing Migrating Strategies Applied to Reliability-Redundancy Optimization of Systems[J]. IEEE Transactions on Reliability, 2009, 58(3): 501-510

[4] 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[J]. 计算机工程与应用, 2006, 35: 41-44

[5] Tizhoosh H R. Opposition-based Learning: A New Scheme for Machine Intelligence [C] // Proceedings of International Conference on Computational Intelligence for Modelling Control and Automation. Vienna, Austria, 2005: 695-701

[6] Tizhoosh H R. Opposition-based reinforcement learning [J]. Journal of Advanced Computational Intelligence and Intelligent Informatics, 2006, 10(3): 578-585

[7] 李勇, 刘建昌, 王昱. 基于对位学习的多目标遗传算法[J]. 系统仿真学报, 2009, 21(24): 7801-7804