

随机蛙跳算法的研究进展

韩毅¹ 蔡建湖¹ 周根贵¹ 李延来² 林华珍¹ 唐加福²

(浙江工业大学经贸管理学院 杭州 310023)¹

(东北大学流程工业综合自动化教育部重点实验室 沈阳 110004)²

摘要 随机蛙跳算法(Shuffled Frog Leaping Lgorithm, SFLA)是进化计算领域中一种新兴、有效的亚启发式群体计算技术,近几年来逐渐受到学术界和工程优化领域的关注。SFLA结合了具有较强局部搜索(Local Search, LS)能力的元算法(Memetic Algorithm, MA)和具有良好全局搜索(Global Search, GS)性能的粒子群算法(Particle Swarm Optimization, PSO)的特点,因此其寻优能力强,易于编程实现。详细阐述了SFLA的基本原理和流程,总结了SFLA目前在优化和工程技术等领域中的研究,展望了SFLA的发展前景。

关键词 随机蛙跳算法, 亚启发式算法, 工程优化, 元算法, 粒子群算法

中图分类号 TP29 **文献标识码** A

Advances in Shuffled Frog Leaping Algorithm

HAN Yi¹ CAI Jian-hu¹ ZHOU Gen-gui¹ LI Yan-lai² LIN Hua-zhen¹ TANG Jia-fu²

(College of Business Administration, Zhejiang University of Technology, Hangzhou 310023, China)¹

(Key Laboratory of Integrated Automation of Process Industry of Ministry of Education, Northeastern University, Shenyang 110004, China)²

Abstract Shuffled Frog Leaping Algorithm (SFLA) is a population-based novel and effective meta-heuristics computing method, which received increasing focuses from academic and engineering optimization fields in recent years. Since SFLA is a combination of Memetic Algorithm (MA) with strong Local Search (LS) ability and Particle Swarm Optimization (PSO) with good Global Search (GS) capability, it is of strong optimum-searching power and easy to be implemented. In this paper, the fundamental principles and framework of SFLA were described. Then, the related researches of SFLA in the current optimization and engineering fields were summed up. Lastly, the future perspectives of SFLA were presented.

Keywords Shuffled frog leaping algorithm, Meta-heuristics algorithm, Engineering optimization, Memetic algorithm, Particle swarm optimization

1 引言

近年来,在进化计算领域兴起了一种新型的优化计算方法,称为随机蛙跳算法或混合蛙跳算法(SFLA)。SFLA由Eusuff和Lansley为解决组合优化问题于2003年最先提出^[1]。由于在文献[1]中,Eusuff和Lansley提到了2000年的一篇未公开发表的文章,因此文献[2,3]认为SFLA是在2000年提出的。2006年,文献[4]的公开发表证明了SFLA的正式提出年份是在2003年。自SFLA提出以来,SFLA受到学者们的广泛关注与应用,并迅速成为进化计算领域的研究热点^[5-15]。SFLA是一种基于种群的亚启发式协同搜索算法,算法的执行过程是模拟自然界的元进化行为^[4]。作为一种全新的优化计算方法,SFLA以随机联合体进化算法(Shuffled Complex Evolution, SCE)^[14]作为广度搜索的执行框架,结合了基于遗传基因的元进化算法(Memetic Algo-

rithm, MA)及基于群体觅食行为的粒子群算法(Particle Swarm Optimization, PSO)的特点,具有易于理解、易于编程实现及寻优能力强等优势。

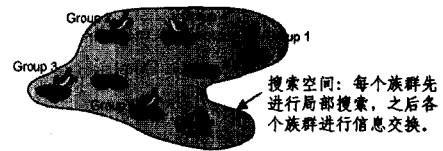


图1 蛙群的觅食行为

SFLA的执行过程模拟了一群青蛙(解)在一片湿地(解空间)中跳动觅食的行为。图1^[8]给出了蛙群觅食行为的示意图。每一只蛙在觅食行为中被看作元或思想的载体,每只蛙可以与其他蛙交流思想并且可以通过传递信息的方式来改进其他蛙的元信息^[11]。这里的元信息是指文化信息或者智力信息,这些信息可以通过诸如模仿等行为在人脑中进行传递。元信息的例子包括歌曲、思想、信息、格言以及服装时

到稿日期:2009-08-25 返修日期:2009-11-07 本文受国家自然科学基金(70625001,70721001,70671095,70971017),浙江省科技计划软科学研究项目(2009C35007)资助。

韩毅(1979—),男,讲师,主要研究方向为生产批量计划与调度、智能优化算法等, E-mail: hanyi@zjut.edu.cn; 周根贵(1958—),男,教授,博士生导师,主要研究方向为物流工程及信息管理; 唐加福(1965—),男,教授,博士生导师。

尚等。元信息的传播方式有多种,譬如一个学者在听到或看到一些有价值的信息时,可以把信息传播给他的同事或学生,可以在他的文章中或其讲授的课程中提及这些信息^[6]。

在 SFLA 中,元信息的改变是通过改变个体的位置来实现的。在算法执行初期,一群蛙被分为多个族群,不同的族群被认为是具有不同思想的青蛙的集合。族群中的蛙按照一定的元进化策略,采用类似粒子群算法的进化方法,在解空间中进行局部深度搜索及内部思想交流。在达到预先定义的局部搜索迭代步数之后,采用随机联合体进化算法的混合过程,将局部思想在各个族群间进行思想交流。这一过程不断重复演进,直到预先定义的收敛性条件得到满足为止。全局性的信息交换和内部思想交流机制的结合,使得算法具有避免过早陷入局部极值点的能力,从而指引算法搜索过程向着全局最优点的方向进行搜索^[2]。

SFLA 具有柔性的框架,其中的各种机制都可用多种算法予以实现。目前,SFLA 已经在许多领域得到了应用,譬如水资源网络优化问题^[1]、连续优化问题^[8]、成品油管网优化问题^[3]、离散优化问题^[2,4,8]、聚类问题^[5,15]、双准则带缓冲区的流水车间调度问题^[6]、桥面修复优化问题^[7]、多目标混合模型装配线排序问题^[9]、TSP 问题^[10]以及多变量 PID 控制器参数调节问题^[12]等。

SFLA 是一种新兴的计算方法。本文以 SFLA 的基本原理为线索,围绕其构架中的重要机制和参数、算法执行过程和应用领域等方面,对 SFLA 的研究现状和研究成果进行全面的综述,使学者们对 SFLA 算法有更为广泛和深入的了解。

2 SFLA 的组成要素与模型

SFLA 是一种结合了确定性方法和随机性方法的进化计算方法。确定性策略使得算法能够有效利用响应信息来指导搜索,随机元素保证了算法搜索模式的灵活性和鲁棒性^[4]。算法在执行过程中,其组成要素与搜索模型是十分重要的两个部分。

2.1 算法组成要素

2.1.1 蛙群(Population)

与其他传统优化算法一样,SFLA 的搜索过程基于一个随机产生的由分布于湿地各处的蛙构成的排序种群,每个种群中的个体蛙都具有一个用来评价个体好坏的适应值。在 SFLA 中,个体蛙被看作元信息的载体,每个元信息包含多个信息元素,这与遗传算法中基因和染色体的概念相类似^[4]。

2.1.2 族群(Memplex)

蛙群产生之后,整个蛙群按照某些具体原则被分割成多个相互独立的排序族群,每个族群具有不同的思想。根据具体的执行策略,族群中的蛙进行解空间内的局部搜索,并且使得元信息在局部个体之间进行传递和广播,这就是元进化过程。图 2^[10]给出了如何将一个由 F 个蛙组成的蛙群进行族群划分的一个例子。

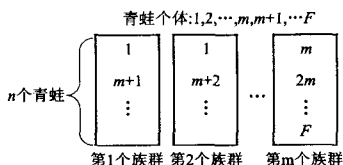


图 2 族群划分

2.1.3 子族群(Sub-memplex)

由于 SFLA 的执行框架十分灵活,在文献[1,4,10]中都提到了子族群的概念。提出子族群是为了预防算法陷于局部最优值而设计的,它是由族群中按照适应值进行选择后产生的青蛙构成的。族群中的蛙所具有的适应值越好,其被选中进入子族群的概率就越大^[1]。子族群代替族群进行解空间内的局部搜索,每进行完一次子族群内的局部搜索,族群内的蛙就需要按适应值大小进行重新排序,并重新生成子族群。

2.1.4 算法参数

与其他优化算法一样,SFLA 亦具有一些必要的计算参数,包括 F :蛙群的数量; m :族群的数量; n :族群中青蛙的数量; $S_{\max}$:最大允许跳动步长; P_x :全局最好解; P_B :局部最好解; P_W :局部最差解; q :子族群中蛙的数量; LS :局部元进化次数以及 SF :全局思想交流次数等。

2.1.5 算法停止条件

SFLA 通常采用两种策略来控制算法的执行时间:

1)在最近的 K 次全局思想交流过程之后,全局最好解没有得到明显的改进;

2)算法预先定义的函数评价次数已经达到。

无论哪个停止条件得到满足,算法都要被强制退出整个循环搜索过程。

2.2 算法模型

算法的模型也就是算法执行时所采用的运算公式,这些公式将伴随着算法的执行思想而给出。在算法执行时,首先产生 F 只青蛙组成蛙群,每只蛙表示解空间的一个解 $X = (x_1, x_2, \dots, x_S)$,其中 S 表示变量的数量。然后将蛙群内的青蛙个体按照适应值进行降序排列,找到全局最好解 P_x 。将蛙群分成 m 个族群,每个族群包含 n 只青蛙,满足关系 $F = m \times n$ 。其中第 1 只蛙进入第 1 个族群,第 2 只蛙进入第 2 个族群,第 m 只蛙进入第 m 个族群,第 $m+1$ 只蛙进入第 1 个族群, ..., 直至所有蛙进入指定位置。根据概率式(1)^[1],随机从族群中选择 q 个个体进入子族群并将子族群进行排序。式(1)中的 j 表示第 j 只蛙, p_j 表示第 j 只蛙被选中的概率。然后找出子族群中的最好解 P_B 和最差解 P_W ,按照式(2)和式(3)对每个子族群进行局部深度搜索。从式(2)可以看出,其公式部分与 PSO 算法模型中的社会认知能力部分十分相似。

$$p_j = 2(n+1-j)/[n(n+1)], j=1, \dots, n \quad (1)$$

$$S = \text{Rand}() \times (P_B - P_W) \quad (2)$$

$$\text{New}P_W = P_W + S, -S_{\max} \leq S \leq S_{\max} \quad (3)$$

式中, $\text{Rand}()$ 表示在 0 到 1 之间进行取值的随机数, S 表示青蛙的跳动步长, $\text{New}P_W$ 表示更新后的 P_W 。如果 $\text{New}P_W$ 是处于可行解空间中,计算 $\text{New}P_W$ 对应的适值。如果 $\text{New}P_W$ 所对应的适值次于 P_W 所对应的适值,则采用 P_x 代替式(2)中的 P_B 重新更新 P_W ;如果仍然没有改进,则随机产生一个新的蛙来代替 P_W 。否则就重复更新过程,直到达到预先设定的局部迭代次数 LS 。完成所有族群的深度搜索后,将所有蛙群中的青蛙进行重新混合和排序,之后再将蛙群分成族群和子族群,继续进行局部搜索,直到任何一个预先设定的算法停止条件得到满足为止^[2]。

3 SFLA 的算法执行步骤^[1]

SFLA 是全局搜索和局部搜索相结合的算法,文献[15]

给出了一些关于算法原理和执行步骤的简单介绍。算法的具体执行步骤按如下方式进行。

3.1 全局搜索过程^[1]

步骤1 初始化 2.1.4 节中所提到的算法参数。

步骤2 随机产生初始蛙群,计算各个蛙的适应值。

步骤3 按适应值大小进行降序排序并记录最好解 P_x , 并且将蛙群分成族群。把 F 个蛙分配到 m 个族群 $Y^1, Y^2, \dots, Y^m$ 中去,每个族群包含 n 个蛙,从而使得 $Y^k = [X(j)^k, f(j)^k | X(j)^k = X(k+m*(j-1)), f(j)^k = f(k+m*(j-1)), j=1, \dots, n, k=1, \dots, m]$ 。这里 $X(j)$ 表示蛙群中的第 j 个蛙, $f(j)$ 表示第 j 个蛙的目标函数值。

步骤4 根据 SFLA 算法公式,在每个族群中进行元进化。

步骤5 将各个族群进行混合。在每个族群都进行过一轮元进化之后,将各个族群中的蛙重新进行排序和族群划分并记录全局最好解 P_x 。

步骤6 检验计算停止条件。如果满足了算法收敛条件,则停止算法执行过程,否则转到步骤3。通常而言,如果算法在连续几个全局思想交流以后,最好解没有得到明显改进则停止算法。某些情况下,最大函数评价次数也可以作为算法的停止准则。

3.2 局部搜索过程^[1]

局部搜索过程是对上述步骤4的进一步展开,具体过程如下:

步骤4-1 设 $im=0$,这里 im 是族群的计数器,用来与族群总数 m 进行比较。设 $iN=0$,这里 iN 是局部进化的计数器,用来与 LS 进行比较。

步骤4-2 根据式(1)在第 im 个族群中选择 q 个蛙进入子族群,确定 P_B 和 P_W 并设 $im=im+1$ 。

步骤4-3 设 $iN=iN+1$ 。

步骤4-4 根据式(2)和式(3)改进子族群中的最差蛙的位置。

(2)中的 P_B ,重新更新最差蛙的位置。

步骤4-6 如果步骤4-5没有改进最差蛙的位置,则随机产生一个处于湿地中任何位置的蛙来替代最差的蛙。

步骤4-7 如果 $iN < LS$,则转到步骤4-3。

步骤4-8 如果 $im < m$,则转到步骤4-2,否则转到全局搜索过程的步骤5。

3.3 算法流程图

SFLA 的算法流程如图3所示。其执行框架十分灵活,具有良好的柔性。学者们在应用该算法时,不需要严格按照上述过程来设计算法。有时为了节省计算时间,可以对 SF-LA 进行一些简化处理,不对族群进行子划分,而直接在族群中采用局部搜索策略即可。

4 SFLA 的相关研究

鉴于 SFLA 的优越性,SFLA 已经广泛应用于许多自然科学与工程科学领域,显示出强大的优势和潜力。关于 SF-LA 的相关研究已经涉及资源网络优化问题、连续优化问题、离散优化问题、聚类问题、流水车间调度问题、装配线排序问题、TSP 问题以及参数调节问题等。

4.1 资源网络优化问题

在资源网络优化方面,2003年 Eusuff 和 Lansey^[1]将 SF-LA 用于求解水资源网络的管径选择和管网扩张问题中。在该文中,解向量由表示管径类型的一串整数值构成,算法作为优化工具,采用成熟软件 EPANET 作为约束检验工具。通过与其他文献的求解结果相比较,作者展示了 SFLA 的优越性,给出了 SFLA 算法在求解这类问题时的适用参数组合。2008年,吴华丽等人^[3]采用 SFLA 求解了成品油管网优化问题,其解向量编码方式与文献[1]的编码方式一致,算法的参数根据文献[1]的参数取值进行选取,计算结果显示了 SFLA 的适用性。

4.2 连续优化问题

2005年,Elbeltagi 等人^[8]对 SFLA 算法、元算法(MA)、粒子群算法(PSO)、蚁群算法(Ant Colony Optimization, ACO)及遗传算法(GA)进行了比较,详细描述了各种算法,给出了各个算法的简单实现过程的伪码及合理化的参数组合。采用 F8 和 F10 这两个具有连续变量的函数,评价了 SFLA 算法的计算性能,结果显示 SFLA 在 F10 函数的求解方面性能不够理想,在 F8 函数的求解上性能优于 GA 算法,与 PSO 算法的性能相近。2008年,Zhang 等人^[11]对 SFLA 算法进行了改进,将 PSO 算法中的个体认知能力部分加入式(2),从而达到改善 SFLA 算法认知行为的目的。改进后的公式如式(4)所示,其中 X_B 是 P_W 所经过的最好位置, $\text{Rand}()$ 是 0 到 1 之间的随机数。

$$S = \text{Rand}() \times (X_B - P_W) + \text{Rand}() \times (P_B - P_W) \quad (4)$$

通过对 4 个具有连续变量的标准函数的测试表明,改进后的 SFLA 算法在达优率、计算时间和达到最优值所需要的函数评价次数方面都得到了显著改善。

4.3 离散优化问题

2005年,Elbeltagi 等人^[8]对 SFLA 算法、元算法(MA)、粒子群算法(PSO)、蚁群算法(Ant Colony Optimization, ACO)及遗传算法(GA)采用建筑业的项目管理问题对 5 种算法的求解结果进行了比较,结果显示 SFLA 求解效果不够理

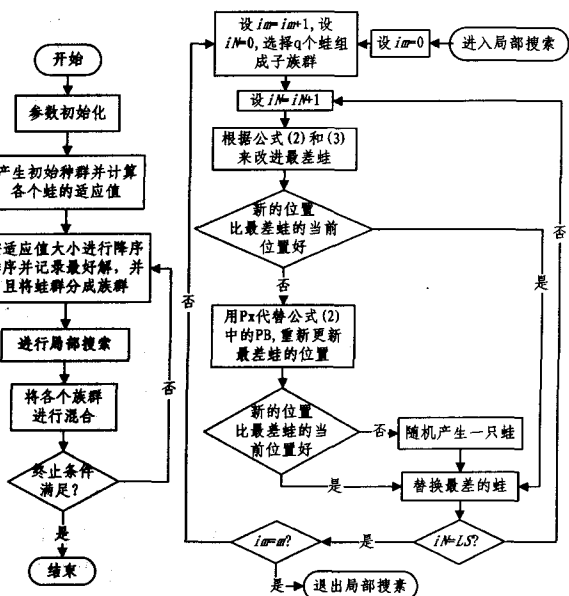


图3 SFLA 算法流程图

步骤4-5 如果步骤4-4改进了最差蛙的位置(解),就用新产生的位置取代最差蛙的位置。否则就采用 P_x 代替式

想,仅优于 GA 算法。2006 年,Eusuff 等人^[4]采用 5 个标准离散函数测试了 SFLA 算法的求解效果,系统地测试并给出了算法的最佳参数组合。之后,他们将 SFLA 算法用于求解地表水模型标定问题,并通过与 GA 相比较来说明 SFLA 算法的有效性。2007 年,李英海等人^[2]对 SFLA 算法进行了改进,通过概率的方式来决定是否改进最差蛙的位置。通过 F7 等 4 个函数的计算结果,表明改进后的算法在求解效果方面得到了改善和提高。

4.4 聚类问题

2008 年,Yang 等^[15]采用 SFLA 与 GA 的混合算法求解了基于聚类的基因选择问题,取得了较好的结果。2009 年,Amili 等人^[5]采用 SFLA 求解了 K 均值聚类问题。通过多组数据集对算法的求解效果进行了测试,并与其他 4 种算法进行了比较,结果显示 SFLA 算法的目标函数值(距离矩阵中所有数值和的最小值)明显优于其他算法的目标值。此外,他们还系统地测试了算法参数组合,给出了算法的最佳参数组合。

4.5 其他优化问题

2006 年,Elbehairy 等人^[6]求解了桥面修复决策问题,比较了 SFLA 算法和 GA 算法在该问题上的求解性能。结果表明,在各自的合理化参数组合下,SFLA 和 GA 都能达到令人满意的求解效果,但总体看来,SFLA 的效果略好于 GA 算法。2007 年,Rahimi-Vahed 等人^[9]采用 SFLA 求解了多目标混合模型装配线排序问题,通过将产品类型与数字进行对应,从而将解向量按整数串形式进行编码。通过将算法与其他 3 种不同的 GA 算法进行比较,表明 SFLA 算法得到了更多的 Pareto 解,错误率比其他 3 个 GA 算法更小,并且 Pareto 解集的多样性更好。2008 年,Luo 等人^[10]采用 SFLA 求解了 TSP 问题,通过与 TSPLIB 中的标准测试问题的结果进行比较,证明了 SFLA 算法的优越性,得到了新的 6 组改进解。2008 年,Huynh^[12]采用混合 SFLA 算法求解了多变量 PID 控制器参数调节问题,结果证明混合 SFLA 算法的控制能力更强,控制效果比 GA 等算法更好。2009 年,Rahimi-Vahed 等人^[6]求解了具有双准则的带缓冲区流水车间调度问题。采用 Pareto 解的数量和错误率作为评价标准,将混合 SFLA 与 2 个 GA 算法进行了比较,结果表明 SFLA 算法在多目标问题求解上有优越性。

结束语 本文对随机蛙跳算法的研究及进展进行了全面的论述。在短短的几年内,SFLA 引起了国际学术界的广泛关注,成为国际优化计算领域的研究热点之一,亦逐渐成为解决实际工程优化问题的热门工具。

SFLA 虽然已对多种优化问题取得了很好的求解效果,并在诸多应用研究领域都取得了长足的发展,但远未达到成熟的阶段,还有很多问题值得深入的分析与探讨。首先,算法是否存在统一的参数? 如果不存在,如何合理设置针对不同问题的参数组合? 其次,算法在求解最优值处于边界点上的问题时并非十分有效,如何进行解决? 这些问题都需要学者们在今后的研究中做进一步探讨。此外,与其他优化算法一样,关于 SFLA 理论方面的研究尚显不足,进一步的理论研究应侧重于算法收敛性、收敛速度、鲁棒性及算法模型改进等方面。总体看来,SFLA 算法扩展了智能优化算法的领域,提供

了求解优化问题的新思路,具有巨大的科研价值和应用潜力。

参考文献

- [1] Eusuff M M, Lansey K E. Optimization of water distribution network design using the shuffled frog leaping algorithm [J]. *Journal of Water Resources Planning and Management*, 2003, 129(3): 210-225
- [2] 李英海, 周建中, 杨俊杰, 等. 一种基于阈值选择策略的改进混合蛙跳算法[J]. *计算机工程与应用*, 2007, 43(35): 19-21
- [3] 吴华丽, 汪玉春, 陈坤明, 等. 基于混合蛙跳算法的成品油管网优化设计[J]. *石油工程建设*, 2008, 34(1): 14-16
- [4] Eusuff M, Lansey K, Pasha F. Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization [J]. 2006, 38(2): 129-154
- [5] Amiri B, Fathian M, Maroosi A. Application of Shuffled frog-leaping algorithm on clustering[J]. *International Journal of Advanced Manufacturing Technology*, 2009
- [6] Rahimi-Vahed A, Dangchi M, Rafiei H. A novel hybrid multi-objective shuffled frog-leaping algorithm for a bi-criteria permutation flow shop scheduling problem[J]. *International Journal of Advanced Manufacturing Technology*, 2009, 41(11/12): 1227-1239
- [7] Elbehairy H, Elbeltagi E, Hegazy T, et al. Comparison of two evolutionary algorithms for optimization of bridge deck repairs [J]. *Computer-Aided Civil and Infrastructure Engineering*, 2006, 21(8): 561-572
- [8] Elbeltagi E, Hegazy T, Grierson D. Comparison among five evolutionary-based optimization algorithms[J]. *Advanced Engineering Informatics*, 2005, 19(1): 43-53
- [9] Rahimi-Vahed A, Mirzaei A H. A hybrid multi-objective shuffled frog-leaping algorithm for a mixed-model assembly line sequencing problem[J]. *Computers and Industrial Engineering*, 2007, 53(4): 642-666
- [10] Luo Xue-hui, Yang Ye, Li Xia. Solving TSP with shuffled frog-leaping algorithm[C]// *The 8th International Conference on Intelligent Systems Design and Application*. Kaohsiung, 2008: 228-231
- [11] Zhang Xun-cai, Hu Xue-mei, Cui Guang-zhao, et al. An improved shuffled frog leaping algorithm with cognitive behavior[C]// *Proceedings of the 7th World Congress on Intelligent Control and Automation*. Chongqing, 2008: 6197-6202
- [12] Huynh Thai-hoang. A modified shuffled frog leaping algorithm for optimal tuning of multivariable PID controllers[C]// *IEEE International Conference on Industrial Technology*. Chengdu, 2008: 1-6
- [13] Liong Shie-yui, Atiquzzaman M. Optimal design of water distribution network using shuffled complex evolution[J]. *Journal of The Institution of Engineers (Singapore)*, 2004, 44(1): 93-107
- [14] 王辉, 钱锋. 群体智能优化算法[J]. *化工自动化及仪表*, 2007, 34(5): 7-13
- [15] Yang Cheng-san, Chuang Li-yeh, Ke Chao-Hsuan, et al. A Combination of Shuffled Frog-leaping Algorithm and Genetic Algorithm for Gene Selection[J]. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 2008, 12(3): 218-226