

新的进化算法——文化算法

杜 琼 周一届

(江南大学 无锡 214000)

摘 要 文化算法源于对人类社会多层面进化的模拟,为进化算法提供了一个新的计算框架。和其他进化算法相比,文化算法概念清晰,更能准确地反映社会的进化过程,并在一些领域取得了成功的应用。本文首先讲述了该算法的生物解释,然后介绍了算法的基本原理和不同版本,并给出了一些成功应用的实例,最后给出了本文的结论和进一步的研究方向。

关键词 进化算法,文化算法,知识表达

A New Evolution Algorithm——Cultural Algorithm

DU Qiong ZHOU Yi-Jie

(Southern of Yangtze University, Wuxi 214000)

Abstract A Cultural Algorithm (CA), rooting from simulation of evolution of human being society, provides a new computable framework of Evolution Algorithms. Concept of CA is more easily understood and more accurately to reflect evolution process of society than some other optimization algorithms. CA has been successfully applied to much engineering. Firstly, this paper depicts natural explanation about CA. Secondly, introduces its basic theory and several development versions of CA, and presents some applications of CA. At last, a brief conclusion and further research direction is given.

Keywords Evolution algorithm, Cultural algorithm, Knowledge expression

1 引言

神秘的大自然总是能够给人类带来惊喜,启迪了无数的科学发明,帮助人们解决了大量的实际问题,例如遗传算法(Genetic Algorithm)、免疫算法(Immune Algorithm)等进化计算方法,不依赖于具体问题,对问题的知识所需较少,甚至不需要,在解决工程实践问题上取得了很好的成绩。因此,进化计算引起了许多研究人员的兴趣,目前进化计算的许多研究还只是集中在生物(或者说是基因)自然选择(竞争)这一层面上。然而许多情况表明,文化能使种群以一定的速度进化和适应环境,而这个速度是超越单纯依靠基因遗传生物进化速度的^[1]。种群在进化过程中,个体知识的积累以及群体内部知识的交流在另外一个层面促进群体的进化,这种知识在此被称为文化。文化被 Durham 定义^[2]为“一个通过符号编码表示众多概念的系统,而这些概念是在群体内部及不同群体之间被广泛和历史般长久传播的”。学者 Renfrew^[3]指出,随着时间的迁移人类在进化过程中逐渐掌握了提取、编码和传播信息知识的能力,这种能力是区别其他物种人类所特有的能力。正是这种能力使人类能够形成、积累和传递经验,可以加快和指导发展的速度与方向。所以说人类社会的进化不仅仅是自然选择的结果而是超越了自然选择的水平,人类文化(文明)的发展对其产生巨大的影响,甚至可以说人类发展到当今这个阶段,文化对其未来的发展将起到主要作用。

文化算法正是受到这一启示而形成的,文化算法是一个多进化过程的算法,分别从微观和宏观不同层面模拟生物层面的进化和文化层面的进化,各进化过程又相互影响相互促进。更进一步地讲,文化算法还是一个分层的算法,可根据问题的复杂度将原算法框架中的微观、宏观两个层面再细分成多个层面,各个层面相互影响以提高算法的计算效率。文化算法超越传统的进化算法,通过模拟微观、宏观各层面的进化,更加准确地反映了物种的进化过程,并在一些问题上取得

了比传统进化算法更好的结果。目前国内刚刚开始关注文化算法的研究。这篇文章的目的是抛砖引玉,使读者能够了解文化算法的特点和目前的研究情况,并展开进一步的研究。

2 文化算法的原理

文化算法框架是由群体空间(Population Space)和信念空间(Belief Space)两部分组成的,如图1。其中,空间中的个体必须具备以下特点:1)有一定的行为规则;2)个体的经验可以被描述、归纳、编码(形式化)和传递;3)个体经验经处理后形成群体经验,群体经验可以限制和修改个体的行为规则。群体空间是从微观角度模拟个体根据一定行为准则进化的过程,而信念空间是从宏观角度模拟文化的形成、传递和比较等进化过程。

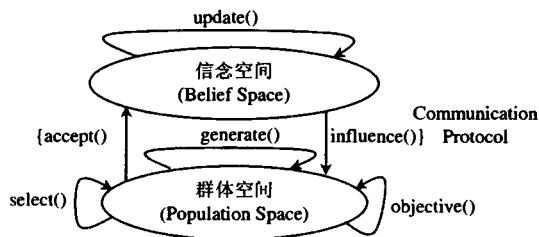


图1 文化算法框架

群体空间与信念空间是两个相对独立的进化过程,两个空间是根据一定的通讯协议相互联系的。群体空间个体在进化过程中,形成个体经验,通过函数 accept()将个体经验传递到信念空间,如图1所示。信念空间将收到的个体经验看作一个单独的个体,根据一定行为规则进行比较优化,形成群体经验。并根据信念空间现有的经验和新个体经验的情况用 update()函数更新群体经验。信念空间在形成群体经验后,通过 influence()函数对群体空间中个体的行为规则进行修改,以使个体空间得到更高的进化效率。群体空间中的 ob-

jective()函数是目标函数,其作用是用来评价空间中个体适应值的。generate()函数根据个体行为规则和父辈个体参数生成下一代个体,select()函数根据规则从新生成个体中选择一部分个体作为下代个体的父辈。文化算法的基本伪代码如下^[4]:

```

Begin
  t=0;
  Initialize Population POP(t);
  Initialize Belief Space BLF(t);
Repeat
  Evaluate Population POP(t);
  Update(BLF(t), Accept(POP(t)));
  Variation(POP(t), Influence(BLF(t)));
  t=t+1;
  Select(POP(t) from POP(t-1));
until termination condition achieved
End

```

文化算法框架提供了一种多进化过程的计算模型,因此从计算模型的角度来看,任何一种符合文化算法要求的进化算法都可以嵌入文化算法框架中作为群体空间的一个进化过程。所以根据不同的进化算法,就会有不同的文化算法,目前有多多个不同的文化算法版本。

3 文化算法的发展与应用

文化算法在解决不同问题时,个体及群体经验的获取和优化也是不尽相同的。所以随着应用文化算法解决问题种类的增多,丰富和发展了文化算法。这些算法都是对基本文化

算法在某些方面的改进和延伸,其核心是一致的。

3.1 解决约束优化问题的文化算法^[5]

文化算法在解决约束优化问题时,关键是如何表达和处理约束条件,如何从群体空间获取个体经验和信念空间如何得到群体经验并改善群体空间的进化过程,即信念空间如何设计。文化算法在求解非约束问题^[6]时,使用了两种知识:1)每次迭代中发现最佳点的个体的确切位置;2)保存和修正目标函数主变量的变化区间在可能发现最好解的范围内。而约束优化问题与非约束优化问题最大的区别是:边界条件的知识如何表示、传递和比较。

3.1.1 约束边界在信念空间中的表示 首先,将空间的各坐标轴用区间进行划分,二维空间问题如图2(a),3维空间为一个立方体,多维空间依此类推,将目标函数置于此空间中,使信念空间等同于问题的求解空间,图2(a)中的曲线表示一个约束优化函数的约束边界,曲线内部为不可行区域,曲线外为可行区域。其次将空间按不同特征进行分类,以便于知识的获取和传递。在此,空间分为4类:1)可行域;2)不可行域;3)半可行域;4)未知域。如图2(b),白色和黑色方格分别代表求解问题的可行域和不可行域,灰色方格中一部分为可行域另一部分为非可行域,在此称为半可行域,其余的为未知域。

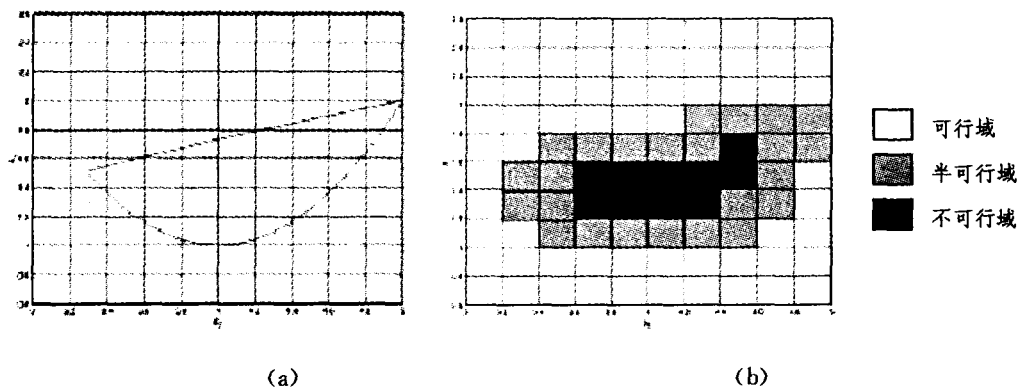


图2 求解问题在信念空间的表示^[5]

信念空间如此设计可以使群体空间中位于非可行域的个体向可行域移动。此外,很多较难求解的约束优化问题的最佳点往往位于可行域与非可行域的边界线上,因此半可行域对发现较好的解起到引导的作用。

3.1.2 信念空间的初始化与更新^[8] 信念空间在此定义为 $\langle N[j], C[i] \rangle$,这两个参数分别表示两种知识,1) $N[j]$,保存目标函数主变量的变化区间;2) $C[i]$,保存每一代中发现最佳点的个体的确切位置。

$N[j]$ 中包含 $\langle l_j, L_j, U_j \rangle$ 三个参数, $j=1, 2, \dots, n$,其中 l_j 表示参数 j 的闭区间, $l_j = [l_j, u_j] = \{x | l_j \leq x \leq u_j, x \in R\}$ 。 l_j 在初始时设置为变量定义域边界的值,可在以后的计算中修改。 L_j 表示参数 j 区间下限 l_j 对应的目标函数的适应值, U_j 是参数 j 区间上限 u_j 对应的目标函数的适应值。

$C[i]$ 中包含 $\langle Class_i, Cnt1_i, Cnt2_i, W_i, Pos_i, Csize_i \rangle, i=1, 2, \dots, m$ 五个参数。 $Class_i$ 表示信念空间中第 i 个单元(在二维空间就是第 i 个方格)的状态,诸如:可行域、不可行域等。 $Cnt1_i, Cnt2_i$ 分别统计位于可行域和非可行域个体的数量,初始化为0。 W_i 表示第 i 个单元的权值,该单元的适应值越高权值就越大(在此以最大化问题为例,反之亦然)。 Pos_i 是一个矢量,表示第 i 个单元最左边角的位置。 $Csize_i$ 表示第 i 个

单元的大小。

在此,信念空间像一个里面藏有地图的滑动窗口,如果说 $N[j]$ 表示了这个 n 维空间窗口的大小,则 $C[i]$ 就是这个窗口里的地图。在 $N[j]$ 限制好的空间里,由 $C[i]$ 引导算法在可行域和半可行域进行搜索,缩小了搜索空间,提高了搜索效率。

1) $N[j]$ 的更新规则如下:

假设是第 i 个体影响参数 j 的区间的下边界,则 j 的下边界和对应的适应值为:

$$l_j^{t+1} = \begin{cases} x_{i,j} & \text{if } x_{i,j} \leq l_j^t \text{ or } objective(x_i) < L_j^t \\ l_j^t & \text{otherwise} \end{cases} \quad (1)$$

$$L_j^{t+1} = \begin{cases} objective(x_i) & \text{if } x_{i,j} \leq l_j^t \text{ or } objective(x_i) < L_j^t \\ L_j^t & \text{otherwise} \end{cases} \quad (2)$$

在此, l_j^t 表示在第 t 次迭代参数 j 的下边界, L_j^t 表示 l_j^t 对应目标函数的适应值。同理假设第 k 个体影响参数 j 的区间的上边界,则 j 的下边界和对应的适应值为:

$$u_j^{t+1} = \begin{cases} x_{k,j} & \text{if } x_{k,j} \geq u_j^t \text{ or } objective(x_k) < U_j^t \\ u_j^t & \text{otherwise} \end{cases} \quad (3)$$

$$U_j^{t+1} = \begin{cases} objective(x_k) & \text{if } x_{k,j} \geq u_j^t \text{ or } objective(x_k) < U_j^t \\ U_j^t & \text{otherwise} \end{cases} \quad (4)$$

在此, u_j^t 表示在第 t 次迭代参数 j 的区间的上边界, U_j^t 表示 u_j^t 对应目标函数的适应值。

2) $C[i]$ 的更新规则如下:

算法在每次迭代过程中, 个体所在的位置是可以获得的, 因此信念空间中每个单元都可以计算出该单元中位于可行域和非可行域的个体的个数, 并由此根据式(5)得到 $Class_i$ 的值。而 $W_i, Pos_i, Csize_i$ 由算法直接修改。

$$Class_i = \begin{cases} unknown & \text{if } Cnt1_i = 0 \text{ and } Cnt2_i = 0 \\ feasible & \text{if } Cnt1_i > 0 \text{ and } Cnt2_i = 0 \\ unfeasible & \text{if } Cnt1_i = 0 \text{ and } Cnt2_i > 0 \\ semi-feasible & \text{otherwise} \end{cases} \quad (5)$$

考虑到算法的效率, $N[j]$ 与 $C[i]$ 的更新频率是不同的, $C[i]$ 可以每次迭代都更新, 而如果 $N[j]$ 每次迭代也都更新会影响算法的速度, 因此, 一般情况下 $N[j]$ 每迭代 $k(k>1)$ 次再更新一次。

求解约束优化问题的文化算法流程如下:

- 1) $t=0$;
- 2) 初始化群体空间 $P[nc, np]^t$ (空间中有 nc 个个体, 目标函数有 np 个自变量);
- 3) 初始化 $B' = (N'(\cdot), C'(\cdot))$;
- 4) 比较群体空间中各个体的适应值;
- 5) 更新信念空间;
- 6) 比较父辈个体的适应值, 根据信念空间获得的知识通过 $Influence()$ 函数产生后代;
- 7) $t=t+1$;
- 8) 从新的群体空间 $P[nc, np]^{t+1}$ 中选择个体;
- 9) 回到 4) 直到终止条件满足。

3.2 文化算法在数据挖掘问题中的应用^[11]

文化算法的框架也可以用于数据挖掘。在这个框架下, 群体空间由数据库中的所有数据组成, 每一条记录是一个个体; 信念空间则由一套层级结构的区域模式构成。在群体空间里, 进化计算搜索的任务是在某种标准下(例如目标函数)在数据库中发现适应值最高的记录。因此这个过程可以被看作是求解约束优化问题的过程。

在信念空间中, 任务是处理从群体空间获取的知识, 这个过程是一个数据挖掘的过程, 可能得到一些未知的模式。在此, 算法的优化搜索和数据挖掘是通过信念空间与群体空间之间的通讯协议整合在一起的。通过 $accept()$ 函数, 信念空间从群体空间得到经验并更新优化信念空间的知识; 通过 $influence()$ 函数信念空间向群体空间提供全局知识, 推动群体空间进化搜索。因此, 进化过程和数据挖掘过程能相互促进, 提高计算效率。

用于数据挖掘的文化算法是在求解约束优化问题的文化算法上发展出来的, 在信念空间的表示、更新以及信念与群体两个空间的通讯等方面进行了修改, 使之更适合于数据挖掘。因此, 该算法是上一节算法的延伸。

除此之外, 文化算法还在多目标优化^[12]、语义网络^[13]、动态环境优化^[14]等方面有成功的应用。目前, 对文化算法的研究才刚刚起步, 相对其它比较成熟的进化算法, 应用的范围还比较少。但随着研究的进一步深入, 文化算法将会应用到越来越多的领域中。

结论与进一步的工作 文化算法提供了一个新的进化计算框架, 其研究刚刚开始, 远没有像遗传算法和免疫算法那样形成系统的分析方法和一定的数学基础, 有许多问题还需要

进一步研究:

1) 并行计算方面。文化算法是一种进化计算框架, 具有其他进化计算的特点, 具有天然的并行性, 因此文化算法的并行化研究会为其拓展更广泛的应用领域。

2) 空间参数的选择与设计。文化算法中空间的选择和设置依赖于具体问题, 对于不同问题其空间设置就有所不同。研究如何选择和设计通用模式, 使其减少对具体问题的依赖, 也将大大促进文化算法的发展和应用。

3) 算法的基础理论研究。文化算法很大程度上依赖于问题知识的表达, 提取和演化。对于如何将知识有效地在计算机中进行获取和处理, 一直是人工智能领域的一项难题。因此人工智能等相关领域的发展也会带动文化算法基础理论研究的发展。

随着文化算法和一些相关领域学科的发展, 文化算法一定会大显身手并有新的突破。本文希望读者能够对文化算法原理、特点和应用范围有初步的认识, 从深度和广度上进一步研究和发展文化算法。

参考文献

- 1 Robert R G. An Introduction to Cultural Algorithms. In: Proc. of the 3th annual Conf. on Evolution Programming, Sebalk, A. V. Fogel L. J., River Edge, NJ. World Scientific Publishing, 1994. 131~136
- 2 Durham W. Co-Evolution: Genes, Culture and Human Diversity. Stanford University Press, Stanford, CA, 1994
- 3 Renfrew A C. Dynamic Modeling in Archaeology: What, When, and Where?, Dynamical Modeling and the Study of Change in Archaeology. In: S. E. van der Leeuw, ed. Edinburgh University Press, 1994
- 4 Reynolds R G, Zhu Shinin. Knowledge-Based Function Optimization Using Fuzzy Cultural Algorithms with Evolutionary Programming. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 2001, 31(1), 1~18
- 5 Carlos C A, Ricardo L B. Constrained Optimization using an Evolutionary Programming-Based Cultural Algorithm. In: en Ian C. Parmee, ed. Adaptive Computing in Design and Manufacture V, Springer, London, April 2002. 317~328
- 6 Chung C-J. Knowledge-Based Approaches to Self-Adaptation in Cultural Algorithms; [PhD thesis]. Wayne State University, Detroit, Michigan, 1997
- 7 Chung C-J, Reynolds R G. A Testbed for Solving Optimization Problems using Cultural Algorithms. In: Larence J. Fogel, Peter J. Angeline and Thomas Back, eds. Evolutionary Programming V; Proc. of the Fifth Annual Conf. on Evolutionary Programming, Cambridge, Massachusetts, MIT Press, 1996
- 8 Jin X, Reynolds R G. Using Knowledge-Based Evolutionary Computation to Solve Nonlinear Constraint Optimization Problems: a Cultural Algorithm Approach. In 1999 Congress on Evolutionary Computation, Washington, D. C., IEEE Service Center, 1999. 1672~1678
- 9 Michalewicz Z, Schoenauer M. Evolutionary Algorithms for Constrained Parameter Optimization Problems. Evolutionary Computation, 1996, 4(1): 1~32
- 10 Koziel S, Michalewicz Z. Evolutionary Algorithms, Homomorphous Mappings, and Constrained Parameter Optimization. Evolutionary Computation, 1999, 7(1): 19~44
- 11 Jin X, Reynolds R G. Data Mining Using Cultural Algorithms and Regional Schemata. In: Proc. of the 14th IEEE Intl. Conf. on Tools with Artificial Intelligence (ICTAI'02), 2002. 33~40
- 12 Coello C A, Becerra R L. Evolutionary Multiobjective Optimization using a Cultural Algorithm. In: 2003 IEEE Swarm Intelligence Symposium, Indianapolis, Indiana, IEEE Service Center, 2003. 6~13
- 13 Reynolds R G, Rychtyckyj. Using Cultural Algorithms to Improve Performance in Semantic Networks. In: Proc. 1999 IEEE Congress on Evolutionary Computation, Washington, D. C., July, 1999. 1651~1656
- 14 Reynolds R G, Saleem S. Cultural Algorithms in Dynamic Environments. In: Proc. of the Congress on Evolutionary Computation, San Diego, California, 2000, 2: 1513~1520