

文章编号:1001-9081(2009)05-1267-03

## 基于思维进化的蚁群算法

贾洪岩, 郭进利

(上海理工大学 管理学院, 上海 200093)

(jhy@usst.edu.cn)

**摘要:**为了改善基本蚁群算法易陷入局部最优从而导致算法过早停滞的缺陷,提出了一种基于思维进化的蚁群算法,阐述了该算法的原理和具体实施方案。选取旅行商问题作为算法的仿真实例,通过对仿真实例进行计算和结果比较,验证了该算法的有效性和可行性。

**关键词:**思维进化;蚁群算法;趋同;异化;旅行商问题

**中图分类号:** TP18 **文献标志码:** A

### Ant colony algorithm based on mind evolution

JIA Hong-yan, GUO Jin-li

(Business School, University of Shanghai for Science and Technology, Shanghai 200093, China)

**Abstract:** The basic ant colony algorithm is easy to fall in local peak. In order to overcome this shortcoming resulting in the precocity and stagnation, a new kind of ant colony algorithm based on mind evolution was proposed. The detailed realization of the method was illustrated and some examples of the traveling salesman problem were computed. Simulation results prove that the new algorithm is effective and feasible.

**Key words:** mind evolution; Ant Colony Algorithm (ACA); similitaxis; dissimilation; Traveling Salesman Problem (TSP)

## 0 引言

蚁群算法(Ant Colony Algorithm, ACA)<sup>[1-2]</sup>采用分布式并行计算和正反馈机制,具有较强的鲁棒性和发现较好解的能力,易于与其他方法结合,在求解旅行商问题、作业车间调度问题、图着色问题等诸多组合优化问题中得到广泛应用<sup>[3]</sup>,并表现出较好的效果。但与此同时也存在着一些缺陷,如收敛速度慢、易出现早熟停滞现象等。针对 ACA 的缺点,已经提出了诸多的改进算法<sup>[4-5]</sup>。

思维进化算法(Mind Evolutionary Computation, MEC)<sup>[6-7]</sup>引入了遗传算法的群体概念,但从根本上与遗传算法存在着不同,它通过趋同和异化两种操作来模拟人类思维的进化过程。趋同是群体内竞争学习的过程,异化是在群体间进行竞争学习。

本文受思维进化算法和蚁群算法的启发,提出了基于思维进化框架下的蚁群算法(ACA based on Mind Evolution, ACA-ME)。它不仅具有蚁群算法快速寻优的能力,而且具有加强解的多样性的能力,能够有效克服基本蚁群算法易陷入局部最优从而导致算法过早停滞的缺陷。

## 1 思维进化算法

思维进化算法模拟人类思维的进化过程,认为思维的进步是一种定向提高过程。该算法提出了趋同和异化两个概念,通过引入公告板的办法,充分利用了进化过程中的信息,增加了进化方向的选择性。

思维进化算法的基本结构如图 1 所示。

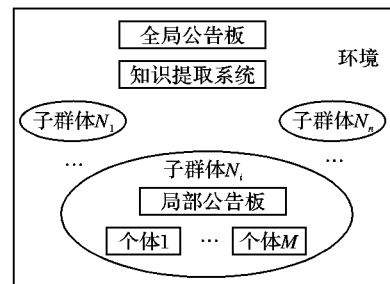


图 1 思维进化算法的基本结构

MEC 的对象为由若干个优胜子群体、临时子群体和全局公告板组成的环境,其中每个优胜子群体或临时子群体由若干个体和局部公告板组成,每一个个体均结合计算内容被赋予一得分,得分是指导环境进化的主要依据。

进化过程由趋同和异化两个算子完成。趋同指的是采用现有的、他人的思维方式或方法解决问题,是子群体内部的个体根据局部公告板的信息竞争学习并逐步收敛到局部最优解的过程。异化指的是摆脱常规的思维方式和方法,提出新的观点、思考方法和解决问题的途径,是在全局公告板的指导下实现全局寻优的过程。其中全局公告板内容是信息提取系统在进化中提取的环境信息。

算法首先设置参数,初始化全局和局部公告板,随机产生若干个体,确定优胜子群体和临时子群体作为初始群体。然后按事先设定好的策略重复进行趋同操作和异化操作,根据设定好的终止条件,就可对问题进行求解。

## 2 基于思维进化的蚁群算法设计

为了克服基本蚁群算法易陷入局部最优的缺陷,本文基

收稿日期:2008-12-05;修回日期:2009-02-13。

基金项目:国家自然科学基金资助项目(70871082);上海市重点学科建设项目(S30504;T0502)。

作者简介:贾洪岩(1979-),男,辽宁阜新,人,硕士研究生,主要研究方向:工业工程、智能算法;郭进利(1960-),男,陕西西安人,教授,博士,主要研究方向:工业工程、运筹学、控制论。

于思维进化思想,提出了基于思维进化的蚁群算法,该算法中子群体内部通过蚁群算法进行趋同操作,信息素记录到局部公告板。充分利用思维进化过程中的趋同和异化操作,使算法跳出局部最优,扩大了解的搜索空间。

本文提出的基于思维进化的蚁群算法其主要组成部分设计如下。

1) 个体和子群体。本算法把问题的解作为个体,子群体有两类:优胜子群体和临时子群体。

2) 公告板。局部公告板用于子群体内的个体张贴信息,为个体间和子群体间提供信息交流的机会,公告板上的信息包含:个体或子群体的序号,信息素的分布,得分。而子群体在全局公告板上张贴信息,全局公告板记录在整个环境中得分排在前面的若干个体信息。

具体方案设计如下。

1) 初始化:设置  $N_s + N_t$  个子群体作为初始群体,将每个群体看成是一个蚁群系统,初始化各参数,确定各群体的初始信息素分布,各群体在解空间随机产生  $M$  个个体,按得分确定  $N_s$  个优胜子群体和  $N_t$  个临时子群体,更新局部公告板。

2) 趋同操作:各群体的内部进化过程采用蚁群算法来进行。根据蚁群算法对各群体进行蚂蚁搜索,计算每个个体的得分,再找出问题需要的最好的个体作为本代的优胜者,更新局部公告板上的个体信息和信息素分布,蚂蚁根据局部公告板上的信息向优胜者学习来提高得分,重复搜索过程。当连续  $n$  次搜索找不到更好的个体时,认为此群体成熟,成熟后的优胜者得分作为此群体的得分。

3) 异化操作:当一个临时子群体成熟后,若得分优于某个优胜子群体,则此优胜子群体被放弃,由临时子群体代替,若得分没有优于任何一个优胜子群体,则此临时子群体被放弃。当一个优胜子群体成熟后,进行局部搜索,若得分提高,则更新局部公告板上的个体信息和信息素分布,否则此优胜子群体被放弃。被收敛后放弃的优胜子群体或临时子群体,保留局部公告板上的信息素分布,并以全局公告板上的信息影响当前信息素分布,指导重新产生新群体。

4) 重复趋同和异化操作,根据设定好的终止条件,就可对问题进行求解。

### 3 算法流程和分析

根据上述分析,本文提出的基于思维进化的蚁群算法的框架描述如下:

Procedure 基于思维进化的蚁群算法

Begin

$d = 0;$  /\* 异化操作次数 \*/

在解空间随机产生  $n_m$  个个体; /\*  $n_m$  为子群体数目 \*/

While(满足最大异化代数  $N_m$ )

$d = d + 1;$

$s = 0;$  /\* 趋同操作次数 \*/

初始化局部公告板;

While(满足蚁群搜索最大迭代次数  $N_c$  或某种条件)

$s = s + 1;$

计算子群体中每个个体的得分;

更新局部公告板;

根据局部公告板信息应用蚁群算法趋同操作;

End while

对成熟优胜子群体进行局部搜索操作;

If (优胜子群体得分提高)

返回继续应用蚁群算法进行趋同操作;

Else

更新全局公告板;

以全局公告板信息指导重新产生新子群体;

End While

End

现有的研究表明,蚁群算法对求解经典旅行商问题 (Traveling Salesman Problem, TSP) 的时间复杂度为  $O(N_c \cdot n^2 \cdot m)$ 。其中,  $N_c$  表示蚁群搜索最大迭代次数,  $n$  表示城市数,  $m$  则表示参与搜索的蚂蚁数量。这里假设子群体数目为  $n_m$ , 子群体内个体数目为  $M (M = m)$ , 最大异化代数为  $N_m$ , 则本文算法的渐进时间复杂度为  $O(N_m \cdot n_m \cdot N_c \cdot n^2 \cdot m)$ 。

对比本文提出的基于思维进化的蚁群算法,可以认为基本蚁群算法只有一个群体空间,而基于思维进化的蚁群算法有多个子群体空间,各子群体根据局部公告板上的信息独立进行蚂蚁搜索,各子群体间通过全局公告板进行异化操作,可实现并行操作,有效提高了算法的效率。

## 4 旅行商问题算例

### 4.1 旅行商问题的描述

旅行商问题的要求是寻找通过所有城市各一次并最终回到出发点的最短路径。TSP 问题是组合优化领域中的经典难题,现在已经被归为 NP-hard 问题之列。其实质为:一个旅行商要去  $n$  座城市推销其商品,假定任意两个城市  $i, j$  之间的距离为  $d_{ij}$  已知,问他应该怎样选择行走路线,使得从某一城市出发,经过每个城市且仅一次,最后回到原始出发城市的总行程最短。

### 4.2 仿真实验

本文选取标准的 30 个城市的 Oliver30 问题和 31 个城市的中国旅行商问题 (Chinese Traveling Salesman Problem, CTSP) 作为仿真实例,其中 CTSP 算例采用 31 个城市的相对坐标为原始数据,如下所示:(拉萨 1304, 2312; 北京 3639, 1315; 上海 4177, 2244; 天津 3712, 1399; 石家庄 3488, 1535; 太原 3326, 1556; 呼和浩特 3238, 1229; 沈阳 4196, 1004; 长春 4312, 790; 哈尔滨 4386, 570; 西安 3007, 1970; 兰州 2562, 1756; 银川 2788, 1491; 西宁 2381, 1676; 乌鲁木齐 1332, 695; 济南 3715, 1678; 南京 3918, 2179; 杭州 4061, 2370; 合肥 3780, 2212; 南昌 3676, 2578; 福州 4029, 2838; 台北 4263, 2931; 郑州 3429, 1908; 武汉 3507, 2367; 长沙 3394, 2643; 广州 3439, 3201; 南宁 2935, 3240; 海口 3140, 3550; 成都 2545, 2357; 贵阳 2778, 2826; 昆明 2370, 2975)。

本文在 Windows XP 下用 Visual C++ 6.0 编制程序,分别用本文算法和基本蚁群算法来求解仿真实例。本文算法参数设置为:优胜子群体个数 2, 临时子群体个数 3, 子群体中个体的数目 30, 子群体 10 代得分不增加认为收敛成熟, 优胜子群体成熟后采用 2-OPT 局部搜索寻优, 全局公告板中保留 10 个优胜者, 环境以进化到 50 代为终止。本文采取 MAX\_MIN 蚂蚁算法作为本文算法内核, 所选取的参数为:  $\alpha = 1, \beta = 4, \rho = 0.8, Q = 100, \tau_{\min} = 0.01, \tau_{\max} = 10$ , 蚂蚁数即为子群体内个体数 30。

本算法程序独立运行 20 次, 将求解的最优解、最差解和平均解与其他算法的计算结果进行了对比, 如表 1、2 所示。

由结果可以看出, 在 Oliver30 算例中, 本算法一直能得到 423.74 的最优值, 结果优于基本蚁群算法和文献[7]的思维进化算法以及文献[8]的蚁群-免疫混合算法。在 CTSP 算例中, 本算法得到了 15378 的最优值, 结果优于文献[9]的 MAX\_MIN 蚂蚁算法, 得到了和文献[10]的两段式遗传算法一样的最优值, 但在平均解上要优于文献[10], 说明本文算法具有较好的稳定性。

实际上,本文算法很快就可以完成计算并得到问题的满意解,在多次实验中,算法以较少的进化代数就找到了问题的最优解,可见算法具有较好的收敛性。从结果对比可以发现,本文算法无论在问题的最优解、最差解还是平均解上,都较基本蚁群算法有了很大改善,在解的质量方面明显优于其他算法。

表1 Oliver30 问题计算结果比较

算法	Oliver30 问题		
	最优解	最差解	平均解
基本蚁群算法	429.43	464.26	444.13
思维进化算法 <sup>[7]</sup>	423.74	—	424.93
蚁群-免疫混合算法 <sup>[8]</sup>	424.64	446.55	433.24
本文算法	423.74	423.74	423.74

表2 中国旅行商问题计算结果比较

算法	中国旅行商问题(CTSP)		
	最优解	最差解	平均解
基本蚁群算法	15 602	15 800	15 622
MAX_MIN 蚂蚁算法 <sup>[9]</sup>	15 470	—	—
两段式遗传算法 <sup>[10]</sup>	15 378	15 795	15 480
本文算法	15 378	15 402	15 390

通过大量仿真实验发现,本文提出的基于思维进化的蚁群算法不但具有蚁群算法的分布式并行计算、正反馈机制和较强的鲁棒性等特点,而且可以较好地克服蚁群算法容易陷入局部最优的缺陷,具有加强解的多样性的能力,能够提高算法的全局搜索能力。

## 5 结语

本文针对蚁群算法易于与其他方法结合的特点,提出了

基于思维进化的蚁群算法,通过对旅行商问题的算例进行计算和结果比较,证明了本文算法能够克服基本蚁群算法易陷入局部最优从而导致算法过早停滞的缺陷,本文算法在最优值和稳定性等方面都取得了不错的效果,今后在算法的趋同和异化操作策略等方面将做进一步研究。

### 参考文献:

- [1] DORIGO M, MANIEZZO V, COLORNI A. Ant system: Optimization by a colony of cooperating Agents[J]. IEEE Transactions on Systems, Man, and Cybernetics, 1996, 26(1): 29-41.
- [2] DORIGO M, GAMBARDELLA L M. Ant colony system: A cooperative learning approach to the traveling salesman problem[J]. IEEE Transactions on Evolutionary Computation, 1997, 1(1): 53-66.
- [3] 段海滨. 蚁群算法原理及其应用[M]. 北京: 科学出版社, 2005.
- [4] 马良, 项培军. 蚂蚁算法在组合优化中的应用[J]. 管理科学学报, 2001, 4(2): 32-37.
- [5] 段海滨, 王道波, 于秀芬. 蚁群算法的研究现状及其展望[J]. 中国工程科学, 2007, 9(2): 98-102.
- [6] SUN CHENG-YI, SUN YAN, WEI LI-JUN. Mind-evolution-based machine learning: Frame work and the implementation of optimization[C]// Proceedings of IEEE International Conference on Intelligent Engineering Systems: INES'98. Washington, DC: IEEE Press, 1998: 355-359.
- [7] 查凯, 曾建潮. 求解 TSP 问题的思维进化算法[J]. 计算机工程与应用, 2002, (4): 102-104.
- [8] 江新姿, 汤可宗, 高尚. 蚁群算法与免疫算法的混合算法[J]. 科学技术与工程, 2008, 8(5): 1327-1330.
- [9] 李如琦, 苏媛媛. 用 MAX\_MIN 蚂蚁算法解决中国旅行商问题[J]. 湖南工业大学学报, 2007, 21(5): 48-50.
- [10] 柴世红. 两段式遗传算法求解 CTSP[J]. 大众科技, 2008(4): 17-19.

(上接第 1266 页)

### 2.3 求解丁烯烷化生产过程的约束优化问题

用 CAMDE 求解丁烯烷化生产过程的约束优化问题,这一过程约束优化的数学模型在文献[5]中已给出在此不再描述。

CAMDE 与分支定界法(aBB)、蚁群约束法(Constrained Ant Colony System, CACS)<sup>[5]</sup>、CAEP 进行比较,4 种算法都独立运行 10 次。

从表 4 看出 aBB 违背约束  $g_1$ 、 $g_3$ 、 $g_6$ , 而 CAMDE 在没有违背任何约束的条件下就得到了较好的解。与 CACS 相比,当 CACS 中的容许度  $a = 0$  或是  $a = 5 \times 10^{-4}$  时显然 CAMDE 的解要好;当容许度  $a = 5 \times 10^{-6}$  时, CACS 违背了约束  $g_3$ 。与 CAEP 相比 CAMDE 最优解要稍微好些。CACS 的最大迭代次数为 2000, CAEP 的最大迭代次数为 1000, 而 CAMDE 迭代次数为 411, 减少了计算量。所以 CAMDE 的优化性能比较令人满意。

## 3 结语

传统文化算法在种群空间中仅采用变异算子,所以进化后期收敛速度变慢,而改进差分算法对进化过程知识缺乏有效利用。针对这些问题,将改进差分进化算法引入种群空间,提出了一种基于文化算法和改进差分进化算法的混合算法。这一算法利用文化算法的框架来提取、保存和整合进化过程知识,并利用这些知识引导基于改进差分进化的种群空间进化。通过仿真结果表明这一算法有比较好的全局搜索能力,加快了收敛速度,并降低了计算量。

文化算法能够支持不同算法的混合求解问题,所以种群空间模型和表示多样化的融合有待于进一步研究,从而扩展其应用范围。

### 参考文献:

- [1] DEB K. An efficient constraint handling method for genetic algorithms[J]. Computer Methods in Applied Mechanics and Engineering, 2000, 186(2/4): 311-338.
- [2] KOZIEL S, MICHALEWICZ Z. Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization[J]. Evolutionary Computation, 1999, 7(1): 19-44.
- [3] CARLOS A COELLO C, RICARDO L, BECERRA A, et al. Constrained optimization using an evolutionary programming-based cultural algorithm[C]// Adaptive Computing in Design and Manufacturing V. Berlin: Springer-Verlag, 2002: 317-328.
- [4] LAMPINEN J. A constraint handling approach for the differential evolution[C]// Proceedings of the Congress on Evolutionary Computation 2002: CEC' 2002. Washington, DC: IEEE Computer Society, 2002, 2: 1468-1473.
- [5] 贺益君, 陈德钊. 连续约束蚁群优化算法的构建及其在丁烯烷化过程中的应用[J]. 化工学报, 2005, 59(9): 1709-1713.
- [6] 黄海燕, 顾幸生, 刘漫丹. 求解约束优化问题的文化算法研究[J]. 自动化学报, 2007, 33(10): 1115-1120.
- [7] 刘纯青, 杨萃元, 张颖. 基于文化算法的聚类分析[J]. 计算机应用, 2002, 26(12): 2953-2960.
- [8] 王奕首, 艾景波, 史彦军, 等. 文化粒子群优化算法[J]. 大连理工大学学报, 2007, 47(4): 539-544.