带有狮王竞比参数的蚁群优化算法①

李小枝[1], 沈记全[1], 杨耿帆[2]

[1](河南理工大学 计算机科学与技术学院, 焦作 454000)
[2](上海三凯建设管理咨询有限公司 技术质量部, 上海 200070)

摘　要: 由于蚁群算法采用随机选择策略, 使得进化速度较慢, 容易出现停滞现象, 从而不能对解空间进一步进行搜索, 不利于发现更好的解. 针对以上问题, 提出了一个带有狮王竞比参数的蚁群优化算法. 该算法借鉴狮子种群生存竞争中狮王法则的作用, 减少大量不必要的搜索, 从而大大缩短了求解时间, 同时又引用了最大—最小蚂蚁系统(MMAS)算法对信息素的限制, 有效地控制了搜索停滞的问题. 通过结合 MMAS 算法的仿真, 结果表明: 带有狮王竞比参数的改良算法, 在求解同样 TSP 问题时, 大大地缩短了优化时间, 并且得到了更优的解.

关键词: 蚁群算法; 竞比参数; 停滞现象; 全局优化

Ant Colony Optimization Algorithm with LionKing Competition Parameter

LI Xiao-Zhi[1], SHEN Ji-Quan[1], YANG Geng-Fan[2]

[1](School of Computer Science & Technology, Henan Polytechnic University, Jiaozuo 454000, China)
[2](Technology and Quality Department, Shanghai Sunking Construction Project Management Co. Ltd, Shanghai 200070, China)

Abstract: The random selection strategy is the basic selection method for ant colony optimization(ACO) algorithm, but it tends toward resulting in the slow convergence and premature convergence. For the above-mentioned problems, this paper proposes a new method called ant colony optimization algorithm with LionKing competition parameter(ACO-). The algorithm profits from the laws of species competition(lion) and MAX-MIN Ant System(MMAS), improved the convergence speed and utilization quality. Meanwhile, in order to avoid stagnation of the search, the range of possible pheromone trails on each solution component is limited to a maximum-minimum interval. In the end, an example of Traveling Salesman Problem(TSP) is given in the paper, which is simulated by using MMAS and ACO-. The simulation resules show that the kind of advanced ant colony algorithm improves the nature of random search, so the algorithm can converge more rapidly to the optimization answer.

Key words: ant colony algorithm; competition parameter; stagnation behavior; global optimization

社会性动物的集群活动往往可以产生惊人的自组织行为, 如个体行为显得简单、盲目的蚂蚁组成蚁群以后就能发现从蚁巢到食物源的最短路径. 生物学家研究发现蚁群根据个体留下的外激素(信息素)来间接通信和相互协调就可以找到最优路径. 基于此, 意大利学者 Marco Dorigo 等人提出了一种基于蚂蚁种群的启发式并行智能进化算法——蚁群算法(Ant Colony Optimization, ACO)[1], 并解决了一系列的组合优化问题[2]. 大量实验表明: 由大量蚂蚁组成的集体行为表现出一种信息正反馈现象, 通过路径的选择机制、信息素更新机制和蚂蚁个体之间的协调机制来寻找最优路径. 蚁群算法不仅能够智能搜索、全局优化, 而且具有稳健性、正反馈、分布式计算、易于其他算法结合等优点[3,4]. 但蚁群算法仍然存在一些缺陷, 后来也出现了许多改进的蚁群算法, 如基于带精英策略的蚂蚁系统的蚁群算法是最早的改进的蚁群算法, 后来提出

① 收稿时间:2011-12-09;收到修改稿时间:2012-01-13

更具代表性的基于最大—最小蚂蚁系统(MAX-MIN Ant System,MMAS)[5]的蚁群优化算法, 近些年又提出了一种自适应信息素改进蚁群算法[6], 改进的种群分类蚁群算法[7], 带有征税算子的蚁群优化算法[8]等等. 但蚁群算法在构造解的过程中, 都是利用随机选择策略, 使得进化速度比较慢, 容易出现停滞. 针对这一问题, 本文提出了一种新的蚁群优化算法, 它是在MMAS 算法的基础上, 通过增加狮王竞比参数, 有效地解决了 ACO 算法优化效率低和搜索停滞问题.

1 蚁群算法的数学模型

20 世纪 90 年代意大利学者 M.Dorigo, V.Maniezzo, A.Colorni 等从生物进化的机制中受到启发, 通过模拟自然界蚂蚁搜索路径的行为, 提出了一种新型的模拟进化算法——蚁群算法[9]. 蚁群算法(ACA)包括两部分: 蚂蚁的活动和信息素的更新. 为了更好地阐述蚁群算法的数学模型, 本文结合旅行商问题(TSP)来说明, 把旅行商看作蚂蚁, 让蚂蚁去搜索城市, 求解满足条件的最优访问路程长度.

第一部分: 蚂蚁运动是根据路径上信息素量来决定行走方向的. 同时, 蚂蚁利用移动转移概率和信息素更新机制, 避免了所有蚂蚁迅速趋向于搜索空间的同一部分[10].

假设初始时刻各个路径上的信息素量相等为 $\tau_{ij}(0)=C(C\text{为常数})$. 蚂蚁 k(k=1,2,3,⋯, m)在运动过程中根据路径上信息素量决定转移方向. 蚂蚁系统使用随机比例的转移规则. 它给出了在 t 时刻位于某城市 i 的蚂蚁 k 选择移动到另一城市 j 的转移概率 $P_{ij}^{k}(t)$ 为:

$$P_{ij}^{k}(t)=\begin{cases}\dfrac{[\tau_{ij}(t)]^{\alpha}*[\eta_{ij}]^{\beta}}{\sum\limits_{s\in Allowed_k}[\tau_{ij}(t)]^{\alpha}*[\eta_{ij}]^{\beta}} & j\in Allowed_k\\ 0 & j\notin Allowed_k\end{cases} \tag{1}$$

其中, α 和 β 参数分别表示蚂蚁在运动中所积累的信息素和启发信息因子的相对重要程度. $Accessed_k$ =(0, 1, 2, ⋯, n-1)表示蚂蚁 k 下一步允许选择的城市. 为了避免蚂蚁对同一个城市的多次访问, 每只蚂蚁都保存一个已访问列表 $Accessed_k$, 用于记录蚂蚁 k 已经访问过的城市, 列表会随着活动过程做动态更新; η_{ij} 为启发函数:

$$\eta_{ij}=1/d_{ij} \tag{2}$$

其中, d_{ij} 表示相邻两城市间的距离. d_{ij} 越小, 则 η_{ij} 就越大, $P_{ij}^{k}(t)$ 也就越大.

第二部分: 在蚁群算法中, 信息素的更新方式分为局部更新和全局更新. 具体的更新方法又分为信息素的增强和挥发减弱. 其更新方式如下:

$$\tau_{ij}(t+1)=(1-\rho)*\tau_{ij}(t)+\Delta\tau_{ij}(t,\ t+1) \tag{3}$$

$$\Delta\tau_{ij}(t,t+1)=\sum_{k=1}^{m}\Delta\tau_{ij}^{k}(t,t+1) \tag{4}$$

其中, ρ 为信息素挥发因子, $1-\rho$ 表示信息素的残留系数, 为了防止信息素的无限积累, ρ 的取值范围为 $\rho\subset[0,1]$; $\Delta\tau_{ij}^{k}(t,t+1)$ 表示第 k 只蚂蚁从时间 t 到 t+1 过程中, 留在路径(i,j)上的信息素量, 其值根据不同的蚂蚁系统表现的优劣程度不同.

在蚁密 ant-density 系统中

$$\Delta\tau_{ij}^{k}(t,t+1)=\begin{cases}Q & \text{如果第k只蚂蚁经过路径}\ (i,j)\\ 0 & others\end{cases} \tag{5}$$

在蚁量 ant-quantity 系统中

$$\Delta\tau_{ij}^{k}(t,t+1)=\begin{cases}Q/d_{ij} & \text{如果第k只蚂蚁经过路径}\ (i,j)\\ 0 & others\end{cases} \tag{6}$$

在蚁环 ant-cycle 系统中

$$\Delta\tau_{ij}^{k}(t,t+1)=\begin{cases}Q/L_{ij} & \text{如果第k只蚂蚁经过路径}\ (i,j)\\ 0 & others\end{cases} \tag{7}$$

其中, Q 为一个常数, 为蚂蚁循环一周后所释放在所有经过路径上的信息素总量; L_{ij} 为第 k 只蚂蚁在本次循环中走过的路径长度, 其特点是行走的路径越短对应保存的信息素值就越大, 反之, 保存的信息素值就越小[11]. 从公式(5)到(7)可以得出, 蚁密系统和蚁量系统采用的是局部信息更新, 蚂蚁每走一步就更新一次, 而蚁环系统采用的是全局信息更新, 只有蚂蚁循环结束后, 在根据的长度平均更新信息素, 故在 TSP 问题求解中性能更好.

2 狮王法则

面对大自然的严峻地淘汰竞争, 为什么狮子可以保证自身种群的繁衍不息, 可持续发展? 这都要归咎于狮群内部竞争的自我淘汰, 使种群保持对外竞争的优势. 狮王在狮群中往往是最高统治者, 它通过合理

的分工, 帮助族群捕猎, 从而保护族群, 使其健康发展. 随着年轻雄狮的成长, 狮王总会迎接新的挑战, 从而抉择出新的更强壮的狮王, 摒弃老狮王. 如此循环, 以确保狮王总是最强壮、最优秀的狮子.

蚁群算法在优化过程中, 信息素全局更新时, 往往待所有蚂蚁都遍历一遍后, 再进行结果比较, 最后根据最优蚂蚁的结果进行更新信息素, 这样就造成大部分的蚂蚁得到的结果没有被利用, 浪费时间, 造成不必要的损失. 鉴于狮王法则, 现引入一个狮王竞比参数, 以作为参考. 在求解一个问题前, 如果蚁群算法首次进行优化, 就可以先派一个蚂蚁去完成任务, 从而将其结果赋值给竞比参数; 如果历史库里记录有该问题优化结果, 就将其结果赋值给竞比参数, 然后再进行优化. 这样, 蚂蚁在优化问题时, 根据自身记录的间接数据与竞比参数作比较, 如果数据比竞比参数大, 就直接终止行动, 初始化其对应的已访问列表 $Accessed_k$, 再投入新一轮的优化. 总之, 引进狮王竞比参数, 就是为了摒弃那些不良的搜索, 从而缩短蚁群优化的时间, 对蚁群优化有重大的意义.

3　带有狮王竞比参数 $\mathrm{L_{lk}}$ 的蚁群优化算法的设计

在此算法中, 蚂蚁算法借鉴了 MMAS 算法, 该算法可以防止算法的过早停滞, 并且在有效性方面比蚂蚁系统(AS)算法[12]有较大改进. 下面是 ACO-$\mathrm{L_{lk}}$ 算法的核心设计:

1) 信息素的初值设置　ACO-$\mathrm{L_{lk}}$ 算法把各路径的信息素初值也设为最大值 $\tau_{\max}$;

2) 信息素的更新方式　在 ACO-$\mathrm{L_{lk}}$ 算法中, 只要蚂蚁完成一次循环, 就更新信息素轨迹. 因此, 经修改后的轨迹更新规则如下:

$$\tau_{\mathrm{ij}}(t+1)=\rho\tau_{ij}(t)+\Delta\tau(t) \tag{8}$$

其中, $\Delta\tau(t)=1/L_{lk}(t)$, $\mathrm{L_{lk}}(t)$ 表示第 t 次迭代后的最优解的值;

3) 信息素的限制　现实中, 不管 ACO 算法选择迭代最优还是全局最优蚂蚁来更新信息素, 都可能导致搜索停滞. 在 TSP 中, 如果某一条边上的信息素过高, 蚂蚁就会更倾向选择这个解元素, 正反馈机制会进一步增强该解元素上的信息素, 使得蚂蚁重复建立同一个解, 对搜索空间的探索将停止. 但是, 通过限制信息素的值域, 使得信息素差异不会过大, 就能避免搜索停滞的问题. 这里根据 MMAS 对信息素的最大值和最小值分别施加了 $\tau_{\max}$ 和 $\tau_{\min}$ 限制, 限制如下公式(9), 使得 $\tau_{\min}<\tau_{ij}(t)<\tau_{\max}$;

$$\tau_{\mathrm{ij}}(t)=\begin{cases}\tau_{\min} & \tau_{ij}(t)<\tau_{\min}\\ \tau_{\max} & \tau_{ij}(t)>\tau_{\max}\end{cases} \tag{9}$$

4) 狮王竞比参数的初始设置　算法开始, 先让一只蚂蚁出发, 等这只蚂蚁循环结束后, 它行走的路程 s_0 就为 $\mathrm{L_{lk0}}$;

5) 狮王竞比参数的更新　首先, 对蚂蚁做了一个改进, 在 ACO-$\mathrm{L_{lk}}$ 算法中要求每只蚂蚁都有一定的记忆功能, 可以记住行走的距离 $\mathrm{s_k}$ 且可以对其行走的路程做简单的加运算, 并实时根据公式(10)更新自己的 $\mathrm{s_k}$ 参数. 在 TSP 中, 这个过程实现比较简单, 因为城市间的距离是已知的.

$$\mathrm{s_k}(i+1)=\mathrm{s_k}(i)+\Delta d_{\mathrm{k}} \tag{10}$$

其中, i 为城市的个数; $\Delta \mathrm{d_k}$ 为蚂蚁 k 从 i 城市走到 i+1 城市所行的距离.

$\mathrm{L_{lk}}$ 竞比参数的更新原则: 蚂蚁每到达下一个城市都要将自己更新过的数据 $\mathrm{s_k}$ 与 $\mathrm{L_{lk}}$ 进行比较, 如果 $\mathrm{s_k}>\mathrm{L_{lk}}$,则直接停止搜索, 初始化可选列表 $Accessed_k$; 否则, 继续前行. 直到 $Accessed_k$ 为空, 同时 $\mathrm{s_k}<\mathrm{L_{lk}}$,最后将 $\mathrm{L_{lk}}=\mathrm{s_k}$.

总之, 从以上算法设计可以看出, 竞比参数 $\mathrm{L_{lk}}$ 就相当于狮王, 它总是记录当前的最优解, 蚂蚁在执行任务时, 总是实时与 $\mathrm{L_{lk}}$ 比较, “弱者淘汰, 强者胜出”的规则, 使得 ACO-$\mathrm{L_{lk}}$ 算法最终收敛于 $\mathrm{L_{lk}}$, 即狮王竞比参数 $\mathrm{L_{lk}}$ 就是 ACO-$\mathrm{L_{lk}}$ 算法的最优解.

4　实验分析

本文采用 MATLAB7.0 开发的仿真平台对 MMAS 算法和 ACO-$\mathrm{L_{lk}}$ 算法进行了模拟. 实验选取 31 个城市的 TSP 问题为测试问题, 其中 31 个城市的坐标已知, 用文本形式导入城市平面坐标数据. 设置算法参数为 m=20、α=1、β=5、ρ=0.5、Q=50、$\tau(0)$=1, 最大迭代次数 $\mathrm{N_{cmax}}$=100*m. 其中 MMAS 算法信息素上下限采用文献[5]中的设置: $\tau_{\max}$=10、$\tau_{\min}$=0.01.

实验进行了 5 次, 其仿真结果如表 1 所示, 分别记录实验得出最优路径的长度和花费的时间, 然后计算出其平均值. 添加狮王竞比参数的蚁群优化算法所获

得的平均路径长度明显优于未添加狮王竞比参数的 MMAS 算获得的路径. 从表中看在同一实验环境下 ACO-L_{lk} 算法找到的平均路径长度为 14.1398，优于 MMAS 算法的 16.0634. 同时从收敛需要是时间上看 ACO-L_{lk} 算法优于 MMAS 算法 1.01.

表 1 MMAS 算法和 ACO-L_{lk} 算法

算法 / 记录项 / 实验次数	MMAS		ACO-L_{lk}	
	时间(ms)	最优路径长度(km)	时间(ms)	最优路径长度(km)
1	3.2	16.037	2.2	13.999
2	3.3	15.931	2.3	14.201
3	3.3	16.361	2.3	14.118
4	3.2	15.879	2.2	13.894
5	3.3	16.109	2.2	14.487
平均值	3.26	16.0634	2.24	14.1398

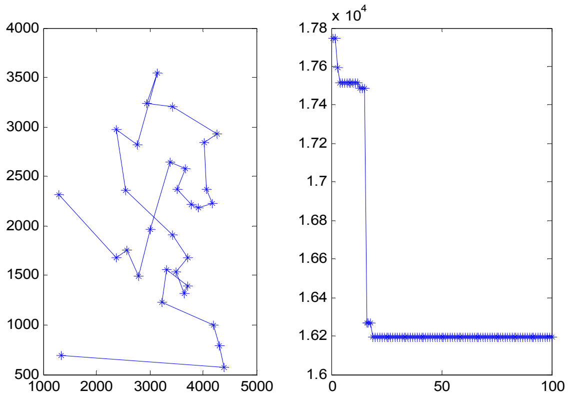

图 1 MMAS 算法找到的最优路径和进化曲线

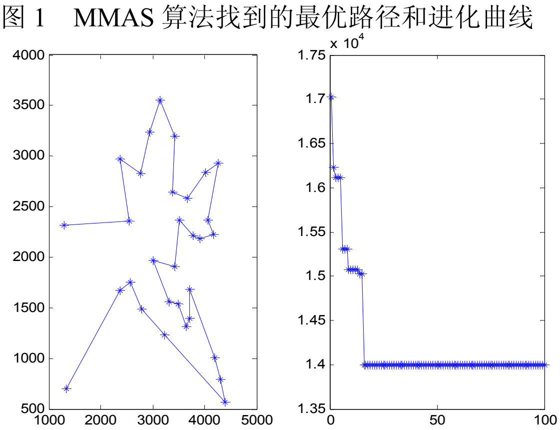

图 2 ACO-算法算法找到的最优路径和进化曲线

5 总结

鉴于狮群法则通过内部竞争自我淘汰，使得种群繁衍不息，持续发展的重要作用，本文增加了狮王竞比参数，在蚂蚁搜索路径时，按一定规则和竞比参数进行比较，及时终止不良搜索，并且也不需要每次蚁群循环完毕后，进行比较找出最优解，有效地提高了优化效率. 另外, ACO-算法继承了 MMAS 算法调整信息素强度的差异，抑制停滞现象的出现. 实验数据也充分说明了带有狮王竞比参数的蚁群优化算法可以有效地提高蚁群算法的全局优化能力.

参考文献

1 Dorigo M, Stutzle T. Ant Colony Optimization Cambridge, MA: MIT Press, 2004. 27–29.

2 Ellabib I, Calamai P, Basir O. Exchange strategies for multiple ant colony system. Journal of Information Sciences, 2007,177(5):1248–1264.

3 Bontoux B, Feillet D. Ant colony optimization for the traveling purchaser problem. Computers & Operations Research, 2008,25(2):628–637.

4 Baskan O, Haldenbilen S, Ceylan H, et al. A new solution algorithm for imporving performance of ant colony optimization. Applied Mathematics and Computation, 2009,211(1): 75–84.

5 Stutzle T, Hoos HH. MAX-MIN ant System. Future Generation Computer Systems,2000,16(3):889–914.

6 周燕霞.一种自适应信息素改进蚁群算法.计算机系统应用,2009,18(10):57–60.

7 刘芳,李义杰.改进的种群分类蚁群算法及应用.计算机系统应用,2010,19(1):144–148.

8 郑松,李春富,王春林,葛铭,薛安克.带有征税算子的改进蚁群优化方法.计算机工程与应用,2011,47(15):32–35.

9 Dorigo M, Di Caro G. Ant Algorithms for Discrete Optimization. Artificial Life, 1999,5(3);137–172.

10 李士勇,等.蚁群算法及其应用.哈尔滨:哈尔滨工业大学出版社,2004.22–26.

11 吴启迪,汪镭.智能蚁群算法及应用.上海:上海科技教育出版社,2004.20–36.

12 杨剑.蚁群算法及其应用研究[博士学位论文].杭州:浙江大学,2007.6–12.