一种改进的蚁群算法求解最短路径问题

毕 军 付梦印 张宇河
(北京理工大学自动控制系,北京 100081)
E-mail:bilinghc@263.net

摘 要 蚁群算法是一种新型的模拟进化算法,为求解复杂的组合优化问题提供了一种新的思路。该文应用蚁群算法求解最短路径问题,对算法的选择策略、局部搜索、信息量修改三方面进行改进,使算法不易陷入局部最优解,并且能较快地收敛到全局最优解。实验结果表明,改进方法是合理的、有效的。

关键词 蚁群系统 最短路径 模拟进化算法

文章编号 1002-8331-(2003)03-0107-03 **文献标识码** A **中图分类号** TP301.6

An Improved Ant Colony Algorithm for the Shortest Path Problem

Bi Jun Fu Mengyin Zhang Yuhe
(Department of Automatic Control,Beijing Institute of Technology,Beijing 100081)

Abstract: Ant colony system is a novel simulated evolutionary algorithm,which provides a new method for complicated combinatorial optimization problems.The algorithm is used for the shortest path problem.It is improved in three parts of selection strategy,local search,and information modification,so that it can not easily run into the local optimum and can converge at the global optimum.The results of experiment show that the improved algorithm is valid.

Keywords: Ant colony system,Shortest path,Simulated evolutionary algorithm

1 引言

最短路径问题是指在一个赋权图的两个节点之间找出一条具有最小权的路径,是图论中研究的一个重要问题[1]。交通网络的线路分布、旅游线路的选择、城市规划及工程建设中的管道网的铺设等问题都与寻找一个图的最短路径问题密切相关,因此对最短路径问题的研究具有重要意义和实用价值。

随着人们对 NP-hard 组合优化问题的研究,提出了许多用以解决该类问题的进化算法,如遗传算法、进化规划、进化策略等。蚁群算法是最近几年才提出的一种新型的模拟进化算法,它是由意大利学者 M.Dorigo,V.Maniezzo 等人首先提出的[2,3],他们称之为蚁群系统(Ant Colony System,ACS)。并应用该算法求解 TSP 问题、分配问题、job-shop 调度问题,取得了较好的结果[3-5]。该文改进了基本的蚁群算法,解决了基本算法在求解最短路径中出现的问题,取得了理想的实验结果。

2 基本蚁群算法原理

蚁群算法是一种模拟进化算法,是受到对真实蚁群行为的研究的启发而提出的。为了说明蚁群系统的原理,先简介一下蚂蚁搜索食物的具体过程。象蚂蚁这类群居昆虫,虽然单个昆虫的行为极其简单,但群体却表现出复杂的行为,能找到由蚁巢到食物源的最短路径。仿生学家经过研究发现,蚂蚁个体之间是通过一种称为外激素(pheromone)的物质进行信息传递的。蚂蚁在运动过程中,能够在它所经过的路径上留下该种物质,而且在运动过程中能够感知这种物质,并以此指导自己的运动方向,蚂蚁倾向于朝着物质强度高的方向移动。因此,由大量蚂蚁组成的群体行为便表现出一种信息正反馈现象:某一路径越短,走过的蚂蚁越多,路径上留下的信息物质越多,则后续蚂蚁选择该路径的概率就越大,并且留下相应的信息物质。最后,蚂蚁个体之间就是通过这种信息物质交流,搜索到一条从蚁巢到食物源的最短路径[2-5]。

蚁群算法是模拟蚁群行为的一种随机搜索优化算法,主要由四个部分组成[5]:选择策略;信息量的局部更新;求局部最优解的局部搜索算法;信息量的全局更新。

3 求解最短路径的蚁群算法的实现

3.1 基本算法的设计

开始时,所有的 m 个蚂蚁都集中在起点 S 处。蚂蚁 i 从 S 点出发,按照选择策略,从和 S 相关联的边的集合中,选择一条边;然后,再从这条边的另一节点 a 开始,从和 a 相关联的边的集合中,选择另一条边;以此类推,直到搜索到终点 T,于是,蚂蚁 i 得到一个从 S 到 T 的解。每当蚂蚁选择完一条边后,就马上更新边上的信息量(局部更新)。蚂蚁 i 搜索完后,蚂蚁 j 从 S 出发,按照和 i 相同的方法,搜索出一条从 S 到 T 的路径,得到一个解。直到所有的 m 个蚂蚁都进行完搜索,得到 m 个解(包括重复的)。以 m 个解为基础,采用局部搜索算法,进行局部搜索,得到局部最优解。根据局部最优解全局计算信息增量(全局更新),全局更新后,继续迭代直到满足停止条件,停止条件为最大迭代次数。在所求得的所有局部最优解中,值最小的解为所求的全局最优解,即最短路径的距离值。用基本的蚁群算法求解最短路径问题的主要实现步骤如下[5,6]:

(1)信息初始化:算法开始运行时,赋予每条边上相等数量的信息量。

作者简介:毕军,男,1973 年生,博士生,主要从事定位定向技术及网络优化方向的研究。付梦印,男,1964 年生,教授,博士生导师,主要从事定位定向技术的研究。张宇河,男,1940 年生,教授,博士生导师,主要从事运动控制和导航技术的研究。

(2)选择策略:位于第 i 个节点的蚂蚁 k,按以下选择策略选择边 (i,j):

$$j=\begin{cases}\underset{s\in V_i}{\operatorname{argmax}}\ t(i,s) & q\le q_0\\ h & q>q_0\end{cases} \tag{1}$$

其中,V_i={和 i 相连的节点}-{蚂蚁 k 已访问过的节点},即每个蚂蚁对每个节点最多只允许访问一次,对不连通的点,则赋给一个足够大的惩罚值;$\operatorname{argmax}t(i,s)$表示在与 i 相关联的边的集合中,具有最大信息量的边的另一节点;q_0 为给定参数,$0<q_0<1$,q 是(0,1)内服从均匀分布的随机变量;h 依照以下概率在 $U_i=V_i-\operatorname{argmax}t(i,s)$内随机取值:

$$P(i,h)=\frac{t(i,h)}{\sum_{s\in U_i}t(i,s)} \tag{2}$$

$t(i,s)$表示边 (i,s)上的信息量,式(2)采用轮盘赌[7]的方式实现。若 $q_0=0.9$,式(1)表明和 i 相关联的信息量最大的边以高概率 0.9 被蚂蚁选中,其余的边以 0.1 的概率按式(2)参与选择。

(3)局部更新信息量:当蚂蚁选中边 (i,j)后,就更新边 (i,j)上的信息量:

$$t(i,j)=\alpha * t(i,j)\quad 0<\alpha<1 \tag{3}$$

每当蚂蚁选中一条边后,就按(3)式更新减少边上的信息量,从而增加蚂蚁选择其它边的概率。

(4)局部搜索:当 m 只蚂蚁从 S 到 T 搜索完后,则求得 m 个解。为了尽可能遍历所有解,分别在这些解的邻域中,采用局部搜索算法(例如 2-OPT)[7],求出局部最优解。

(5)全局更新信息量:当所有的 m 只蚂蚁都得到局部最优解后,全局更新边上的信息量:

$$t(i,j)=\beta * t(i,j)+\sum_{k=1}^{m}\Delta t^k(i,j) \tag{4}$$

$$\Delta t^k(i,j)=\begin{cases}\dfrac{wconst}{d_k} & \text{若蚂蚁 } k \text{ 经过边}(i,j)\\ 0 & \text{若蚂蚁 } k \text{ 没有经过边}(i,j)\end{cases}$$

β 是给定参数,$0<\beta<1$,表示随着时间的推移,以前留下的信息逐渐消逝,用参数 $1-\beta$ 表示信息消逝程度;$\Delta t(i,j)$是边 (i,j)上的信息增量;d_k 是蚂蚁 k 求得的局部最优解;$wconst$ 是常量。

(6)求全局最优解:到当前迭代次数为止,所建立的所有局部最优解中,值最小的解作为当前迭代次数的全局最优解。

3.2 算法的改进

笔者从一系列的实验结果发现,用基本蚁群算法求解最短路径过程中常出现两个问题:搜索陷入局部最优解,即搜索进行到一定程度后,所有的个体发现的解基本完全一致,出现停滞现象,不能再对解空间进一步搜索,导致可能无法找到全局最优解;收敛到全局最优解的时间长,求解结果反复在局部最优解和全局最优解之间振荡。

为了解决出现的问题,文章从蚁群算法的选择策略、局部搜索算法以及信息量的全局修改三方面进行改进。

(1)分析式(1)知,若参数值较大(如接近 1),则多数蚂蚁易选择信息量最大的边,这样在搜索过程中可能容易出现多数蚂蚁搜索到相同的路径,使得搜索到的解空间较小,不利于发现全局最优解,算法容易收敛到局部最优解。若 q_0 较小(如接近 0),则信息量最大的边被选择的概率小,其它边被选择的概率大,能扩大搜索到的解空间,但搜索呈现一定的盲目性,不容易收敛。综合考虑这两个方面,采用变参数 q_0,即 q_0 的值和迭代次数 $cycle$ 有关,是迭代次数的分段函数:

$$q_0=\begin{cases}c_1 & 0<cycle\le n_0\\ c_0 & n_0<cycle\le n_1\\ c_1 & n_1<cycle\le n_{max}\end{cases} \tag{5}$$

式中 n_0,n_1 分别为迭代次数;n_{max} 是最大迭代次数;$0.8<c_1<1$,$0<c_0<0.4$。式(5)表明,算法开始运行时,以较大的概率选择信息量最大的边,直到迭代到第 n_0 次。此后,为了防止搜索陷入局部最优解,便于发现全局最优解,以较大的概率在局部最优解的邻域中搜索其它解,直到迭代到第 n_1 次。为了使算法收敛到全局最优解,n_1 次后再以较大的概率选择信息量最大的边,直到运行结束。

(2)选择合适的局部搜索算法,能扩大搜索空间,有利于发现全局最优解。受遗传算法的启发[7],将变异思想引入到局部搜索算法,给出一个具有变异特征的局部搜索算法。设蚂蚁 k 搜索到从 S 到 T 的路径 e_k 为:$a_0(S),a_1,a_2,\cdots,a_j,\cdots,a_{n+1}(T)$。$a_i$ $(0\le i\le n+1)$分别为图中的节点。给定变异概率 P_b,在 n 次循环中,产生 n 个(0,1)内均匀分布的随机数。若第 $j(1\le j\le n)$个随机数 $r_j<P_b$,则第 a_j 个节点要发生变异。设存在节点 a_{j-2},a_j 和 a_{j-1} 相连外,有 w 个节点 $\overset{1}{a}_{j-1},\overset{2}{a}_{j-1},\cdots,\overset{w}{a}_{j-1}$ 和 a_{j-1} 相连。进行 w 次循环,产生 w 个(0,1)内均匀分布的随机数,若第 $h(1\le h\le w)$个随机数 $r_h<P_s$(P_s 为给定的选择概率),则用 $\overset{h}{a}_{j-1}$ 代替 a_j,得到一个解 e_k':$a_0(S),a_1,a_2,\cdots,\overset{h}{a}_{j-1},\cdots,a_{n+1}(T)$。若路径 e_k'的值$|e_k'|<|e_k|$,则用 e_k'作为蚂蚁 k 搜索到的局部最优解。如果该算法和常用的 2-OPT 局部搜索算法相结合,则更能加大对解空间的探测力度。

(3)为了使算法较快地收敛到全局最优解,采用下式对信息量全局更新:

$$t(i,j)=\begin{cases}\text{式(4)} & 0<cycle\le n_1\\ \beta * t(i,j)+\Delta t(i,j) & n_1<cycle\le n_{max}\end{cases} \tag{6}$$

$$\Delta t(i,j)=\begin{cases}\dfrac{wconst}{d_g} & \text{若边}(i,j)\text{是全局最优解(当前迭代次数)的一条边}\\ 0 & \text{若边}(i,j)\text{不属于全局最优解}\end{cases}$$

d_g 是全局最优解,即最短路径的距离值。式(6)表明,经过 n_1 次迭代后,只修改全局最优解边上的信息量,使最短路径边上的信息量增加较快,算法容易收敛到全局最优解。

4 实验结果

图 1 是给定的具有 15 个节点的加权有向图,该图以邻接表[1]的数据结构在计算机中存储。节点 1 为起点 S,节点 15 为终点 T,求 S 到 T 的最短路径。蚁群算法的参数 $\alpha=\beta=0.9$;$wconst=50$;蚂蚁数量 $m=8$;信息量初始化为 10;迭代次数 $cycle$ 最大为 250。图 2 是基本蚁群算法每一次迭代的求解结果,给定参数 $q_0=0.83$。分析图 2 知,算法迭代 100 次以后,求解的值基本收敛于 16,收敛的路径为:1—2—11—13—15,算法陷入局部最优解。图 3 是改进的蚁群算法的求解结果,变参数 q_0 按照式(5)改变,当迭代 $0<cycle\le 50$ 时,$q_0=0.88$,当 $50<cycle\le 100$ 时,$q_0=0.26$,当迭代次数 $100<cycle\le 250$ 时,$q_0=0.88$。局部搜索算法中给定变异概率 $P_b=0.3$,选择概率 $P_S=0.3$。从图 3 知,迭代次数从 50 到 100 之间,算法扩大了搜索到的解空间,防止陷入局部最优。迭代 100 次以后,改进的算法求解值收敛于 14,收敛的路径为:1—3—9—6—12—15,是图 1 的最短路径,

表明改进的算法能较快地收敛于全局最优值。

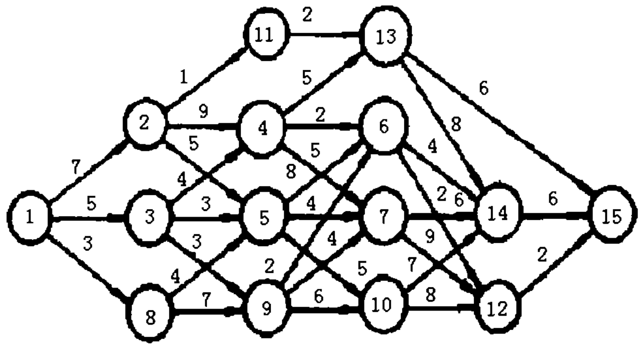

图 1 15个节点的加权有向图

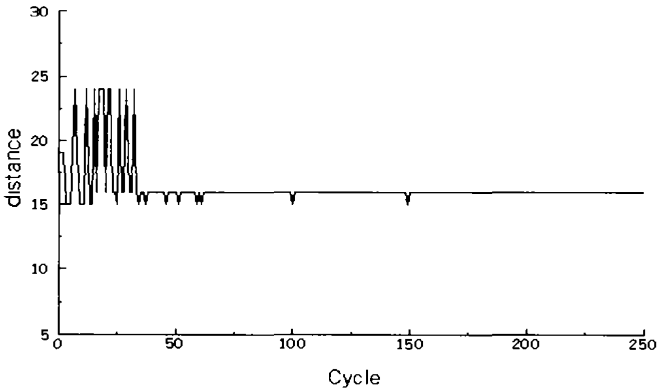

图 2 基本蚁群算法求解图

5 结论

该文给出的改进方法，较好地解决了基本蚁群算法在求解最短路径中容易陷入局部最优解的问题，实验结果表明改进的算法能较快地收敛到全局最优解。蚁群算法是一种新型的进化算法，其研究刚刚起步不久，有许多问题有待解决。从理论上对该算法进行更深刻的研究将是进一步研究的内容。

（收稿日期：2002年1月）

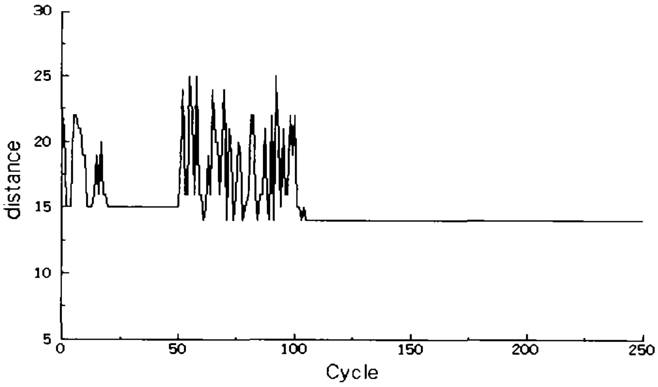

图 3 改进的蚁群算法求解图

参考文献

1.严蔚敏，吴伟民.数据结构[M].北京：清华大学出版社，1997
2.Colorni A,Dorigo M,Maniezzo V.Distributed optimization by ant colonies[C].In:Proc of 1st European Conf Artificial Life Pans,France:Elsevier,1991:134~142
3.Colorni A,Dorigo M,Maniezzo V et al.Ant system for job-shop scheduling[J].Belgian J of Operations Research Statistics and Computer Science,1994;34(1):39~53
4.Dorigo M,Maniezzo V,Colorni A.Ant System:Optimization by a Colony of Cooperating Agents[J].IEEE Transactions on Systems,Man,and Cybernetics-part B,1996;26(1):29~41
5.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
6.林锦，朱文兴.凸整数规划问题的混合蚁群算法[J].福州大学学报（自然科学版），1999;27(6):5~9
7.邢文训，谢金星.现代优化计算方法[M].北京：清华大学出版社，1999

（上接94页）

对于图3所示的图符，提取了13维特征：图符外接矩形的高和宽，图符的外边缘长度，图符的前景像素点的数目，图符所覆盖的面积，图符的圆形度，图符的内环数目，图符内环周长的方差与均值之间的比值，内环高度的最大值和最小值，内环宽度的最大和最小值，图符的轴向最大投影值和同向外接矩形的投影值。在实测过程中，在不同大小的训练数据集上把由单决策树构成的学习和分类系统与由MCMD构成的学习和分类系统进行判决精度比较。测试情况如图4所示。

6 结论

文章针对标准决策树算法在解决多模式类学习问题上的不足，提出了相应的解决方法MCMD。标准决策树算法在决策树的每个节点计算中选取对 M 个模式类别总的贡献 $G_n=\sum_m(\alpha_m G_{n,m})$ 最大的属性。这种方法对2类学习问题（即 $M=2$）是可解决的但对多类问题却是无法满足的。而该文使用的MCMD则借鉴了覆盖算法解决多类学习的优点，用多决策树合成的方法比较好地处理了这种问题。从表1的实验结果可以看出，即使在模式类别不是很多的情况下（3类和6类数据）标准决策树算法就存在不足。而在模式类极多的图形识别问题上（60个模式类）这种差别就更为明显（如图4所示）。

用多棵决策树进行表决并合成分类器的方法，例如Boosting[4]和Bagging[5]，是近年来决策树算法研究的新的领域。而该文所研究的MCMD是采用多棵决策树进行逻辑合成，而不是基于投票表决的方法。MCMD最终得到的仍然是一棵完整的决策树，它保留了符号学习的基本性质。MCMD已经成功地应用于笔者研制的城市地形图的图符自动输入系统。在总的特征空间保持不变的条件下，MCMD还可以满足增加可识别的模式类别时减少系统学习开销的要求。（收稿日期：2002年1月）

参考文献

1.Michalski R S,Mozetic I,Hong J R et al.The Multi-purpose Incremental Learning System AQ15 and its Testing Application to Three Medical Domains.AAAI,1986
2.University of California.Irvine Repository of Machine Learning database,obtainable by anonymous FTP to ftp.ics.uci.edu in the /pub/machine-learning-databases directory
3.Qualian J R. C4.5:Programs for Machine Learning.San Mateo,Calif.,Morgan Kaufmann,1993
4.Freund Y,Schapire R E.Experiments with a new boosting algorithm[C].In:Proceedings Thirteenth International Conference on Machine Learning,San Francisco:Morgan Kaufmann,1996:148~156
5.Breiman L.Bagging predictors[J].Machine Learning,1996;24:123~140