用于求解函数优化的蚁群算法设计

杜呈欣[1],陈小强[2],熊伟清[3]
DU Cheng-xin[1],CHEN Xiao-qiang[2],XIONG Wei-qing[3]

1.兰州交通大学 电子与信息工程学院,兰州 730070
2.兰州交通大学 自动化与电气工程学院,兰州 730070
3.宁波大学 信息科学与工程学院,浙江 宁波 315211
1.School of Electronics and Information Engineering,Lanzhou Jiaotong University,Lanzhou 730070,China
2.School of Automatization and Electric Engineering,Lanzhou Jiaotong University,Lanzhou 730070,China
3.Faculty of Information Science and Engineering,Ningbo University,Ningbo,Zhejiang 315211,China
E-mail:duchengxin@gmail.com

DU Cheng-xin,CHEN Xiao-qiang,XIONG Wei-qing.Ant colony algorithm design for function optimization.Computer Engineering and Applications,2007,43(25):57-59.

Abstract: To solve function optimization problem,we have described ant colony algorithm adding binary coding of genetic algorithm and developed pheromone update strategy.In the experiment,we have used a new structure of searching matrix.Because the matrix's dimension decreases,the performance of algorithm improves considerably.The improved algorithm has been tested for variety of different classical test functions.And the algorithm can handle these optimization problems very wel1.

Key words: ant colony algorithm;function optimization;TSP;genetic algorithm

摘 要:为了求解一般的函数优化,在对标准蚁群算法研究的基础上,将遗传算法的编码方式引入蚁群算法,对蚁群算法的信息素更新进行改进,并提出一种搜索矩阵表达方式,减少了搜索矩阵的规模,从而提高了搜索效率。通过对几个经典测试函数的求解,证明了算法的有效性。

关键词:蚁群算法;函数优化;TSP;遗传算法
文章编号:1002-8331(2007)25-0057-03 **文献标识码:**A **中图分类号:**TP301.6

1 引言

生物学研究表明在自然界中一群互相协作的蚂蚁能够找到食物源和巢之间的最短路径,而单只蚂蚁则不能。蚂蚁间相互协作的方法是它们在所经过的路上留下一定数量的信息素(迹),该迹能被其它蚂蚁检测出来,一条路上的迹越多,其它蚂蚁将以越高的概率跟随此路径,从而该路径上的迹会被加强。蚁群算法(Ant Colony Algorithm)是受到人们对蚁群集体行为的研究成果的启发而提出的一种基于种群的模拟进化算法,属于随机搜索算法。该算法思想由意大利学者 M.Dorigo 等人首先提出[1,2],充分利用了蚁群搜索食物的过程与著名的旅行商问题(TSP)之间的相似性,通过人工模拟蚂蚁搜索食物的过程来求解 TSP,为了区别于真实蚂蚁群体系统,称这种算法为“人工蚁群算法”。该模型已成功应用于求解 TSP 问题、分配问题、job-shop 调度问题等 NP-complete 的组合最优化问题[2,3],结果可与模拟退火,遗传算法等通用的启发式算法相媲美。虽然对此方法的研究刚刚起步,但研究表明蚁群算法在求解复杂优化问题(特别是离散优化问题)方面的一些优越性,证明它是一种很有发展前景的方法,这种带有构造性特征的新的搜索方法已产生并被广泛应用。

蚁群算法是一种随机搜索算法,与其它模拟进化算法一样,通过候选解组成的群体在进化过程来寻求最优解。蚁群算法具有如下优点:较强的鲁棒性、正反馈机制、分布式计算、易于与其它方法给合、全局优化。

众多研究已经证明蚁群算法具有很强的发现较好解的能力,这是因为该算法不仅利用了正反馈原理,在一定程度上可以加快进化过程,而且是一种本质并行的算法,不同个体之间不断进行信息交流和传递,从而能够相互协作,有利于发现较好解。本文探讨利用蚁群算法来求解一般函数化。

2 基本蚁群算法系统模型

蚁群算法的思想最先用于求解旅行商问题(TSP),下面就以 TSP 问题为例来说明该算法思想。设 m 为蚁群数量;d_{ij} 为城市 i、j 之间的距离;$\tau(t)$为 t 时刻连接城市 i 和 j 的路径(i,j)上的残留信息量,初始时刻各路径上信息量相等,设 $\tau(0)=C$(C 为常数);η 表示城市 i 转移到城市 j 的期望程度,可根据某种

基金项目:甘肃省自然科学基金(the Natural Science Foundation of Gansu Province of China under Grant No.3ZS024-B25-034)。
作者简介:杜呈欣(1980-),男,研究方向为进化计算;陈小强(1966-),副教授,研究方向为计算电磁学和并行计算;熊伟清(1966-),男,副教授,研究方向为进化计算、软件工程。

启发式算法具体确定,在 TSP 问题中一般取 $\eta_{ij}=1/d_{ij}$。

蚂蚁 $k(k=1,2,\cdots,m)$根据各条路径上的信息量决定转移方向,t 时刻蚂蚁 k 从城市 i 向城市 j 转移的概率 $P_{ij}^k(t)$计算如下:

$$P_{ij}^k=\begin{cases}\dfrac{\tau_{ij}^\alpha(t)\cdot\eta_{ij}^\beta(t)}{\sum\limits_{s\in allowed_k}\tau_{is}^\alpha(t)\cdot\eta_{is}^\beta(t)} & j\in allowed_k\\ 0 & 否则\end{cases}\tag{1}$$

其中,$allowed_k=\{0,1,\cdots,n-1\}-tabu_k$ 表示蚂蚁 k 下一步允许选择的城市。与自然蚁群系统不同之处在于人工蚁群系统具有一定的记忆力,$tabu_k(k=1,2,\cdots,m)$用于记录蚂蚁 k 所走过的城市,集合 $tabu_k$ 随着进化过程进行动态调整。人工蚁群保留了自然蚁群信息素挥发特点,随着时间的推移,以前留下的信息逐渐消逝,参数 $\rho(0\leqslant\rho<1)$表示信息素的持久性,$1-\rho$ 则表示信息素的衰减度。在每只蚂蚁完成对所有 n 个城市的访问后(即一次循环结束),各路经商的信息素量根据公式(2)(3)进行调整:

$$\tau_{ij}(t+n)=\rho\cdot\tau_{ij}(t)+(1-\rho)\cdot\Delta\tau_{ij}\tag{2}$$

$$\Delta\tau_{ij}=\sum_{k=1}^m\Delta\tau_{ij}^k\tag{3}$$

其中,$\Delta\tau_{ij}^k$ 表示第 k 只蚂蚁在本次循环中留在路径(i,j)上的信息素量,$\Delta\tau_{ij}$ 表示本次循环中路径(i,j)上的信息素增量。

$$\Delta\tau_{ij}^k=\begin{cases}\dfrac{Q}{L_k} & 若第 k 只蚂蚁在 t+1 时刻经过路径(i,j)\\ 0 & 否则\end{cases}\tag{4}$$

其中,Q 是一个常数,表示蚂蚁所留的信息素量,L_k 表示第 k 只蚂蚁在本次循环中所走路径的长度。在初始时刻,$\tau_{ij}(0)=C$,$\Delta\tau_{ij}=0(i,j=0,1,\cdots,n-1)$。$\eta_{ij}$ 表示由城市 i 转到 j 的期望程度,可根据某种启发算法具体确定。根据其具体问题选择不同的启发算法,$\tau_{ij}(t)$、$\Delta\tau_{ij}(t)$及 P_{ij}^k 的表示形式各不相同。M.Dorigo 定义了三种不同的模型:ant-cycle system、ant-quantity system、ant-density system,它们的差别在表达式(4)的不同[4]。

ant-quantity system 模型中:

$$\Delta\tau_{ij}^k=\begin{cases}\dfrac{Q}{d_{ij}} & 若第 k 只蚂蚁在时刻 t 和 t+1 经过路径(i,j)\\ 0 & 否则\end{cases}\tag{5}$$

ant-density system 模型中:

$$\Delta\tau_{ij}^k=\begin{cases}Q & 若第 k 只蚂蚁在时刻 t 和 t+1 经过路径(i,j)\\ 0 & 否则\end{cases}\tag{6}$$

由于 ant-quantity system、ant-density system 利用的是局部信息,ant-cycle system 利用的则是整体信息,因此在求解 TSP 时 ant-cycle system 性能较好,因此通常采用它为基本模型。

下面是蚁群算法伪码:

```
Begin
    初始化:
    nc=0(nc 为迭代步数)
    将 m 个蚂蚁随机置于初始位置;
Loop:将所有蚂蚁的初始出发点置于当前解集中;
    for i=0 to n-1 do
      for k=1 to m do
        按概率选择顶点 j;
        移动蚂蚁 k 至顶点 j;
        将顶点 j 置于蚂蚁 k 的当前解集
      end for
    end for
    计算各蚂蚁的目标函数值;
    更新当前的最优解;
    重置所有工作变量;
    更新信息素;
    nc++;
    if nc<NcMax(预定的迭代次数) goto Loop;
    输出全局最优解;
End
```

算法时间复杂度为 $O(nc\cdot m\cdot n^2)$。

算法中的参数选择目前尚无理论依据,参数 Q、α、β、ρ 可以依据试验确定其最优组合[5,6]。经验结果为:(1)$1\leqslant\alpha\leqslant5$;(2)$1\leqslant\beta\leqslant5$;(3)$0.5\leqslant\rho\leqslant0.99$;(4)$1\leqslant Q\leqslant10\,000$。

基本的蚁群算法也存在一些缺陷:初期信息素匮乏,求解速度慢[7],需要较长的搜索时间,当目标规模较大时容易陷入局部最优解,即产生过早收敛。搜索过程中每个蚂蚁的运动是随机的,在搜索的初级阶段,各路经上信息素量相差不明显,通过信息反馈,使得较好路径上的信息素量逐渐增大,当问题规模较大时,很难在短的时间内从大量可能的路径中找出一条较好的路径。因此,对基本蚁群算法的改进主要是为了缩短运行时间和避免产生过早收敛。

3 用于函数优化问题的蚁群算法设计

用蚁群算法求解函数优化问题时,通过对目标函数的自适应来调整蚂蚁的搜索行为,同时通过路径选择过程中的多样性来保证得到更多的搜索空间,从而快速的得到全局最优解。经过多个经典测试函数的检验,用蚁群算法求解此类问题能够得到满意的实算结果。

函数优化问题一般为求极值问题,其中包括极大或极小值,算法设计中通常转化为求极小值问题,并通过改进目标函数使得所有解空间上解的目标函数值为正。算法的数学模型为:$\min f(x_i)$;$x_i\in[x_{\max},x_{\min}]$,其中 $i=1,2,3,\cdots$。

3.1 编码

受遗传算法(Genetic Algorithms)求解函数优化编码方式[8]的启发,采用二进制编码。根据计算要求的精度选择适当的位数,对解空间$\{x_1,x_2,\cdots,x_i,\cdots\}$上每个变量 x_i 用 N 位的二进制码$\{b_{N-1}b_{N-2}\cdots b_1b_0\}$表示,其中 $b_j\in\{0,1\}$,$j=1,2,\cdots,N$,b_{N-1} 为最高位,b_0 为最低位。若 N 位二进制码对应十进制数为 d,则 $x_i\in[x_{\max},x_{\min}]$与每个二进制数的对应关系为:

$$x_i=\frac{x_{\max}-x_{\min}}{2^N-1}*d+x_{\min}\tag{7}$$

3.2 定义有向图 G

这一步要解决的问题是如何将函数优化中的极值问题转化为求解 TSP 问题中的最短路径问题。文献[9]给出一种定义有向图的方式,定义有向图 $G=(C,L)$,其中顶点集合 C 为:$\{c_0(v_s),c_1(v_n^0),c_1(v_n^1),c_3(v_{n-1}^0),c_4(v_{n-1}^1),\cdots,c_{2n-3}(v_2^0),c_{2n-2}(v_2^1),c_{12n-1}(v_1^0),c_{2n}(v_1^1)\}$,$v_s$ 为起始顶点,顶点 v_j^0 和 v_j^1 分别用于表示二进制编码中第 j 位 b_j 取值为 0 和 1 的状态。考虑到这种方法不能很好的利用矩阵空间,降低了算法的执行效率,我们提出一种新的搜索矩阵表达方式,在相同编码位数的情况下,矩阵

规模减小一半。这样,充分利用存储空间,提高了算法的执行效率。

定义有向图 $G=(C,L)$,其中顶点集合 C 为:$\{(c_0,c_0),(c_0,c_1),(c_0,c_2),\cdots,(c_0,c_N),(c_1,c_0),(c_1,c_1),(c_1,c_2),\cdots,(c_1,c_N),\cdots,(c_N,c_0),(c_N,c_1),(c_N,c_2),\cdots,(c_N,c_N)\}$,$(c_0,c_0)$为启始定点(蚂蚁地出发点),顶点$(c_i,c_j)(i,j=1,2,\cdots,N)$,$c_i$和 c_j的大小关系确定二进制位取 0 后 1 的状态;有向弧集合 L 为{从(c_0,c_0)出发,到达(c_0,c_1)或(c_1,c_0);若第 s 步在(c_i,c_j),下一步能达的顶点为(c_i,c_s)或(c_s,c_j)}。

用连接矩阵表示有向图,N 位编码的连接矩阵为 $(N+1)*(N+1)$,从(0,0)出发,到达(0,1)表第一位取 0,到达(1,0)表第一位取 1;若第 s 步在(i,j),下一步可能到达为到(i,s)或(s,j),每一位若 $i<j$ 则取 0,反之取 1。下面以 $N=8$ 举例说明,(10110010)的连接矩阵如图 1 所示。

	C0	C1	C2	C3	C4	C5	C6	C7	C8
C0	1	0	0	0	0	0	0	0	0
C1	1	0	1	0	0	0	0	0	0
C2	0	0	0	0	0	0	0	0	0
C3	0	0	1	0	0	0	0	0	0
C4	0	0	1	0	0	1	1	0	0
C5	0	0	0	0	0	0	0	0	0
C6	0	0	0	0	0	0	0	0	0
C7	0	0	0	0	0	0	1	0	1
C8	0	0	0	0	0	0	0	0	0

图 1　10110010 的连接矩阵

3.3　信息素

通过将函数优化向有向图转化,求解优化问题的蚁群算法中的信息素表示有所变化,τ_{ij} 值不再是路径上的,而是每个节点上的。在信息素更新时,仅对最佳路径上的信息素量进行更新,正确的指导蚂蚁下一次搜索,消除了在计算信息素更新时非最佳路径对蚂蚁的搜索的影响,从而避免了大量无效搜索,明显提高搜索效率。

4　测试函数的应用实例

(1)求解函数 $f(x)=5e^{-0.5x}\sin 30x+e^{0.2x}\sin 20x+6$,$x\in[0,8]$的最小值

该函数为单变量函数(如图 3 所示),具有多个局部最小值,对优化变量的取值是十分敏感。蚁群算法参数设置为 $Q=100$、$\alpha=3$、$\beta=2$、$\rho=0.9$,编码长度(N)为 20,蚁群规模(M)为 50,搜索迭代次数($NcMax$)为 30。经过 9 步迭代后得到最优解:函数在极值点(二进制编码为 11011100101001001000)处取得最小值为 1.257 363。

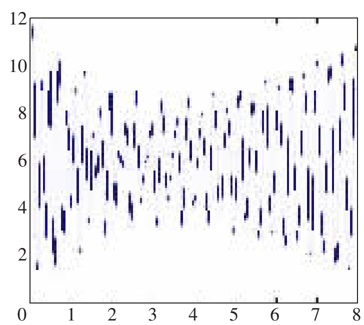

图 2　单变量函数图

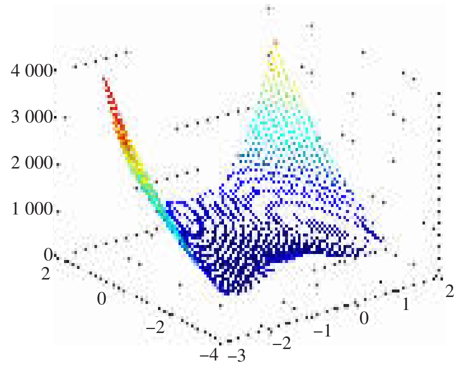

图 3　二变量二次函数图

(2)求解函数 $f(x)=100(x_1^2-x_2)^2+(1-x_1)^2$,$x_1$、$x_2\in[-2.048,2.048]$的最大值

该函数是局部有些凹陷的二变量二次函数(如图 2 所示),蚁群算法参数设置为 $Q=100$、$\alpha=3$、$\beta=2$、$\rho=0.9$,编码长度(N)为 40,蚁群规模(M)为 50,搜索迭代次数($NcMax$)为 60。运行结果如表 1 所示。

表 1　运行结果

运行次数	得到最优目标值	目标函数平均值	实际最优值	得到实际最优值的次数和比率	
600	3 905.926 227	3 874.024 634	3 905.926 227	299 次	49.833%

(3)求解六峰驼返回函数 $f(x)=(4-2.1x_1^2+\frac{1}{3}x_1^4)x_1^2+x_1x_2+(-4+4x_2^2)x_2^2$,其中 $x_1\in(-3,3)$,$x_2\in(-2,2)$的最小值

该函数为三变量、单峰、二次函数,该函数在两个不同点:$(x_1,x_2)=(-0.089\,8,0.712\,6)$和$(0.089\,8,-0.712\,6)$处有一个全局最小值-1.031 6。蚁群算法参数设置为 $Q=100$、$\alpha=3$、$\beta=2$、$\rho=0.9$,编码长度(N)为 40,蚁群规模(M)为 50,搜索迭代次数($NcMax$)为 60。运行结果如表 2 所示。

表 2　运行结果

运行次数	得到最优目标值	目标函数平均值	实际最优值	得到实际最优值的次数和比率	
600	−1.029 754	−0.591 569	−1.031 6	0 次	0%

以上选取了三个常用的不同类型的测试函数作为实算例子,试验结果表明用蚁群算法求解函数优化问题具有较好的性能。

5　讨论

通过对三组实验结果的比较可以发现:求解 4(1)时,算法在很短时间内(9 次迭代)就能得到最优解;相反,在求解 4(3)时,受编码方式、参数设置、计算精度等因素的影响总是接近但得不到理论上的最优解。下一步的工作以一种新的编码方式,通过更加合理的参数设置使算法能够在较高执行效率的情况下得到高质量的解。可以肯定,作为一种高效并行的全局搜索方法,蚁群算法以其特有的算法特点使其在许多实际问题中的应用会越来越广,同时,强大的计算机模拟工具的出现,必将使蚁群算法的研究取得长足的进展。(收稿日期:2006 年 12 月)

参考文献:

[1] Dorigo M,Gambardella,Maria L.Ant colonies for the traveling salesman problem[J].Biosystems,1997,43(2):73-81.

(下转 158 页)