

动态跃迁转移蚁群算法

胡 勇

(南京师范大学计算机系, 南京 210097)

摘 要: 给出了一种改进的蚁群算法, 该算法对蚂蚁初始位置选择上进行优化, 能较大地提高进化速度, 并且还通过动态地调整跃迁转移概率, 减少了停滞, 加快了收敛速度。实验表明对于某些TSP问题, 实验结果优于国外最新的成果。

关键词: 模拟进化算法; 蚁群算法; 旅行商问题; 动态跃迁转移

An Ant Colony Algorithm Based on Dynamic Transition

HU Yong

(Department of Computer Science, Nanjing Normal University, Nanjing 210097)

[Abstract] This paper proposes a dynamic transition ant colony algorithm. The experiments demonstrate that the evolution can be quickened and the computational precision can be effectively improved through adopting the dynamic transition method. Some experiments result on the dynamic transition algorithm are better than some latest algorithm.

[Key words] Simulated evolutionary algorithm; Ant colony algorithm; TSP; Dynamic transition

人们从生物的特性中受到启发, 提出了许多用以解决复杂优化问题的新方法, 并成功地应用于实际问题。特别发现蚂蚁群落在进行合作时基本上是自组织的, 蚁群算法(ant colony algorithm, ACA)就是在这一基础研究上由意大利学者M. Dorigo, V. Maniezzo, A. Colomi等人首先提出的一种新型的模拟进化算法^[1,2], 该算法是一种利用计算机模拟蚁群活动的启发式搜索算法, 可对组合优化问题进行求解。用该算法求解TSP问题^[1]、分配问题^[3]、job-shop调度问题^[3]取得了一系列较好的实验结果。因此, 蚁群算法逐渐引起了一些研究者的关注, 并用该算法来解决实际问题, 如求解连续空间优化问题^[4]。尽管此算法被成功地应用解决多种组合优化问题, 但该算法也存在进化速度较慢、进化过程容易停滞的缺陷。例如一些研究者提出的杂交算法^[6]、最大最小算法^[7]等, 尽管这些算法比原先的算法有了改进, 但杂交算法采用无方向随机杂交, 杂交效率低, 提高速度不明显; 最大最小算法只是通过对信息素的大小进行限制, 算法停滞的可能性仍然较大。本文从另一个角度出发, 采用动态跃迁转移的方法, 对蚂蚁选择城市(也是选择路径)进行了优化, 使得算法能减少停滞, 加快收敛速度。

1 基本蚁群算法的原理

基本蚁群算法通过模拟生物界中的蚂蚁搜索食物的过程, 也就是通过人工蚂蚁之间的信息交流与相互协作最终找到从蚁穴到食物源的最短路径, 这种算法被称为“人工蚁群算法”。为了说明人工蚁群算法的原理, 先简要介绍一下蚂蚁搜索食物的具体过程:

蚂蚁之间通过一种叫“外激素”的物质进行交流, 正如M. Dorigo等人在文献[1]中指出的: 蚁群算法的最大的特点是蚁群中的蚂蚁采用“外激素”为媒介的间接的异步的联系方式。蚂蚁在行动(寻找食物或者寻找回巢的路径)中, 会在它们经过的地方留下“外激素”。这些物质能被同一蚁群中后来的蚂蚁感受到, 并作为一种信号影响后到者的行动(具体表现在后到的蚂蚁选择有这些物质的路径的可能性, 比选择没有这些物质的路径的可能性大得多, 选择较短的路

径的可能性比选择较长的路径的可能性也大得多), 而后到者留下的外激素会对原有的外激素进行加强, 并如此循环下去, 最后几乎所有的蚂蚁集中到一条最短或接近最短的路径上。因此, 由大量蚂蚁组成的蚁群的集体行为便表现出一种信息正反馈现象: 某一路径上走过的蚂蚁越多, 则后来者选择该路径的概率就越大。蚂蚁个体之间就是通过这种信息的交流达到搜索食物的目的。

2 基本蚁群系统模型及其实现

为了便于理解, 以求解平面上 n 个城市的TSP问题($0, 1, \dots, n-1$ 表示城市序号)为例说明蚁群系统模型。 n 个城市的TSP问题就是寻找通过 n 个城市各一次且最后回到出发点的最短路径。之所以选择TSP问题, 一方面因为蚁群算法最初用于求解TSP问题, 便于比较, 另一方面, TSP是典型的组合优化难题, 常常用来验证某一算法的有效性。对于其它问题, 可以对此模型稍作修改便可应用^[6]。虽然它们从形式上看略有不同, 但基本原理是相同的, 都是通过模拟蚁群行为到达优化之目的。

为模拟实际蚂蚁的行为, 首先引进如下记号: 设 m 是蚁群中蚂蚁的数量, $d_{ij}(i, j=1, 2, \dots, n)$ 表示城市 i 和城市 j 之间的

距离, $b_i(t)$ 表示 t 时刻位于城市 i 的蚂蚁的个数, $\sum_{j=1}^n b_i(t), \tau_{ij}(t)$

表示 t 时刻在 ij 连线上残留的信息量。初始时刻, 各条路径上信息量相等, 设 $\tau_{ij}(0) = C$ (C 为常数)。蚂蚁 $k(k=1, 2, \dots, m)$ 在运动过程中, 根据各条路径上的信息量决定转移方向, $p_{ij}^k(t)$ 表示在 t 时刻蚂蚁 k 由位置 i 转移到位置 j 的概率。

$$p_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}^a(t) \eta_{ij}^b(t)}{\sum_{s \in allowed_k} \tau_{is}^a(t) \eta_{is}^b(t)} & j \in allowed_k \\ 0, & otherwise \end{cases} \quad (1)$$

作者简介: 胡 勇 (1973—), 男, 讲师, 主研方向为智能控制

定稿日期: 2003-11-14

E-mail: huyong@njnu.edu.cn

其中, $\text{allowed } k = \{0, 1, \dots, n-1\} - \text{tabu}_k$ 表示蚂蚁 k 下一步允许选择的城市, 与实际蚁群不同, 人工蚁群系统具有记忆功能, $\text{tabu}_k (k=1, 2, \dots, m)$ 用以记录蚂蚁 k 当前所走过的城市, 集合 tabu_k 随着进化过程作动态调整。 α, β 分别表示蚂蚁在运动过程中所积累的信息及启发式因子, 它们在蚂蚁选择路径中所起的不同作用。 η_{ij} 表示由城市 i 转移到城市 j 期望程度, 对于 TSP 问题, 一般可设为城市 i 转移到城市 j 的距离的倒数。

随着时间的推移, 以前留下的信息逐渐消逝, 用参数 $1-\rho$ 表示信息消逝程度, 经过 n 个时刻, 蚂蚁完成一次循环, 各路径上信息量要根据下式作调整

$$\tau_{ij}(t+n) = \rho \cdot \tau_{ij}(t) + \Delta \tau_{ij} \quad \rho \in (0, 1) \quad (2)$$

$$\Delta \tau_{ij} = \sum \Delta \tau_{ij}^k \quad (3)$$

$\Delta \tau_{ij}^k$ 表示第 k 只蚂蚁在本次循环中留在路径 ij 上的信息量, $\Delta \tau_{ij}^k$ 也表示本次循环中路径 ij 上的信息量的增量。

$$\Delta \tau_{ij}^k = \begin{cases} \frac{Q}{L_i^k}, & \text{若第 } k \text{ 只蚂蚁经过 } ij \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

其中, Q 是常数, L_i^k 表示第 k 只蚂蚁在本次循环中所走路程的长度。在初始时刻, $\tau_{ij}^k(0) = C$ (常量), $\Delta \tau_{ij}^k = 0 (i, j=0, 1, \dots, n-1)$ 。根据具体算法的不同, $\tau_{ij}^k(t)$, $\Delta \tau_{ij}^k(t)$ 及 $P_{ij}^k(t)$ 的表达形式可以不同, 要根据具体问题而定。M. Dorigo 曾给出 3 种不同模型, 分别称之为 ant-cycle system, ant-quantity system, ant-density system^[7]。它们的差别在于表达式(4)的不同。在 ant-quantity system 模型中,

$$\Delta \tau_{ij}^k = \begin{cases} \frac{Q}{d_{ij}}, & \text{若第 } k \text{ 只蚂蚁在 } t \text{ 和 } t+1 \text{ 间经过 } ij \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

在 ant-density system 模型中,

$$\Delta \tau_{ij}^k = \begin{cases} Q, & \text{若第 } k \text{ 只蚂蚁在 } t \text{ 和 } t+1 \text{ 间经过 } ij \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

它们的区别在于后两种模型中利用的是局部信息, 而前者利用的是整体信息。

3 改进的蚁群算法

经过仿真实验发现, 随着仿真蚂蚁经过一段时间的运行, 路径上的信息素会逐步集中于较短的路径上, 而这很可能是局部的最短路径, 也就是算法可能已陷入局部的最优解而停滞, 不能得到全局的最优解。因此, 为了能减少停滞, 可以动态的增加搜索的多样性, 也就是随着信息素残留增加, 蚂蚁局部选择可能性加大, 而通过增加搜索的多样性, 可以减少停滞。具体方法是, 在采用局部和全局混合模型的基础上, 根据停滞的时间, 动态调整 q_0 值, 即停滞的时间越

长, q_0 的值会被动态地减少, 停滞的时间越短, q_0 的值会被动态地加大的办法来优化求解性能。对于算法中的其它的参数如 C, α, β, ρ , 可以用实验方法确定其最优组合, 停止条件可以用固定进化代数或者当进化趋势不明显时便停止计算。另外在蚂蚁的起始城市的选择上, 将蚂蚁分配到较偏远的城市里, 减少蚂蚁较长路径的反复, 实验证明通过采

用以上策略, 可以有效地提高优化的速度, 降低停滞的可能性, 从而提高优化的效果。下面就是该改进的蚁群算法:

```
begin
  初始化过程:
  nc := 0;
  tao[i][j] := C;
  tabu[k] = Φ;
  for(nc=0; nc<Ncmax; nc++)
  { 将一群蚂蚁随机地放在较偏远的周边的 ambientCityNum 个城市里, 并设置它们的 tabu[k][0];
    for(index=0; index<CityNum; index++) /* index 表示已走城市的个数 */
    {
      for(k=0; k<M; k++)
      {
        q=rand()*1.0/RAND_MAX; /* 产生 0~1 的随机数 q */
        if(q<q0)
        {
          选择城市 j, 使得 tao[s][i]/e[s][i] 最大;
          j ∈ {0, 1, ..., n-1} tabu[k] /* 当前城市为 s */
          else
            根据公式 (1) 的概率选择城市 j;
        }
        把所选择的的城市序号加到 tabu[k] 中 /* 动态调整集合 tabu[k] */;
        利用公式 (5) 调整 tao[i][j]; /* 局部调整 */
      }
      确定本次循环中找到的最佳路径;
      利用公式 (4) 调整 tao[i][j]; /* 全局调整 */
      if(finCounter>(finCounterLine+1)*STOPLINE) /* 如果停滞代数越多, 动态调整 q0 值 */
      {
        long pAttenuation;
        finCounterLine=(int)(finCounter/STOPLINE);
        pAttenuation=finCounter;
        pAttenuation=pAttenuation/STOPLINE;
        q0=attenuation[pAttenuation];
      }
    }
  }
  输出最佳路径及最佳结果
end
```

4 实验结果

分别以 eil51 的 TSP 问题为例, 一方面选择国际上最新的蚁群算法, 另一方面用本文的改进算法实验, 并将结果比较如下 (表 1):

表 1 以 51 个城市的 eil51 TSP 问题两种算法实验结果

算 法	进化 100 代时的路径长度	最短路径长度	收敛进化的代数	说明
ACS	585.213	433.411	1802	文献[5]
ACS+NN	510.315	429.452	1224	文献[5]
EA+NN	482.356	430.321	1032	文献[5]
本文算法	428.127	416.525	957	

根据上面的实验可知本文的算法在解决 TSP 问题时, 收敛效果要好于其它的算法, 特别是运行初期的收敛速度非常快, 最后能得到较好的解。

5 结论

本文提出改进的蚁群算法, 可以有效地减少基本蚁群算法中容易停滞的缺陷, 同时加快了收敛速度, 有利于实际运

(下转第 171 页)

表4 标准PSO, GA和HPSO对不同维数的

Griewank 函数50次优化试验结果

Func 3		Griewank			
维数	周期	适应度	PSO ($\times 10^{-1}$)	GA	HPSO ($\times 10^{-1}$)
2	200	均值	3.715	1.095	4.807
		方差	3.853	0.554	3.619
10	1 000	均值	64.611	1.472	60.875
		方差	31.015	0.431	33.689
20	2 000	均值	27.332	2.256	34.151
		方差	23.147	1.952	27.300
30	3 000	均值	5.475	2.761	3.090 5
		方差	9.035	1.932 6	3.338 5

表5 标准PSO, GA和HPSO对不同维数的

Rastrigin 函数50次优化试验结果

Func 4		Rastrigin			
维数	周期	适应度	PSO	GA	HPSO
2	200	均值	0.084 5	0.167	0
		方差	0.067 1	0.183	0
10	1 000	均值	3.861	6.719	1.486×10^{-8}
		方差	1.118	1.189	3.389×10^{-8}
20	2 000	均值	16.981	18.83	1.664
		方差	4.464	8.660	1.179
30	3 000	均值	40.019	37.83	11.297

4 试验结果与分析

在对多模函数Rastrigin进行优化时,随着维数的增高,大量的局部极值点的引入,使PSO的性能逐渐下降。Griewank 函数相当于 Sphere 函数加入了噪音项 $\prod \cos((x_i - 1)/\sqrt{i})$,随着维数的增加,该项趋于0,适应度函数空间趋于平滑,因此在维数较低和维数较高时PSO都能同HPSO一样有很好的优化性能。显示出PSO在搜索空间中局部极值点较少的情况下良好的局部寻优能力。而HPSO在各种情况都显示出很强的优化能力。

图4、图5是HPSO和PSO对50维Rosenbrock函数和50维Rastrigin函数进行优化的适应度和多样性的对比。以HPSO达到优化要求作为终止(即输出适应度值和目标值的误差小于 10^{-3})。如图所示,随着多样性的降低,PSO很快陷入早熟收敛。而HPSO的适应度在不断调整,使种群的在收缩与扩张之间保持协调,保证了整个优化过程持续收敛。

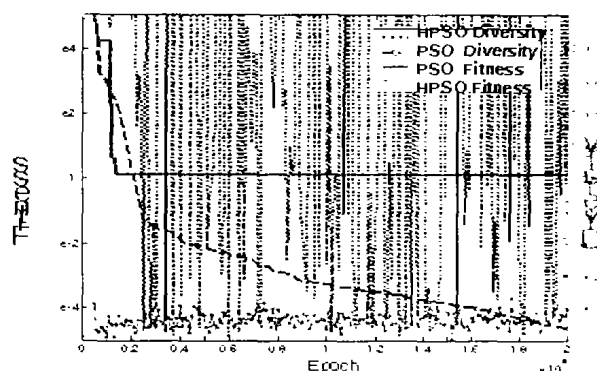


图4 HPSO, PSO对50维Rosenbrock函数优化适应度和多样性

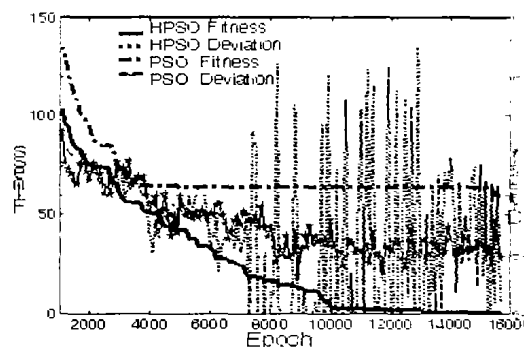


图5 HPSO, PSO对50维Rastrigin函数优化适应度和多样性

5 结论

本文通过对标准PSO算法的信息传递结构以及种群多样性变化进行分析,提出了HPSO算法对标准PSO算法进行改进,该算法引入了交叉算子和多样性控制因子。仿真试验表明, HPSO在群体信息传递结构和对种群多样性协调的控制方面有了改进,在单模和多模函数优化问题,以及在高维的情况下,都显示出良好的性能。

参考文献

- 1 Kennedy J, Eberhart R C. Particle Swarm Optimization. Proc. IEEE International Conference on Neural Networks, IEEE Service Center, 1995, VI: 1942-1948
- 2 Van den Bergh F. An Analysis of Particle Swarm Optimizers, [Phd thesis], 2001-11
- 3 Kennedy J. Small-Worlds and Mega-Minds: Effects of Neighborhood Topology on Particle Swarm Performance. Proc. of the 1999 Congress of Evolutionary Computation, 1999, 3: 1931-1938
- 4 Riget J, Vesterstroem J S. A Diversity Guided Particle Swarm Optimizer - the ARPSO. Technical Report No. 2002-02. University of Aarhus, EVALife, 2002

- 2 Colomi A, Dorigo M, Maniezzo V. An Investigation of Some Properties of an Ant Algorithm. Proc PPSN' 92, 1992: 509-520
- 3 Colomi A, Dorigo M, Maniezzo V, et al. Ant System for Job-shop Scheduling. Belgian Journal of Operations Research and Statistic Computing Science, 1994, 34(1): 39-53
- 4 Cheng-Fa T, Chun-wei T. A New Approach for Solving Large Traveling Salesman Problem Using Evolutionary Ant Rules. IEEE, 2002
- 5 Stutzl T, Hoos H. The MAX-MIN Ant System and Local Search for the Traveling Salesman Problem. In Proc. ICEC97/1997 IEEE 4th Int. Conf. Evolutionary Computation, 1997: 309-314

(上接第168页)

用。但是蚁群算法研究的时间还不长,多数成果都只是基于大量实验的数据分析,蚁群算法的数学理论基础还不坚实,其中的各种参数选择比较复杂,所以算法的理论问题还有很多问题有待进一步研究。

参考文献

- 1 Colomi A, Dorigo M, Maniezzo V. Distributed Optimization by Ant Colonies. Proc 1st European Conf Artificial Life. Paris, France: Elsevier, 1991: 134-142