

一种自适应信息素改进蚁群算法^①

Adaptive Ant Colony Algorithm Based on Pheromone

周燕霞 (浙江东方职业技术学院 工程技术系 浙江 温州 325011)

摘 要: 针对基本蚁群算法容易导致早熟、停滞现象, 改变其信息素释放策略, 提出自适应信息素改进蚁群算法, 使得信息素的更新采用了一种新的面向数据的传递方式。改进后的算法在收敛速度和解的多样性之间取得良好的平衡。通过解 TSP 实例试验结果表明, 其比基本蚁群算法具有更好的搜索能力, 且其误差率可以保持较低的水平。

关键词: 蚁群算法 信息素释放 信息素更新

1 引言

蚁群算法^[1]本质上是一种随机搜索算法^[2], 以其高效的性能在近年来倍受关注。蚁群算法通过许多蚂蚁共同完成寻求最优解的任务。蚂蚁个体在候选解的空间中独立地搜索解, 并在所寻得的解上留下一一定量的信息素, 蚂蚁个体之间通过这些信息素进行信息传递和相互协作, 从而完成复杂任务。

在蚁群算法的最初阶段, 所有解上的信息素都是相同的。随着算法的推进, 蚂蚁所寻解的性能越好, 则留在该解上的信息素就越多^[3]。在一定范围内, 蚂蚁能够察觉到这种信息素并指导它的行为, 使得信息素较多的解被选择的可能性增大, 导致该解上的信息素强度进一步增强, 由此构成了一个学习信息的正反馈过程, 使得算法渐渐趋于收敛, 并逐渐逼近最优解。

在上述蚁群寻解过程中, 蚂蚁个体总是依赖于其他蚂蚁的反馈信息来强化学习, 而不去考虑自身经验的积累^[4]。这样的盲从行为, 很容易导致早熟、停滞现象。基于此, 本文改变传统蚁群算法的信息素释放策略^[5], 提出了一种自适应^[6]信息素改进蚁群算法。

2 信息素释放规则改进

本文提出的信息素释放规则总体思路为: 当蚂蚁个体进行路径搜索时, 如果找到一段很短的子路径, 它就释放出一定量的信息素。这些信息素一方面会被直接传递给位于该子路径上的蚂蚁, 另一方面将以该

路径为中心向外传递, 从而影响该路径附近其它蚂蚁的行为, 使它们在寻找路径时以较大的概率在下一步选择此路径。

下面以一个实例来说明本文提出的方法。如图 1 所示, 假设蚂蚁欲从节点 a 到节点 f, 其中路径(a, b)为一距离很短的子路径。假设路径(a, b)和(a, c)上的信息素的浓度相同, 则由于路径(a, b)比路径(a, c)短, 蚂蚁会以更大的概率选择节点 b 为下一步要到达的位置。

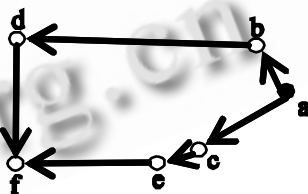


图 1 算法原理示意

如果在此之前, 另一只蚂蚁刚好从节点 c 走到节点 e, 其将产生相应浓度的信息素, 并向周围进行传递。由于距离远近的不同, 路径(a, c)的信息素的浓度将大于路径(a, b)上的信息素的浓度, 当大到一定程度时, 蚂蚁将以更大的概率选择节点 c 为下一步要到达的位置, 从而使蚂蚁选择路径 a—c—e—f 为最短路径。

蚂蚁 N 刚走过的两个节点 i 和 j 之间的距离为 D_{ij}^N 。该蚂蚁将分别以节点 i 和节点 j 为中心向周围传

① 收稿时间:2009-02-19

递信息素,其中,节点*i*和节点*j*的信息素值都设为 Q_0 。对于其他的任一节点*k*,如果它位于蚂蚁*N*所产生的信息素的传递范围之内,则设由蚂蚁*N*所产生的传递到该节点的信息素浓度为 P_i 和 P_j 。针对求解问题的数据特点,对 P_i 和 P_j 进行处理,设处理后的信息素浓度为 E_i 和 E_j ,应用 E_i 和 E_j 去更新路径(*k*, *i*)和(*k*, *j*)上的信息素值,这样就得到了第*N*只蚂蚁在一次循环中留在各有关路径上的信息素值的计算公式,分别如公式(1)、(2)、(3)所示。

$$\Delta\lambda_{ij}^N = (Q + R) / D_{ij}^N \quad (1)$$

$$\Delta\lambda_{ki}^N = D_{ki}^N + R / E_i \quad (2)$$

$$\Delta\lambda_{kj}^N = D_{kj}^N + R / E_j \quad (3)$$

其中 R 为具体的数据对象参数。从以上公式可看出,该算法模型可根据其处理数据的对象改变多条路径上的信息素的浓度。

3 自适应信息素改进蚁群算法

3.1 改进后的算法设计

基于以上分析,本文提出一种基于信息素的改进蚁群算法。该算法对基本蚁群算法的路径选择规则进行改进,使信息素更新采用一种新的面向数据的传递方式。

假设蚂蚁总数为 m ,在下一时刻,按照公式(4)进行信息素的更新:

$$\tau_{ij}(T+1) = \begin{cases} \tau_{ij}(T) - nc / D_{ij}, & \text{若有大于 } m / nc \text{ 只蚂蚁选择同一路径 (nc为小常量)} \\ \tau_{ij}(T) + nc / D_{ij}, & \text{若有小于 } m / nc \text{ 只蚂蚁选择同一路径} \end{cases} \quad (4)$$

由于蚂蚁常常选择信息素较大的路径,因此当多只蚂蚁选中同一路径后,由于信息素值增加的幅度太大,容易造成多只蚂蚁集中到该路径,所以仅取 nc / D_{ij} 为增加的信息量;当选择该路径的蚂蚁达到一定数量或多数蚂蚁选择该路径后,如果当前距离超过上一次的最优路径长度,则终止遍历,信息素增量取 $-nc / D_{ij}$,通过大幅度削减其信息素值使其趋于各条路径信息素值的平均值,从而使蚂蚁选择其他路径的可能性增加,让搜索得到的解趋于多样化。

本文算法按照公式(5)进行信息素的整体更新:

$$\tau_{ij}(T+1) = (1 - \rho) \cdot \tau_{ij}(T) + \sum_{l=1}^m \gamma_l \cdot \Delta\tau_{ij}^l(T) \quad (5)$$

其中 $\Delta\tau_{ij}^l(T)$ 按公式(6)确定:

$$\Delta\tau_{ij}^l(T) = \begin{cases} Q / L^l(T), & \text{第 } l \text{ 只蚂蚁经过路径 } (i, j) \\ 0, & \text{其他} \end{cases} \quad (6)$$

$L^l(t)$ 为本次迭代中第*l*只蚂蚁遍历的路径的全长, γ_l 为第*l*只蚂蚁所对应的解对该路径的信息素影响程度, γ_l 的计算方法如下:设经过路径(*i*, *j*)的蚂蚁总数为 k ,对它们在本次迭代中遍历的路径全长由小到大进行排序,所得的序号存放于数组 arr 中,即 $\text{arr}[l]$ 表示第*l*只蚂蚁对应的序号,则 γ_l 按公式(7)计算:

$$\gamma_l = (1 - \varepsilon_{ij}) \times k - \text{arr}[l] + 1 (\varepsilon_{ij} \text{ 为路径 } (i, j) \text{ 的路径权值}) \quad (7)$$

本文从以下方面来说明由上式确定的 ε_{ij} 的合理性。从由节点*i*出发的各条路径来看,当路径(*i*, *j*)的路径权值比较高时,说明该路径上的信息素值相对较高,此时 $(1 - \varepsilon_{ij}) \times k$ 的值就比较小。对于较少的几个较优解,由于它们的 $\text{arr}[l]$ 的值比较小,为正值,因而在该路径上将增加信息素值;而对较多的其他解,由于它们的 $\text{arr}[l]$ 值比较大,因而它们的码为负值,在该路径上将减少信息素值,这样会使路径(*i*, *j*)上的总体信息素值减少。反之,当路径(*i*, *j*)的权值比较低时,会使该路径上的总体信息素值增加。这样可以调节各路径上的信息素浓度,以防止解过于集中。

从各蚂蚁本次所经过的路径来看,若第*l*只蚂蚁遍历路径总长度比较小,即 $\text{arr}[l]$ 的值比较小,则将增加路径上的信息素值,而且 $\text{arr}[l]$ 越小,即长度越短,路径上信息素增加的程度越大,这将有效地强化短路径上的信息素;反之,若第*l*只蚂蚁遍历路径总长度比较大,则将减少相应路径上的信息素浓度,而且 $\text{arr}[l]$ 值越大,即长度越长,路径上信息素减少的程度就越大。这样可在防止信息素的过分集中的同时,保持较好路径上信息素值。

从节点*i*本身来看,当节点*i*上的路径比较多时,路径上的差异就会比较小,各个解对信息素的影响也就比较均匀。反之,当节点*i*上的路径比较少时,路径上的差异就会比较大,较优解对信息素的更新就会有较大的影响,从而使较优解路径上集中较高的信息素。该算法的这种自适应信息素更新机制可以动态调节信息素,在收敛速度和解的多样性之间取得良好的平衡。

假定所有蚂蚁的运动速度相同，对于基本蚁群算法，随着时间的推移，以前留在各条路径上的信息素逐渐消逝，如：用参数 w 表示信息素的挥发程度。经过 T 个时刻，蚂蚁完成一次循环，各路径上信息素的浓度要根据式(5)作整体调整。

为了使算法能有更好的自适应性，随着时间的推移，令在时刻 T 的信息素浓度的变化为：

$$\Delta\lambda_{ij}^T = \Delta\lambda_{ij}^{old} \times I + (Q + R) / D_{ij}$$
 (8)

其中， $\Delta\lambda_{ij}^{old}$ 是上一次经过路径(i, j)的信息素浓度的变化。使用该更新规则以后，时间 T 由蚂蚁爬行的距离与运动速度 V 确定： $T = S/V$ 。

3.2 信息素传递算法流程

信息素传递算法流程如下：

- ① 初始化：初始化蚁群，初始时，设迭代次数 $i=0$ ，蚂蚁总数为 m 。
- ② 将 m 只蚂蚁按某种初始规则分别置于 n 个节点上。实验中，可随机地放置蚂蚁。
- ③ 应用信息素更新规则改变信息素值。在构造解时，蚂蚁 N 走过的每条路径由公式(5)，(6)，(7)求得。
- ④ 若所有 m 只蚂蚁都构造完成，则转步骤⑤；否则转步骤③。
- ⑤ 若终止条件满足，则结束；否则 $i=i+1$ ，转步骤②进行下一属性群的计算。终止条件可设定最短路的长度下限，也可限定运行时间。

4 试验与分析

本文选用 TSP 库中的实例 rat99 和 rd100 问题对该算法进行实验分析，并与基本蚁群算法进行对比分析(理想状况下)。针对不同的 TSP 实例，基本蚁群算法(ACO)和自适应信息素改进蚁群算法(Ant Expand Galant，简称 AXG)的分析结果如表 1 所示。其中算法参数取值^[7]为： α (信息素浓度)=0.4； β (蚂蚁智能度)=0.6； ρ (信息素扩散系数)=0.5； Q (蚂蚁释放的信息素量)=4。设蚂蚁数目等于节点数，运行 20 次，每次迭代 500 次。

“平均时间”指一次运行中找到最好解的平均时间，“最优解”是指找到的最短路径的长度。从表 1 结果可以看出，自适应信息素改进算法比基本蚁群算法具有更好的搜索能力，且其误差率可以保持比较低的水平，因此总体来说其具有较强的搜索能力。

为了进一步检验该算法，本文选用 TSP 库中的实例 ch130、lin105、kroB150、berlin52、和 rat575 问题对该算法进行实验分析，针对不同的 TSP 实例，该算法的分析结果如表 2 所示。算法参数设置如下： $\rho=0.7$ ； $Q=2$ 。同样设蚂蚁数目等于节点数，运行 10 次，每次迭代 450 次。

表 1 两种算法实验效果分析

算法	问题	最优解	平均解	误差率	最优解次数	平均时间(s)
ACO	rat99	1228.45	1236.32	4.7%	420	145
	rd100	7923.76	7942.88	5.4%	412	432
AXG	rat99	1218.15	1220.87	2.1%	432	121
	rd100	7916.85	7928.41	2.3%	426	421

表 2 AXG 算法实验效果分析

问题	最优解	平均解	误差率	最优解次数	平均时间(s)
ch130	6123.19	6128.87	2.4%	389	113
lin105	14383.6514384.32	2.2%	362	176	
kroB150	26141.4326147.48	2.1%	387	265	
berlin52	7551.36	7556.38	2.2%	354	167
rat575	6780.02	6786.57	2.6%	346	159

表 2 结果进一步可以证明，本文算法具有较好的搜索能力，且其误差率可以保持比较低的水平。

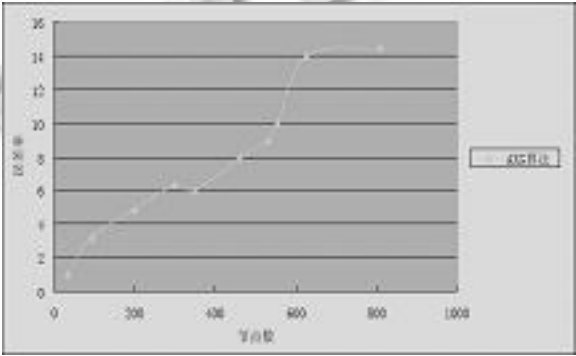


图 2 AXG 算法误差效果分析

如图 2 所示，在节点数量少的情况下(200 节点以内)，本文算法误差率保持很低的水平。随着节点数增加，算法的误差率相应有所上升。

从收敛速度^[8]来讲(如图 3 所示)，当节点数在 200 以内时，本文算法可以保持较好的收敛性。

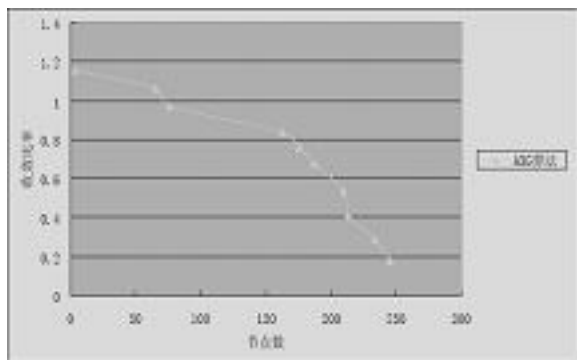


图3 AXG算法收敛效果分析

5 结语

在一般蚁群算法中,对所有蚂蚁所经过的路径增加其信息素,要么只增加最优适应度的路径上的信息素值,其余路径上的信息素值则被削减;或者基于固定等级的算法,让适应度相对较好的若干条路径根据其解的优劣程度决定信息素的增加程度,这些做法都采用了固定的信息素增减比例,而忽视了解的分布情况。

本文算法可根据所处理的数据量的变化自适应地更新信息素值,以动态地调整各路径上的信息素的分布,使之不至于过分集中或者分散,在加速收敛的同时避免早熟现象。

由于蚂蚁常常选择信息素较大的路径,当多只蚂蚁选中同一路径后,信息素增加的幅度太大就容易使多只蚂蚁集中到该路径,所以本文取 $(Q+R)/D_{ij}$ 为增加的信息素值;若选择该路径的蚂蚁达到一定数量或多数蚂蚁选择该路径后,当距离超过上一次的最优路径

长度时,终止遍历,并减少其总信息素量,使各条路径信息元素趋于平均值,从而使蚂蚁选择其他路径的可能性增加,使搜索得到的解趋于多样化。

参考文献

- 1 Colomi A, Dorigo M, Maniezzo V, et al. Distributed optimization by ant colonies. Proceedings of the 1st European Conference on Artificial Life, 1991, 134 - 142.
- 2 Stützle T, Hoos HH. MAX-MIN Ant System. Future Generation Computer Systems Journal, 2000, 16(8): 889 - 914.
- 3 Botee HM. Evolving Ant Colony Optimization. Complex Systems, 1998, 1(2): 149 - 159.
- 4 Dorigo M, Gambardella LM. Ant Colonies for the Travelling Salesman Problem. Biosystems, 1997, 43(2): 73 - 81.
- 5 孟令奎, 吴沉寒, 谢文军. 面向智能信息素释放的网格GIS资源查找算法. 计算机工程, 2007, 33(21): 197 - 198.
- 6 章春芳, 陈峻, 陈娟. 用自适应的多种蚁群算法求解频率分配问题. 计算机应用, 2005, 25(7): 1641 - 1644.
- 7 叶志伟, 郑肇葆. 蚁群算法中参数设置的研究——以TSP问题为例. 武汉大学学报(信息科学版), 2004, 29(7): 579 - 601.
- 8 刘立东, 蔡淮, 赵旭. 一类自适应蚁群算法的收敛性分析. 计算机应用, 2007, 27: 73 - 75.