

文章编号: 1001-0920(2009)10-1509-04

基于免疫修复的快速蚁群优化算法

闭应洲^{1,2}, 丁立新¹, 陆建波²

(1. 武汉大学 软件工程国家重点实验室, 武汉 430072; 2. 广西师范学院 信息技术系, 南宁 530001)

摘要: 蚁群优化算法通过信息素记录搜索过程中获取的知识, 并基于信息素搜索新的解. 影响信息素质量的因素主要是信息素更新策略和蚂蚁已找到的候选解的质量. 为了提高已有候选解的质量, 提出基于免疫原理识别候选解中的“病变”成分, 并对其“病变”成分进行修复. 经免疫修复后, 候选解的质量大大提高, 由它更新的信息素更好地反映了优质解的特点, 从而加快了信息的正反馈过程. 实验结果验证了该算法的有效性.

关键词: 蚁群优化算法; 免疫修复; 信息素更新策略

中图分类号: TP301.6

文献标识码: A

Fast ant colony optimization algorithm with immunity repairing

BI Ying-zhou^{1,2}, DING Li-xin¹, LU Jian-bo²

(1. State Key Laboratory of Software Engineering, Wuhan University, Wuhan 430072, China; Department of Information Technology, Guangxi Teachers Education University, Nanning 530001, China. Correspondent: BI Ying-zhou, E-mail: byzhou@163.com)

Abstract: The pheromone in ant colony optimization(ACO) are used to reflect the ants' search experience, and the ants exploit them to probabilistically construct solutions to the problem. The main factors affecting the quality of the pheromone include the policy of updating the pheromone and the quality of the constructed solutions. In order to improve the constructed solutions, a method is presented to analyze the invalid components of the constructed solution, and the invalid components are repaired with immunity operator. After immunity repairing, the quality of the constructed solutions is improved, which can exactly reflect the character of high quality solution, and speed up the positive feedback procedure. Experiment results show the effectiveness of the method.

Key words: Ant colony optimization; Immunity repairing; Policy of updating the pheromone.

1 引言

蚁群优化算法和其他启发式算法一样, 都是通过对已有候选解进行学习, 获取启发性知识来指导下一步的搜索, 但它属于基于模型的搜索^[1-3]. 蚁群优化算法通过信息素记录搜索过程中获取的知识, 并基于信息素搜索新的解, 因此信息素的质量十分重要, 而影响信息素质量的因素主要有: 蚂蚁释放信息素的策略和蚂蚁已找到的候选解的质量. 最初的蚂蚁系统(AS)算法有 3 个不同的版本, 分别为蚂蚁密度、蚂蚁数量和蚂蚁周期. 在蚂蚁密度和蚂蚁数量这两个版本中, 蚂蚁从一个城市转移到另一个城市后就直接释放信息素. 但在蚂蚁周期中, 只有所有蚂蚁都构建出一条路径后才执行信息素更新. 实践证明, 蚂蚁周期算法性能更高, 蚂蚁密度和蚂蚁数量已被淘汰^[4].

为了进一步提高算法的性能, 大量的人类智慧已被集成到 ACO 算法当中. 其中包括: 1) 每一代中只有构建最好路径的蚂蚁, 或是构建至今最优路径的蚂蚁, 才被允许释放信息素; 2) 对路径上的信息素浓度取值范围进行限制^[4,5]; 3) 信息素动态更新策略^[6]等. 通过改进信息素更新策略, 这些新的算法又比 AS 算法获得更好的性能. 为了克服蚂蚁算法初期信息素匮乏, 求解速度慢的问题, 文献[7,8]提出将遗传算法与蚂蚁算法融合, 采用遗传算法生成信息素分布.

在演化算法中, 为了降低传统交叉算子的破坏性, 文献[9,10]提出将生命科学中的免疫学原理与遗传算法结合起来, 基于免疫算子(利用领域知识)对交叉结果进行修复、改进, 整体提高算法的性能. 其原理是基于领域知识发现染色体中的“病变基因,

收稿日期: 2008-12-19; 修回日期: 2009-03-09.

基金项目: 国家自然科学基金项目(40761027); 广西教育厅科研项目(200707MS171).

作者简介: 闭应洲(1967—), 男, 广西横县人, 副教授, 博士, 从事智能计算、智能信息处理的研究; 丁立新(1967—), 男, 湖南益阳人, 教授, 博士生导师, 从事演化计算、智能信息处理的研究.

并修复这些病变基因”。由于演化算法中染色体的基因对应于蚁群算法中候选解的成分,本文提出应用免疫算子修复蚂蚁构建的候选解,提高候选解的质量,使得由它更新的信息素更好地反映优质解的特点,从而加快信息的正反馈过程,达到提高蚁群优化算法性能的目的.实验表明,通过免疫修复蚂蚁构建的候选解后,本文算法不仅耗用时间少,而且提出了求得的最优解精度.本文提出的ACO算法被称为免疫蚁群优化算法(IACO).

2 蚂蚁系统

AS算法有两个主要步骤,即蚂蚁构建问题的解和信息素的更新. $\tau_{ij}(t)$ 表示 t 时刻在城市 i, j 之间的信息素浓度.蚂蚁 $k(k=1, 2, \dots, m)$ 在搜索过程中,根据各条路径上的信息量决定搜索的转移方向. $p_{ij}^k(t)$ 表示在 t 时刻蚂蚁 k 由城市 i 转移到城市 j 的概率,即

$$p_{ij}^k = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{k \notin \text{tabu}_k} [\tau_{ik}(t)]^\alpha [\eta_{ik}]^\beta}, & j \notin \text{tabu}_k; \\ 0, & j \in \text{tabu}_k. \end{cases} \quad (1)$$

其中: η_{ij} 为先验知识或称为能见度,在TSP问题中为城市 i 转移到城市 j 的启发信息,一般取 $\eta_{ij} = 1/d_{ij}$, d_{ij} 为路径 ij 长度; α 为在路径 ij 上信息素浓度的重要程度; β 为启发信息的重要程度; $\text{tabu}_k(k=1, 2, \dots, m)$ 用来记录蚂蚁 k 当前所走过的城市,称为禁忌表.经过 n 个时刻,所有蚂蚁都完成了一次周游.当所有蚂蚁都构建好路径后,各边上的信息素将被更新.首先,所有边上的信息素都会蒸发,即

$$\tau_{ij} \leftarrow (1 - \rho)\tau_{ij}, \quad (2)$$

其中 ρ 为信息素蒸发系数.其次,在蚂蚁经过的边上都要增加信息素,即

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^k. \quad (3)$$

其中: $\Delta\tau_{ij}^k$ 是第 k 只蚂蚁向它经过的边释放信息素,有

$$\Delta\tau_{ij} = \begin{cases} Q/l_k, & \text{如果}(i, j) \text{在} L_k \text{上;} \\ 0, & \text{其他.} \end{cases} \quad (4)$$

这里: Q 为常数, l_k 为路径 L_k 的长度.在搜索过程中,蚂蚁通过信息素模型来描述搜索经验,并反映问题的特征,而且蚂蚁不断释放信息素并在这些信息素的指引下完成新的搜索.

3 蚂蚁系统的改进

蚂蚁系统(AS)用于组合优化具有很强的发现较好解的能力,在动态环境下也表现出高度的灵活性和健壮性.但AS算法存在搜索时间过长、易于停滞的问题.为了克服这些缺点,不少学者提出了改进

算法,其中比较成功的是最大最小蚂蚁系统(MMAS)算法和蚁群系统(ACS)算法^[2,4].MMAS算法^[5]在AS的基础上的主要改进为:1)只有每一代中构建最好路径的蚂蚁,或是构建至今最优路径的蚂蚁,才被允许释放信息素,以加快收敛速度;2)对路径上的信息素浓度取值范围进行限制,以克服停滞问题.最小信息素 τ_{\min} 和最大信息素 τ_{\max} 定义为

$$\tau_{\max}(t) = 1/\rho \times 1/L^*, \quad (5)$$

$$\tau_{\min}(t) = \frac{\tau_{\max}(1 - \sqrt[n]{P_{\text{best}}})}{(n/2 - 1) \sqrt[n]{P_{\text{best}}}}. \quad (6)$$

ACS对AS的改进主要体现在以下3个方面:1)采用伪随机比例规则选择下一个访问城市,最大限度地利用蚂蚁所积累的最优经验;2)信息素蒸发和信息素释放都只在至今最优路径上执行;3)蚂蚁每次使用边 (i, j) 从城市 i 移动到城市 j 后,就会降低该边上的信息素浓度,以增加探索其余边的可能性.文献^[6]对ACO算法进行改进,并取得了很好的效果,这些改进分别为:1)从不同边部城市出发和最近节点选择策略;2)信息素动态更新策略;3)最优个体变异策略.

4 基于免疫修复的蚁群优化算法

免疫的主要思想是:首先,基于对求解问题(即抗原)的分析,从中提取有用的领域知识(即疫苗);然后,利用领域知识得到求解问题的方案(抗体);最后,将此方案以免疫算子的形式具体实施.IACO的基本思想是:利用免疫算子修复人工蚂蚁构建的候选解中的“病态”基因,提高候选解的质量.从学习的角度来说,是得到高质量的训练数据,从而为得到高质量的学习结果奠定基础.然后,在高质量的信息素矩阵的指导下,产生高质量的候选解,这样就形成了一个良性循环.

4.1 免疫算子

1) 疫苗的抽取

在TSP问题中,一个具有高适应值的解决方案必然包括,而且在很大程度上包括相邻城市间距离较短的路径.这就要求在一个遍历回路中,一个城市的前驱和后继一般是它的近邻,在此将它称为“近邻原则”.如果大部分城市在遍历时违反了原则,则该回路就会由很多较长的边组成,导致它的回路路径很长.基于以上分析,在算法的开始阶段应首先将各城市的近邻城市求出来.每个城市的近邻根据城市之间的距离选择,一般选最近的5到10个城市.“近邻原则”和各城市的近邻城市就是解决TSP问题的疫苗.

2) 接种疫苗

接种疫苗就是应用预先获取的知识对候选解的

某些基因位进行修复,具体过程如下:

① 对组成候选解的所有边进行分析,并用一个结构数组记录每条较长边的一对顶点和它们之间的距离.将该结构数组 A 称为修改数组,其中数组的每个元素包含 3 个部分:两个顶点,一个距离.在此,将其中一个顶点称为源顶点,另一个称为目的顶点,根据数组元素中的距离按照降序排序.

② 对修改数组 A 的所有源顶点进行检验,判断其目的顶点是不是其近邻.比如,对于某个源顶点 V_a ,其目的顶点为 V_b ,如果 V_b 不是 V_a 的近邻,则意味着该候选解在这个基因位上可能存在问题,因此考虑对其进行修复.2-修正的机理如图 1 所示.图中:左边为候选解中城市遍历图,右边为改进后的遍历图.

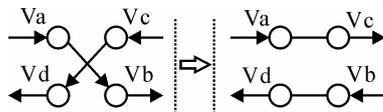


图 1 2-修正免疫机理

③ 3-修正算子与 2-修正算子相似,但在每次修复中修正 3 条边,其机理如图 2 所示.图中 3 条加粗的边被 3 条虚线替换.

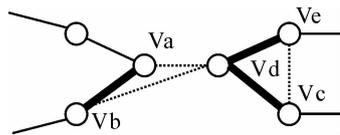


图 2 3-修正免疫机理

4.2 ACO 算子

采用类似于最大最小蚂蚁系统(MMAS)的算法和近邻节点选择策略,即只有每一代中构建最好路径的蚂蚁才被允许释放信息素,而下一个城市的选择只局限于离城市 i 较近的部分城市.当所有蚂蚁走完全部城市后,按下式对路径上的信息素进行更新:

$$\tau_{ij} \leftarrow (1 - \rho)\tau_{ij} + \Delta\tau_{ij} \tag{7}$$

$$\Delta\tau_{ij} = \begin{cases} 1/l_{\min}, & \text{如果 } (i, j) \in \text{iteration-best-tour}; \\ 0, & \text{其他.} \end{cases} \tag{8}$$

其中: ρ 表示信息素蒸发系数, l_{\min} 表示每一代最佳路径的长度, $(i, j) \in \text{iteration-best-tour}$ 表示边 (i, j) 属于每一代最佳路径.但信息素浓度取值按式(5)和(6)进行限制.免疫蚁群优化算法(IACO)的工作框架如图 3 所示.具体步骤如下:

Step1: 抽取疫苗.

Step2: 初始化.按照式(5)将每个边上的信息素初始化,将 m 只蚂蚁随机地分配到 n 个城市,并将出发点城市设置到禁忌表中.

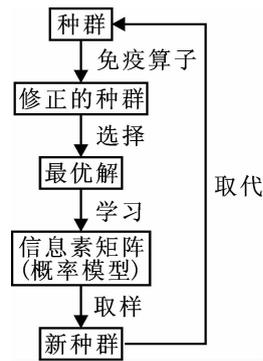


图 3 免疫蚁群优化算法 IACO 的工作框架

Step3: 构建候选解.每只蚂蚁按式(1)从城市 i 的近邻中选择下一个城市,如果所有的近邻都已访问过,则从没访问过的城市选择一个最近的城市,并修改禁忌表.

Step4: 接种疫苗.当 m 只蚂蚁走完所有城市后,对所有的候选解接种疫苗,并应用 2-修正算子对每个候选解进行修正.

Step5: 评估候选解.计算经免疫修正后的候选解长度,并根据路径长度选择最优候选解,并应用 3-修正算子对最优候选解进行修正.

Step6: 信息素更新.根据式(7)和(8)更新所有边上的信息素浓度.

Step7: 搜索终止判断.判断设置的搜索次数是否达到,如果未到,则清空禁忌表,转 Step3 继续执行,否则停机推出.

5 实验

为了比较算法的性能,本文选用了 2 个中等规模的 TSP 问题:TSP200 和 Fl417.其中:TSP200 包含 200 个城市,而这些城市的位置都是通过随机函数在二维空间生成的, $x \in [0, 1000], y \in [0, 1000]$,城市之间的距离为欧氏距离;Fl417 来自 TSPLIB^[11],其包含的城市个数为 417.之所以选择这两个 TSP 问题,是因为在不应用与问题相关领域知识的情况下,使用 ACO 算法求解超过 500 个城市的 TSP 问题花费时间太多,而城市数目太少(少于 100 个)又过于简单.计算机实验平台是:CPU: 2.93GHz;Ram:1G;操作系统:Windows XP;开发工具:VC6.0.

ACO 算法中参数设置为: $\alpha = 1, \beta = 3, \rho = 0.1$;蚂蚁数为 20;最大运行代数为 1800 代.ACO 算法选择下一城市时,只从 10 个离当前城市 i 较近的近邻中选择,如果所有的近邻之前已被选择,则选择距离城市 i 最近的未访问城市.基于上述参数设置,独立运行 20 次,实验结果如表 1 所示.

IACO 算法中参数设置为:第 1 代到第 10 代, α

$= 1, \beta = 2$; 第11代到第30代, $\alpha = 2, \beta = 3$; TSP200中 $\rho = 0.2$, Fl417中 $\rho = 0.4$. 蚂蚁数为20, 最大运行代数数为30代, 基于上述参数设置, 独立运行20次, 实验结果如表2所示.

表1 ACO算法对TSP200, Fl417的部分运行结果

代数 /代	TSP200($t = 39.8s$)			Fl417($t = 163.8s$)		
	最小	最大	平均	最小	最大	平均
100	11574	12740	12239	13729	14069	14180
500	10933	11113	10985	12817	12938	12904
1000	10919	11092	10931	12780	12938	12890
1500	10892	11092	10910	12769	12928	12880
1800	10787	11002	10890	12710	12920	12822

表2 IACO算法对TSP200, Fl417的部分运行结果

代数 /代	TSP200($t = 6.7s$)			Fl417($t = 46.9s$)		
	最小	最大	平均	最小	最大	平均
初始	11046	11444	11202	12232	12722	12598
10	10863	10980	10890	12228	12409	12390
20	10725	10908	10835	12125	12308	12353
30	10615	10748	10658	11862	12226	12118

由实验结果可看出, 经免疫修复后 IACO 算法比 ACO 算法又快又好. 尽管对蚂蚁构建的候选解进行免疫修复需要时间, 但总体来说还是有价值的. 在免疫算子的作用下, 候选解的质量有了很大提高, 针对每代最优解的学习结果(信息素矩阵)正确反映了搜索空间的特点, 即城市之间的依赖关系, 因此, 更充分地利用这些联接信息有利于搜索效率的提高.

ρ 表示信息素蒸发系数, 根据式(7), 如果 ρ 的值较大, 则未被最优蚂蚁选中的边上的信息素浓度将迅速降低, 这使搜索空间缩小, 算法的收敛速度更快, 但易陷入局部最优. IACO 算法的实验结果表明, 在 α, β 参数相同的情况下, 当 $\rho = 0.2$ 时比 $\rho = 0.4$ 时所花的时间更多. 在 TSP200 中 $\rho = 0.2$ 时的结果好些; 而 Fl417 中 $\rho = 0.4$ 时的结果好些. 研究参数 α, β, ρ 的最佳配置, 对发挥蚁群算法在实际问题中的作用有很重要的意义.

6 结 论

蚁群优化算法属于基于模型的搜索. 它的工作机理就是选取好的候选解进行学习, 通过获取启发性知识(信息素模型)指导下一步的搜索. 利用免疫算子修复人工蚂蚁构建的候选解中的“病态”基因,

以提高候选解的质量. 从学习的角度来说, 是得到高质量的训练数据, 从而为得到高质量的学习结果奠定基础. 在高质量信息素矩阵的指导下, 产生高质量的候选解, 这样就形成了一个良性循环, 从而使算法获得良好的性能.

参考文献(References)

- [1] Dorigo M, Socha K. An introduction to ant colony optimization [R]. Belgium: Universite Libre de Bruxelles, 2006.
- [2] Dorigo M. The ant colony optimization metaheuristic: Algorithms, applications, and advances [R]. Belgium: Universite Libre de Bruxelles, 2000.
- [3] Zlochin M, Birattari M, Meuleau N, et al. Model-based search for combinatorial optimization: A critical survey [J]. Annals of Operations Research, 2004, 131(1-4): 373-395.
- [4] Dorigo M, Stutzle T. Ant colony optimization [M]. London: MIT Press, 2004.
- [5] Stutzle T, Hoos H. Max-min ant system [J]. Future Generation Computer Systems, 2000, 16(9): 889-914.
- [6] 朱庆保, 杨志军. 基于变异和动态信息素更新的蚁群优化算法 [J]. 软件学报, 2004, 15(2): 185-192. (Zhu Q B, Yang Z J. An ant colony optimization algorithm based on mutation and dynamic pheromone updating [J]. J of Software, 2004, 15(2): 185-192.)
- [7] Ding J, Tang W, Wang L. Parallel combination of genetic algorithm and ant algorithm based on dynamic K-means cluster [C]. Lecture Notes in Artificial Intelligence. Berlin: Springer, 2006, 4114: 825-830.
- [8] 丁建立, 陈增强, 袁著祉. 遗传算法与蚂蚁算法的融合 [J]. 计算机研究与发展, 2003, 40(9): 1351-1356. (Ding J L, Chen Z Q, Yuan Z Z. On the combination of genetic algorithm and ant algorithm [J]. J of Computer Research and Development, 2003, 40(9): 1351-1356.)
- [9] Bi Y Z, Ding L X, Ying W Q. Towards the role of heuristic knowledge in EA [C]. Lecture Notes in Computer Science. Berlin: Springer, 2007, 4683: 621-630.
- [10] Bi Y Z, Ding L X, Ying W Q. Towards a less destructive crossover operator using immunity theory [C]. Lecture Notes in Computer Science. Berlin: Springer, 2007, 4490: 1061-1067.
- [11] <http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/>.