

基于多信息素的蚁群算法*

An Ant Colony Algorithm Based on Multi-Pheromones

区云鹏, 韦兆文, 蒋慧超

OU Yun-peng, WEI Zhao-wen, JIANG Hui-chao

(广西大学计算机与电子信息学院, 广西南宁 530004)

(School of Computer, Electronics and Information, Guangxi University, Nanning, Guangxi, 530004, China)

摘要:针对传统增强型蚁群算法容易出现早熟和停滞现象的缺陷,提出一种多信息素的蚁群算法(MPAS),并以 TSPLIB 的数据为例对该算法进行实验测试。MPAS 算法将信息素分为局部和全局两种不同的信息素,在搜索过程中,对局部和全局信息素采用不同的更新策略和动态的路径选择概率,使得在搜索的中后期能更有效地发现全局最优解。在中大型问题上 MPAS 算法有着更好的发现最优解的能力。

关键词:蚁群算法 信息素 旅行商问题

中图分类号: TP301.6 **文献标识码:** A **文章编号:** 1002-7378(2008)03-0240-03

Abstract: This paper improves an ant colony algorithm based on multi-pheromones and solves the problem existed in classical augment ant colony algorithm. The basic idea is to divide the pheromone into local pheromone and global pheromone. Then their pheromones are updated using different strategies during searching optimal path. Many experiments based on the data of TSPLIB show the advantages of this algorithm in sweeping problems.

Key words: ant colony algorithm, pheromones, traveling salesman problem

TSP 问题是一个 NP 难问题,其定义是:给定 n 个城市的集合 $\{1, 2, \dots, n\}$ 及城市之间的花费 c_{ij} ($1 \leq i \leq n, 1 \leq j \leq n, i \neq j$), 找到一条过且只过所有城市一次并且花费最小的回路^[1]。针对这一问题,意大利学者 Macro Dorigo 等人于 1991 年提出了蚁群算法,并取得了较好的成果^[2,3]。蚁群算法属于演化算法,其核心就是在“探索”和“利用”之间寻找一个平衡点。即要使得算法的搜索空间尽可能的大,以寻找那些可能存在最优解的解空间;同时又要充分利用当前群体内具有的有效信息,使得算法的侧重点放在那些可能具有较高适应值的个体所在的空间内,从而以较大概率收缩到全局最优解^[4]。

Macro 等用于解决 TSP 问题的蚁群系统,当搜索空间较小时,可以搜索到满意解,而在搜索空间较

大时,机器运算次数将迅速增多,并且会出现停滞现象。针对这种情况学者们提出了一些改进方法,如最大-最小蚂蚁系统(MMAS)^[5],最优-最差蚂蚁系统(BWAS)^[6],遗传算法与蚁群算法的融合(GAAA)算法^[7]等。这些增强型蚁群算法,主要是通过增强全局最优解或局部最优解的所有路径中信息素强度来提高蚁群算法的搜索能力。本文引入多种信息素的概念,提出一种多信息素蚁群算法(MPAS),以期在加速收敛发现最优和防止停滞之间找到平衡。

1 蚁群算法分析

蚁群算法实现的关键是信息素的释放和更新,以及蚂蚁个体依据信息素进行路径选择的策略。传统蚁群算法在后期容易陷入局部最优解的原因是由正反馈现象所引起。在每次循环结束时,对信息素进行全局更新,从而增强了目前最优路径对蚂蚁的“吸引力”。在前期,这“吸引力”对算法起到了很好的加速作用。然而,它也会导致算法过早停滞。若单纯地调低信息素的权重,会削弱正反馈的机制的作用,很

收稿日期: 2007-09-25

作者简介: 区云鹏(1981-),男,硕士,主要从事人工智能研究。

* 广西自然科学基金项目(桂科自 0640026)资助。

难使信息素集中分布,蚂蚁也就失去了选路的参照,退化为简单的以路径为参照的搜索.目前,绝大多数的传统增强型蚁群算法均采用一种不变的策略模式去对一种信息素进行更新,以确定的概率选择方法进行选择,不能有效避免算法容易出现早熟和停滞现象.如果对蚁群算法的局部信息素和全局信息素进行区分,采用不同的更新策略,根据发现解的情况动态地调整选择概率,这样就可以在加速收敛发现最优和防止停滞之间找到平衡.

2 基于多信息素的蚁群算法

2.1 路径选择概率公式

蚂蚁从城市 i 转移到城市 j 的概率计算公式为

$$p_{ij}^k(t) = \begin{cases} \frac{F(\tau_{ij}^k(t), \lambda_{ij}^k(t), \rho) \times \eta_{ij}^k(t)}{\sum_{s \in \text{allowed}_k} F(\tau_{ij}^k(t), \lambda_{ij}^k(t), \rho) \times \eta_{ij}^k(t)}, j \in \text{city}_k; \\ 0, j \notin \text{city}_k, \end{cases} \quad (1)$$

其中 $\text{city}_k (k = 1, 2, \dots, m)$ 表示蚂蚁 k 已经走过城市的集合,开始时 city_k 中只有一个元素,即蚂蚁 k 的出发城市,随着计算的进行, city_k 中的元素不断增加; $\text{allowed}_k = \{0, 1, \dots, n\}$ city_k 表示蚂蚁 k 下一步允许选择的城市; η_{ij} 是能见度,一般取路径 (i, j) 长度的倒数; $\tau_{ij}(t)$ 表示 t 时刻在城市 i 与城市 j 连线上的局部更新的信息素的浓度. $\lambda_{ij}(t)$ 表示 t 时刻在城市 i 与城市 j 连线上进行全局更新的信息素的浓度; α 表示信息素浓度 τ_{ij} 的重要性; δ 表示信息素浓度 λ 的重要性; β 负责调节能见度 η 的相对重要程度; $F(\tau, \lambda, \rho)$ 是一个以信息素 τ, λ 和权重 ρ 为参数的函数, $F(x, y, \rho) = \rho \times x + (1 - \rho) \times y$, 权重 ρ 可变(取值区间为 $[0, 1]$), 用于控制两种信息素的比重. 当发现新的解时,减少 ρ 的值以增加全局信息素的权重,从而有利于更快的发现其附近的解;若迭代多次而没有发现新解,则增加 ρ 的值削弱全局信息素的权重,从而扩大搜索的范围.

2.2 不同信息素更新公式

蚂蚁在每移动一步之后,进行的局部信息素更新公式为

$$\tau_{ij}(t + \omega) = (1 - \theta) \times \tau_{ij}(t) + \theta \times \Delta\tau, \theta \in (0, 1), \quad (2)$$

其中, $\Delta\tau$ 表示信息素的释放量,其值可视路径长短而定,路径越短,释放的量就越多;反之则越少. θ 为一个参数.

当蚂蚁完成一次循环,各路径上的全局信息素

的浓度根据公式

$$\lambda_{ij}(t + \omega) = \lambda_{ij}(t) + \Delta\tau_{ij} \quad (3)$$

进行调整,其中: $\Delta\tau_{ij}$ 表示第 k 只蚂蚁在本次循环完成后在整条路径上进行全局更新的信息素的浓度(其取值为目前最优路径长度的倒数).

在算法执行的过程中,可以动态地改变 α, β, δ 及 ρ 的取值. 在初始阶段 α, δ 可以附较高的值. 在中后期当出现停滞时,可以提高 δ 的取值(信息素的值始终在 $[0, 1]$ 区间)和增加 ρ 的值,降低全局更新信息素的权重,降低蚂蚁对全局信息素的敏感度,从而扩大其搜索范围,提高其全局搜索能力. 当搜索陷入长期的停滞时,重新初始化各路径上信息素的值,以解决搜索停滞的问题.

2.3 算法流程

多信息素蚁群算法流程如下.

(1) 初始化

随机产生一个初始解,它是由一只蚂蚁按选择相邻最近的未访问的城市而形成的. 初始化每条路径的信息素. 其信息量是初始解的值 init_value 与城市的规模 n 的乘积的倒数:

$$\frac{1}{\text{init_value} \times n}$$

(2) 迭代过程

While not 结束条件 do

(2.1) for $i = 1$ to n do (循环 n 次)

for $k = 1$ to m do

(2.1.1) 每只蚂蚁根据公式(1)的选择概率选择下一个城市

(2.1.2) 按公式(2)进行局部信息素更新
end for k

end for i

(2.2) for $k = 1$ to m do (每只蚂蚁均访问了所有的城市,现在返回初始的城市)

(2.2.1) 每只蚂蚁返回初始的城市

(2.2.2) 按公式(2)进行局部信息素更新
end for k

(2.3) 按公式(3)进行全局信息素的更新

(2.4) 根据本次得到的结果,调整 $\alpha, \beta, \delta, \rho$ 的

值

(2.5) if (满足搜索长期停滞的条件)

(2.5.1) 重新初始化各路径的信息素浓度

end if

end while

3 数据实验及结果分析

从 TSPLIB 中选用两种 TSP 问题 (pcb442、pr2392)^[8], 用多信息素蚁群算法和传统蚁群算法在计算机上用 C 语言编程进行测试 (计算机为 P4 2.6G, 512MB 内存). $\alpha, \beta, \delta, \rho$ 的初始值分别为 1.0、1.0、2.0 和 0.5, 蚂蚁的个数为 25.

小规模 pcb442 问题运行 100 次, 每次 300s 的结果见表 1. 中大规模的 pr2392 问题运行 30 次, 每次 1500s 的结果见表 2.

表 1 pcb442 问题测试结果

算法	最优解	最优解平均值	平均代数	平均时间(s)	发现最优解的概率(%)
ACS	50778	50780.31	3214.81	102.99	94
BWAS		50813.78	2271.48	76.09	70
MMAS		50778.62	3604.74	80.33	94
MPAS		50797.09	2700.50	121.05	79

表 2 pr2392 问题测试结果

算法	最优解	最优解平均值	平均代数	平均时间(s)	发现最优解的概率(%)
ACS	3780.32	379607.90	4415.33	1402.41	0
BWAS		379237.80	2666.90	1080.28	0
MMAS		378635.30	2997.57	830.74	0
MPAS		378743.40	2237.03	981.52	40

表 2 结果显示, MPAS 在对 pr2392 问题运行 30 次的实验中, 有 12 次发现最优解, 占总次数的 40%; 而其它传统增强型蚁群算法没有发现最优解. 在最优解平均值方面, MPAS 与 MMAS 相当, 明显优于 ACS 和 BWAS. MPAS 中平均运行时间上并不具有优势但是在平均迭代次数上有明显的降低.

表 1 和表 2 结果表明, MPAS 在解决小规模问题的时候运行时间和迭代次数并不具有优势. 在传统算法中, 通常只对一种信息素进行更新操作, 而 MPAS 要花费更多的时间处理全局和局部两种不同的信息素, 因而其每代所花费的时间大于其它算法. 在小规模问题上, 其解空间较小, 只需要较少的几次搜索就可以发现最优解, MPAS 在迭代次数方

面的优势不明显.

4 结束语

本文提出一种改进的基于多信息素的蚁群算法, 与传统增强型蚁群算法的搜索机制相比, 由于该算法对全局和局部信息素采用了不同的更新策略和动态的路径选择概率, 使得在大规模系统中, 搜索的中后期能更有效地发现全局最优解全局信息素, 从而提高了发现全局最优解的概率. 此外, 该算法通过改变全局信息素与局部信息素的权重, 引导蚂蚁搜索的方向, 从而加快收敛速度, 能够以较高的效率逼近最优解, 即明显降低了迭代次数. 实验也表明在中大规模系统中, MPAS 有较高的发现全局最优解的能力并有效降低了迭代次数.

参考文献:

- [1] 王晓东. 计算机算法分析与设计[M]. 北京: 电子工业出版社, 2001: 114-146.
- [2] Dorigo M, Maniezzo V, Colomi A. Positive feedback as a search strategy[R]. Technical Report, 1991: 91-016.
- [3] Dorigo M, Gambardella L M. Ant colony system: a cooperative learning approach to the traveling salesman problem[J]. IEEE Trans on Evolutionary, 1997, 1(1): 53-66.
- [4] 陈宏建, 陈峻, 徐晓华, 等. 改进的增强型蚁群算法[J]. 计算机工程, 2005, 31(2): 176-178.
- [5] Stutzlet T, Hoos H. Improvements on the ant system: introducing MAX-MIN ant System; proceedings of the International Conference on Artificial Neural Networks and Genetic Algorithms [C]. Wien: Springer Verlag, 1997: 245-249.
- [6] Cordon O, Fernandez I, Herrera F. A new ACO model integrating evolutionary computation concepts; the best-worst ant system; abstract proceedings of ANTS2000-From Ant Colonies to Artificial Ants; A Series of International Workshops on Ant Algorithms [C]. [S. l.]: [s. n.], 2000: 22-29.
- [7] 丁建立, 陈增强, 袁著祉. 遗传算法与蚁群算法的融合[J]. 计算机研究与发展, 2003, 40(9): 1531-1536.
- [8] Gerhard Reinelt. TSPLIB [EB/OL]. [2007-09-01]. <http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95>.

(责任编辑: 韦廷宗)