

文章编号:1005-0523(2013)06-0030-05

# 优化蚁群算法在最短路径诱导中的应用

陶宇,吴中

(河海大学土木与交通学院,江苏南京 210098)

**摘要:**最短路径算法在许多应用领域和研究中起着十分重要的作用。现有文献对最短路径问题提出了大量的优化求解方法和算法,大部分研究仅针对固定权值网络,对权值随时间变化等时变情况考虑较少。在通信系统、智能交通系统等实际网络及应用领域中,随着时间的变化,边的权值往往也同时改变。因此,时变网络中最短路径求解问题的研究更具有实用意义。针对一般算法存在的缺陷,现提出三点优化,使算法既能避免陷入局部最优解,又能更快地收敛到全局最优解。

**关键词:**最短路径;时变网络;优化;路径诱导

**中图分类号:**TP301.6

**文献标志码:**A

均衡网络流问题的最短路径算法,包括 Frank-Wolfe 算法<sup>[1]</sup>和弹性需求用户均衡问题的网络变化法<sup>[2]</sup>。而上述两种算法的一个共同核心部分就是最短路径的搜索。对于网络中边属性值为时变的情况,传统的最短路径算法不适用,要求解某一类动态网络中的最短路径问题,有必要建立一套严格的约束条件。国内众多知名学者提出多种优化方法<sup>[3-4]</sup>,俞云新等人提出的“两步走”策略及基于粒子群的参数最优组合算法,克服了蚁群算法随机参数的缺陷,王更生等人提出了基于 DACS3 的改进蚁群算法,此算法在搜索下一城市时,仅考虑信息素强度特点,使运算简单、快捷。问题在于,用 TSP(旅行商)问题描述蚁群算法模型,其属于 NP(非确定多项式)问题,很难兼顾时间效率和较优路径两个方面。在已有成果的基础上,本文对蚁群算法问题进一步进行优化研究。

由 Dorigo<sup>[5-6]</sup>首先提出的智能蚁群算法(ACO)是一种新的仿生类进化算法。此种启发式算法模仿了蚂蚁的觅食行为。蚁群算法有两个核心点,分别为城市间转移策略和信息素更新策略;其中转移概率的计算要依赖信息素的浓度。因此在蚁群算法中,信息素的更新策略显得尤为重要,直接决定了算法的收敛速度和收敛效果。

## 1 模型简介

蚁群算法的网络模型可表示为  $(V, A, T)$ , 其中  $V=(v_1, v_2, \dots, v_n)$  是有限点集;  $A$  是有限边集,  $A \in V \times V$ ;  $T = \{g_{ij}(t) | (v_i, v_j) \in A, t \in [t_0, t_m]\}$  是定义于每一条边  $(v_i, v_j)$  在时间  $t \in [t_0, t_m] \in R$  的区间上的函数。 $g_{ij}(t)$  表示非负实数,为节点  $v_i$  在  $t$  时刻出发到达节点  $v_j$  所需要的时间;  $[t_0, t_m]$  为时间上下界,  $\forall t \in [t_0, t_m]$ ,  $t + g_{ij}(t) \in [t_0, t_m]$ ,  $t_0$  是最早出发时间,  $t_m$  是终止时间。

模型描述为

$$\text{Min} \sum_{i=1}^{n-1} T_{v_i, v_{i+1}} \quad (1)$$

收稿日期:2013-07-31

作者简介:陶宇(1989—),男,硕士研究生,研究方向为交通运输规划与管理。

$$\text{s.t. } T_{v'_i, v'_{i+1}} = g_{v'_i, v'_{i+1}} \left( \sum_{i=1}^{k-1} T_{v'_i, v'_{i+1}} + t_s \right) \quad (2)$$

$$(v'_1, v'_2), \dots, (v'_{n-1}, v'_n) \in A \quad (3)$$

$$v'_i \in V, i = 1, 2, \dots, n \quad (4)$$

$$0 < k < n, k \in N \quad (5)$$

式中:  $v'_0$  表示出发地;  $v'_n$  表示目的地;  $t_s$  为出发时间。式(2)表示在  $t_s$  出发,由路径  $(v'_1, v'_2, \dots, v'_n)$  从节点  $v'_k$  到节点  $v'_{k+1}$  所需的时间。式(3)(4)为约束条件,保证了路径  $(v'_1, v'_2, \dots, v'_n)$  在网络上的连通性。

蚁群转移概率的计算公式:

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in w_k} [\tau_{il}(t)]^\alpha [\eta_{il}]^\beta} & j \in w_k \\ 0 & \text{else} \end{cases} \quad (6)$$

$w_k = \{c - \text{tabu}_k\}$  为蚂蚁  $k$  下一步允许选择的城市,  $\text{tabu}_k$  记录了蚂蚁  $k$  所走的城市;  $\alpha$  表示信息启发因子,体现了某条轨迹的重要程度;  $\beta$  表示期望启发因子,体现了能见度的重要程度;  $\eta_{ij}(t)$  表示启发函数,其表达式如下:

$$\eta_{ij} = \frac{1}{d_{ij}} \quad (7)$$

其中  $d_{ij}$  表示城市  $i$  与城市  $j$  之间的距离。蚂蚁每走完一步,更新残留信息。 $t+n$  时刻在路径  $(i,j)$  上的信息量按下式调整。

$$\tau_{ij}(t+n) = (1-\rho) \times \tau_{ij}(t) + \tau_{ij}(t) \quad (8)$$

$$\tau_{ij}(t) = \sum_{k=1}^m T_{ij}^k(t) \quad (9)$$

式中:  $\rho$  为信息素的挥发系数;  $\tau_{ij}(t)$  为某次循环中路径  $(i,j)$  上信息素的增加量;  $T_{ij}^k(t)$  为第  $k$  只蚂蚁在某次循环中,在路径  $(i,j)$  上残留的信息素量。

## 2 算法优化

### 2.1 蚁群初始分布的优化

基于信息素扩散策略的双种群蚁群算法<sup>[7]</sup>(PDDPAS),该算法通过建立信息素扩散模型,在每种蚁群局部信息素更新上采用扩散策略,使得蚂蚁个体间能更好地发挥协作能力。算法中,种群一的  $\beta$  和  $\rho$  都取较大值,目的是加快其收敛速度,种群二采用信息素扩散策略,提高其寻优能力,使算法具有更好的计算性能。现提出优化。一个种群的分布规律表现为:各分节点蚂蚁的分布个数随着离目的地节点距离的增加而增多,另一种群则相反,离目的地节点越近,各分节点上蚂蚁的分布个数占蚂蚁总数的比例越高。经过指定次数迭代后,进行信息的交换。双种群的信息素平均运算按照下式进行。

$$\left. \begin{aligned} \tau_{ij}^A &= \frac{(\tau_{ij}^A + \tau_{ij}^B)}{2} \\ \tau_{ij}^B &= \tau_{ij}^A \end{aligned} \right\} \quad (10)$$

信息得到交换后,不但可以使两个种群分享彼此的搜索成果,而且避免了单个种群经过多次内部迭代后出现停滞的状态。如此改进后,两个种群便有侧重的进行搜索,再配合杂交算法<sup>[8]</sup>,此算法开始运行时,赋予每条边上相等的信息量,此后每次迭代的路径选择都通过轮盘赌来实现,每当有蚂蚁选中一条路径,此路径上的信息量就按照一定的策略减少,增加了蚂蚁选择其他路径的概率。直到每只蚂蚁都找到目的节点,此时第一轮扫描结束。第一轮扫描过后,能得到若干个解,找到如下两个解,按指定方式进行杂交:例如,路径  $L_1 = \{2, 3, 9, 8, 5, 11\}$  和路径  $L_2 = \{1, 3, 7, 6, 5, 13\}$ , (数字代表节点)。因为  $L_1, L_2$  有共同节点3

与5,现提出,两条路径3到5之间的数字序列可以交换,交换后产生的新路径为:  $L'_1 = \{2, 3, 7, 6, 5, 11\}$ ,  $L'_2 = \{1, 3, 9, 8, 5, 13\}$ 。如果  $L'_1 \leq \min(L_1, L_2)$  或  $L'_2 \leq \min(L_1, L_2)$ , 则按下式对  $L'_1$  或  $L'_2$  对应路径上每条边的信息素进行更新,  $L_1$  或  $L_2$  被取代, 否则不进行任何更新。

$$\tau_{ij} = \tau_{ij} + \frac{1}{L'_1} \quad \text{或} \quad \tau_{ij} = \tau_{ij} + \frac{1}{L'_2} \quad (11)$$

优化后,蚁群一更擅长选取离目的地较远的最优路径,蚁群二则刚好相反。蚁群二的存在,大大加快了算法的收敛速度。杂交法和信息交换法配合使用后,不仅有效避免了算法收敛于局部最优解和早熟收敛的情况,而且收敛速度明显加快,找到最优解的稳定性也得到了提高。

### 3.2 参数 $\alpha, \beta$ 的优化

当  $\alpha \in (1.0, 4.0)$  时,蚁群算法有较好的综合求解性能。现提出优化,  $\alpha$  (信息启发因子)采用如下公式进行更新,不再是固定不变的参数。

$$\alpha_{\text{NEW}} = \frac{g_{ij}^g}{g_{ij}^i} \alpha_{\text{OLD}} \quad (12)$$

式中:  $g_{ij}^g$  为  $ij$  路径上花费时间的全局最优解;  $g_{ij}^i$  为  $ij$  路径上花费时间的局部最优解。

优化后的算法体现了一种激励机制,使最优路径能更快速的被搜索到,算法的收敛速度得以加快。

$\beta$  (期望启发因子)不再是一个固定值的数值,众所周知,天气情况 ( $\omega_{ij}$ )、道路等级 ( $h_{ij}$ )、道路情况 ( $r_{ij}$ ) 等不确定因素对公路交通会有较大影响。现提出,上述3个态参数的取值范围参考下表。其中  $k$  为一较大数值。

表1 状态参数取值范围

Tab.1 Range of the state parameter

天气情况 ( $\omega_{ij}$ )				道路等级 ( $h_{ij}$ )			道路情况 ( $r_{ij}$ )			
晴天	小雪(雨)	中雪(雨)	大雪(雨)	高速	一、二级	三、四级	畅通	拥堵	中断	
取值	0	0.15~0.25	0.25~0.45	0.45~0.8	0.1~0.3	0.3~0.4	0.4~0.9	0	0.2~0.5	$k$

另设置参数  $\varphi$ , 开始计算之前,先给  $\varphi$  设置几个阈值。  $\beta$  的取值根据  $\omega_{ij}$ ,  $h_{ij}$ ,  $r_{ij}$  3个参数相加之和间于  $\varphi$  的不同的阈值之间而有所不同。优化后的算法更符合实际情况,更容易找出真正的最优解。

$\rho$  (信息素挥发度)不再是一个固定的数值,现引入  $\rho^k$ , 此参数表示第  $k$  只蚂蚁的信息素挥发度。现优化为  $\rho^k = \rho \left( \frac{T^g}{T^i} \right)$ 。式中  $T^i$  表示第  $k$  只蚂蚁当前选择路径总共花费的时间,  $T^g$  表示目前最优路径所要花费的总时间。使得精英蚂蚁的信息素挥发变慢,加快非精英蚂蚁信息素挥发速度。同时也体现了对精英蚂蚁的一种奖励机制。算法的全局收敛性和收敛速度得以进一步提高。

### 3 程序模拟及效果

为了得出较优的算法,现配合使用上述3种优化方法,并根据优化后的算法编写相关程序,对  $\varphi$  的阈值设置、 $\alpha$  (信息启发因子)、 $\beta$  (期望启发因子)、 $\rho$  (信息素挥发度)的最优取值以及算法的最终效果进行考证。图1为用C语言编写的部分优化函数,图2为演示窗口。

在图2演示窗口中,右下角蓝色圆圈为食物,左上角绿色圆圈为蚁窝,棕色方块为障碍物,黑色小圈表示蚂蚁(找到食物后,蚂蚁变成紫色)。演示中选择了显示信息素的分布,绿色线条代表蚁窝信息素,颜色深浅表示信息素浓度大小;蓝色线条代表食物信息素。右上图为初始画面,右下图为已找到最优路径的画面,其中蓝色线条颜色最深的路径即为最优路径。

本文根据TSP问题中的Eil51这组数据对算法做了仿真,编写了相关程序,对基本蚁群算法和本文提

出的优化后的算法进行运算。其中,迭代次数和运算次数分别为1 000次和50次,具体参数见表2。最后绘制了两种算法的收敛过程对比图,详见图3。

```
// =====一些接口函数 =====
public int GetX()
{ return NowLocation.X;}
public int GetY()
{return NowLocation.Y;}
public bool HaveFood()
{if (FoodNum > 0) return true;
else return false;}
// 以下这几个设置可以实时更改
public void SetSmellDropRate(float value)
{if (value >= 1 || value <= 0) return;
SmellDropRate = value;
PlayGround.ShowMessage = string.Format("信息素释放率更改为{0}",value);}
public void SetErrorRate(float value)
{if (value >= 1 || value <= 0) return;
ErrorRate = value;
PlayGround.ShowMessage = string.Format("蚂蚁出错概率更改为{0}", value);}
```

图1 优化函数

Fig.1 Optimization function

表2 参数取值及运算结果

Tab.2 Selection of parameters and operation results

参数	基本蚁群算法	优化后的算法
$\alpha$	1	不固定
$\beta$	5	不固定
$\rho$	0.5	不固定
种群一在起点处蚂蚁数	40	15
种群一在靠近终点处蚂蚁数		5
种群二在起点处的蚂蚁数		5
种群二在靠近终点处蚂蚁数		15
最优路径长度	437.826 068	428.324 485
平均路径长度	445.736 519	435.815 457

### 4 结语

对蚁群算法进行了优化,通过信息素强度的局部和全局更新策略以及引入新的算法,有效解决了一般算法容易陷入局部收敛的问题,寻求全局最优解的能力大为增强。由实际模拟结果得出,采用上述改进方法后的最大优势是搜索效率和结果的合理性都大为改善。

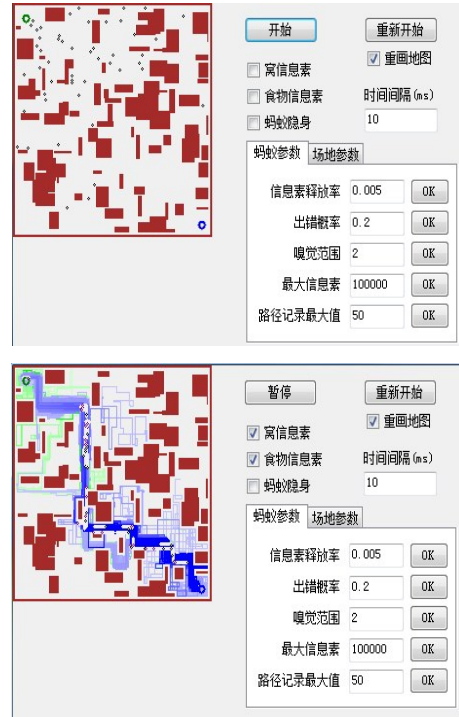


图2 演示窗口

Fig.2 Demonstration window

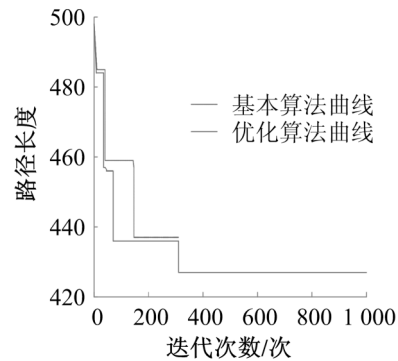


图3 两种算法收敛过程图

Fig.3 Convergence figure of two kinds of algorithms

**参考文献:**

- [1] 徐猛,屈云超,高自友. Frank-Wolfe算法求解交通分配问题[J]. 交通运输系统工程与信息,2008,8(3):14-22.
- [2] 张波,隗志才,林徐勋. 基于有限理性的弹性需求随机用户均衡交通分配模型[J]. 计算机应用研究,2011,28(9):3268-3271.
- [3] 俞云新,王更生. 基于粒子群的蚁群算法参数最优组合研究[J]. 华东交通大学学报,2010,27(1):47-51.
- [4] 王更生,俞云新,蔡求元,等. 基于DACS3的改进蚁群算法求解TSP问题[J]. 华东交通大学学报,2010,27(2):57-62.
- [5] DORIGO M, BIRATTARI M, STUTZLE T. Ant colony optimization-artificial ants as a computational intelligence technique[J]. IEEE Computational Intelligence Magazine, 2009, 1(4): 28-39.
- [6] DORIGO M, CARO G D, GAMBARDELLA L M. Ant algorithms for discrete optimization[J]. Artificial Life, 1999, 5(2): 137-172.
- [7] 朱凯燕,成谱斌. 基于信息素扩散机制的双种群蚁群优化算法[J]. 微计算机应用,2010,31(7):22-26.
- [8] 陈立潮,刘佳,吕亚男. 带杂交算子的蚁群算法求解动态网络中的最短路径问题[J]. 计算机工程与科学,2007(5):81-82.

## Application of Optimized Ant Colony Algorithm in the Shortest Route Guidance

Tao Yu, Wu Zhong

(College of Civil and Transportation Engineering, Hohai University, Nanjing 210098, China)

**Abstract:** The shortest path algorithm has played an important role in many applications and researches. There are many literatures concerning optimization and algorithm for the shortest path problem, but most studies were applied to the network of fixed weight values without considering weight value change over time. In the actual network, such as intelligent transportation system and communication system, the weight values may change over time. Therefore, the study of shortest path problem in time-varying network is more valuable. In response to the defects of the general algorithm, this paper puts forward some improvements to converge faster the shortest path problem to the global optimal solution by avoiding the local optimal solution.

**Key words:** the shortest path; time-varying network; optimization; route guidance