改进蚁群算法求解时变网络中最短路径问题

刘永强 常 青 熊华钢

(北京航空航天大学 电子信息工程学院, 北京 100191)

摘 要: 给出一种时变网络中蚁群算法的信息素更新策略,使边上残留信息素能够正确反映时变网络中边上权值的变化情况;改进了传统蚁群算法的相邻节点选择策略,使蚂蚁只需计算与当前节点存在直接路径的节点的转移概率,降低算法的计算量;将蚁群算法和遗传算法结合,将蚁群算法每次遍历后形成的解作为初始群种进行单点交叉计算,避免陷入局部最优解,提高算法收敛速度. 仿真结果表明,改进的蚁群算法能够有效求解时变网络中最短路径问题,比传统蚁群算法得到全局最优解的概率更大,算法的收敛速度更高.

关 键 词: 时变网络; 最短路径; 蚁群算法

中图分类号: TP 301. 6

文献标识码: A **文 章 编 号**: 1001-5965(2009)10-1245-04

Improved ant colony algorithm for shortest path problem in time-dependent networks

Liu Yongqiang Chang Qing Xiong Huagang

(School of Electronics and Information Engineering, Beijing University of Aeronautics and Astronautics, Beijing 100191, China)

Abstract: An improved pheromone update strategy suitable for time-dependent networks was proposed. Under this strategy, the residual pheromone of each road can accurately reflect the change of weighted value of each road. An improved selection strategy between adjacent cities was used to compute the cities' transfer probabilities, as a result, the calculation of algorithm was greatly reduced, and the compute speed was greatly increased. To avoid the algorithm converging to the local optimal result, the ant colony algorithm was combined with genetic algorithm. In this way, the solutions after each traversal were used as the initial species to carry out single-point crossover. An improved ant colony algorithm for the shortest path problem in time-dependent networks based on these improved strategies was presented. The results of experiment show that the improved algorithm has greater probability to get the global optimal solution, and the convergence rate of algorithm is better than traditional ant colony algorithm.

Key words: time-dependent networks; shortest path problem; ant colony algorithm (ACO)

最短路径算法在许多研究和应用领域中起着重要作用. 现有的文献中提出了大量求解最短路径问题的方法和算法,但这些研究大部分针对固定权值的网络,没有考虑权值随时间变化等时变情况. 在实际网络中,如通信系统、智能交通系统等应用领域中,边的权值往往是随时间变化的. 因此,研究时变网络中最短路径的求解问题更加具有实用价值.

文献[1]中首次指出传统的最短路径算法不能适用于网络中边属性值为时变随机的情况,必须建立严格的一致性约束条件才能求解某一类动态网络中的最短路径问题. 文献[2]提出了一种动态网络模型,并用改进的 Dijkstra 算法进行求解,但时间复杂度很高. 文献[3]采用遗传算法求解动态网络中的最短路径问题,但遗传算法容易过早地收敛,使搜索陷入局部最优解. 文献[4]采

收稿日期: 2008-10-15

基金项目: 国家自然科学基金资助项目(60879024, 60872062)

作者简介: 刘永强(1978 -),男,辽宁本溪人,博士生, yongqiangliu@sina.com.

用反馈神经网络进行求解,但要求神经网络的权值矩阵必须对称,而且要求边的走行时间函数必须是连续有界的. 文献[5]提出将传统蚁群算法应用于时变网络,但传统蚁群算法的信息素更新策略不能正确反映时变网络中边上权值的时变情况及路径选择情况,算法效率较低.

本文提出一种适用于时变网络的蚁群算法信息素更新策略,使边上残留信息素能够如实正确地反映时变网络中边上权值的变化情况. 在此基础上提出时变网络中求解最短路径问题的改进蚁群算法(TDNACO, Time-Dependent Networks Ant Colony Algorithm). 此算法在离散和连续的网络中都有效,不仅能快速地收敛,而且搜索到全局最优解的概率大大提高.

1 时变网络模型

1.1 模型介绍[2]

通常将动态网络模型表示为 (V, A, T),其中 $V=(v_1, v_2, \cdots, v_n)$是有限点集;$A$是有限边集 $A\in VxV$; $T=\{g_{i,j}(t)\mid (v_i, v_j)\in A, t\in[t_0, t_n']\}$是对于每一条边 (v_i, v_j)在时间 $[t_0, t_n']\in\mathbf{R}$区间上定义的函数. 其中,$g_{i,j}(t)$为非负实数,表示从节点 v_i 在时刻 t出发到达节点 v_j所需的时间;$[t_0, t_n']$表示时间的上下界,$\forall t\in[t_0, t_n'], t+g_{i,j}(t)\in[t_0, t_n']$,$t_0$是网络节点的最早出发时间,$t_n'$是时间区间的端点.

1.2 动态网络的最短路径问题模型[2]

模型描述为

$$\min\sum_{i=1}^{n-1}T_{v_i'v_{i+1}'} \tag{1}$$

$$T_{v_k'v_{k+1}'}=g_{v_k'v_{k+1}'}\left(\sum_{i=1}^{k-1}T_{v_i'v_{i+1}'}+t_s\right) \tag{2}$$

$$(v_1', v_2'), \cdots, (v_{n-1}', v_n')\in A \tag{3}$$

$$v_i'\in V \quad i=1, 2, \cdots, n \tag{4}$$

$$0<k<n \quad k\in\mathbf{N} \tag{5}$$

其中,v_0为出发点;v_n为目的地;t_s为出发时间. 式(2)表示在 t_s出发,由路径 $(v_1', v_2', \cdots, v_n')$从节点 v_k到节点 v_{k+1}'所需的时间. 约束条件式(3)、式(4)保证了路径 $(v_1', v_2', \cdots, v_n')$在网络上是连通的,不存在断路.

2 基本蚁群系统模型

蚁群算法(ACO, Ant Colony Algorithm)是一种新的启发式算法,作为新的仿生类进化算法是由 Dorigo[6]首先提出的. 该算法模仿蚂蚁觅食时的行为,按照启发式思想,通过外激素的诱发作用逐渐收敛到问题的全局最优解.

设 m是蚁群中蚂蚁的数量,$d_{ij}(i, j=1, 2, \cdots, n)$表示城市 i和城市 j之间的距离;$\tau_{ij}(t)$表示 t时刻在城市 i, j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 $k(k=1, 2, \cdots, m)$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由城市 i转移到城市 j的概率:

$$p_{ij}^k(t)=\begin{cases}\dfrac{[\tau_{ij}(t)]^\alpha[\eta_{ij}]^\beta}{\sum\limits_{l\notin U_k}[\tau_{il}(t)]^\alpha[\eta_{il}]^\beta} & j\notin U_k\\ 0 & j\in U_k\end{cases} \tag{6}$$

其中,η_{ij}为先验知识或称为能见度,在最短路径问题中为城市 i转移到城市 j的启发信息,一般取 $\eta_{ij}=1/d_{ij}$;α为在路径 ij上残留信息的重要程度;β为启发信息的重要程度;$U_k(k=1, 2, \cdots, m)$是禁忌表,用以记录蚂蚁 k当前所走过的城市,下一步计算时不再允许选择. 集合 U_k随着进化过程作动态调整.

经过 n个时刻,所有蚂蚁都完成了一次周游. 它们本次周游的禁忌表将满,此时应清空,将当前蚂蚁所在城市置入 U_k,准备下一次周游. 这时,计算每一只蚂蚁所走过的路径 L_k,并保存最短路径:

$$L_{k_{\min}}=\min(L_k) \quad k=1, \cdots, m \tag{7}$$

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,蚂蚁完成一次循环以后,各路径上的信息量要根据式(8)作调整:

$$\tau_{ij}(t+1)=(1-\rho)\tau_{ij}(t)+\rho\Delta\tau_{ij} \tag{8}$$

$$\Delta\tau_{ij}=\sum_{k=1}^{m}\Delta\tau_{ij}^k \tag{9}$$

$$\Delta\tau_{ij}^k=\begin{cases}\dfrac{Q}{L_k} & \text{当第 } k\text{只蚂蚁在时刻 } t\text{和 } t+1\text{之间经过 } ij\text{时}\\ 0 & \text{否则}\end{cases} \tag{10}$$

其中,$\Delta\tau_{ij}^k$表示第 k只蚂蚁在本次循环中留在路径 ij上的信息量;$\Delta\tau_{ij}$表示本次循环中路径 ij上的信息量的增量;Q为常数;L_k表示第 k只蚂蚁在本次循环中所走过的路径的长度.

一般设置周游次数计数器 N_C,当达到设定值时结束. 最短路径为

$$L_{\min}=\min(L_{k_{\min(l)}}) \quad l=1, 2, \cdots, N_C \tag{11}$$

3 蚁群算法改进策略

3.1 相邻城市选择策略

在基本的蚁群算法中,当蚂蚁从城市 i选择

下一个城市 j时,需要将 $n-1$个城市与禁忌表比较,再计算 $n-1$个城市的转移概率,需要较长的计算时间.

在实际应用中,蚂蚁一定是在一个具有拓扑关系的网络中爬行,蚂蚁从城市 i选择下一个城市时,不可能是在所有非禁忌表中的城市间选择,而只能是选择与城市 i之间存在直接路径的城市,即存在边 (v_i, v_j)的节点中选择. 因此,在计算转移概率时,仅需要计算这部分城市的转移概率即可. 从而大大降低了基本蚁群算法的计算量,提高算法计算速度.

3.2 信息素动态更新策略

蚁群算法中的核心点有两个,一个是城市间转移概率的计算策略,一个是信息素更新策略;而转移概率的计算也依赖于信息素的大小. 因而在蚁群算法中,信息素更新策略至关重要,将决定算法的收敛速度.

在基本蚁群算法中,通常采用式(8)来更新信息素. 这种信息素更新策略基本保证了在蚂蚁完成一次遍历之后在边上权值不变的情况下,在蚂蚁选择的路径上产生与这次选择相对应的信息素浓度. 在时变网络中,边上的权值是时间的函数,随时间的变化而产生变化. 因此,传统的信息素更新策略将不能正确反映与路径选择相对应的信息素情况. 虽然蚁群算法有一定的适应性,在经过若干次遍历后,仍能产生最优解,但无疑将大大增加算法的计算时间.

此处给出一种在时变网络中蚁群算法的信息素更新策略,使其能正确地反映边上权值的变化情况,同时也能反映本次最优解的路径选择.

$$\tau_{ij}(t+1) = [(1-\rho)\tau_{ij}(t) + \rho\Delta\tau_{ij}] \cdot \lambda_{ij} \tag{12}$$

$$\lambda_{ij} = g_{ij}^{\text{old}} / g_{ij}^{\text{new}} \tag{13}$$

g_{ij}^{old}和 g_{ij}^{new}分别表示 (i, j)上的原行走时间及更新后的行走时间.

$$\Delta\tau_{ij} = \sum_{k=1}^{m}\Delta\tau_{ij}^{k} \tag{14}$$

$$\Delta\tau_{ij}^{k} = \begin{cases} \dfrac{Q}{L_k} & \text{当第 } k\text{只蚂蚁在时刻 } t\text{和 } t+1\text{之间经过 } ij\text{时} \\ 0 & \text{否则} \end{cases} \tag{15}$$

通过这种信息素更新策略,使蚂蚁每遍历一次之后在路径上产生的信息素不仅能反映本次路径选择的情况,同时也能够反映当前时刻最优路径上各边的权值变化情况.

3.3 最优个体交叉策略

基本的蚁群算法容易过早收敛,使搜索陷入局部最优解. 为了避免此种情况,将蚁群算法和遗传算法结合,将蚁群算法每次遍历后形成的解作为初始群种,之后扫描双亲,查看双亲中是否有相同的基因,如有,则在所有相同基因对中挑选出单点交叉后所得到的后代最优的基因对,来进行单点交叉. 采用这种方法进行交叉,不会产生非可行解;同时使得交叉操作具有了方向性,不仅能够加速收敛到全局最优解,而且在一定程度上避免陷入局部最优解.

4 改进的蚁群算法步骤

根据如上改进策略,本文给出了适用于时变网络的蚁群算法 TDNACO,算法步骤如下.

1) 算法开始运行时,按如下公式初始化每条边上的信息量:

$$\tau_{ij}(0) = C\ (C\text{为常数}) \tag{16}$$

同时设置迭代次数;

2) 将 m只蚂蚁放在初始节点 s;

3) 根据下式计算蚂蚁转移概率:

$$p_{ij}^{k}(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{l\in W_k}[\tau_{il}(t)]^{\alpha}[\eta_{il}]^{\beta}} & j\in W_k \\ 0 & \text{else} \end{cases} \tag{17}$$

其中,$U_k\ (k=1, 2, \cdots, m)$用以记录蚂蚁 k当前所走过的城市;$W_k = \{$和 i相连的节点 $- U_k\}$;$\eta_{ij} = g_{ij}(t)^{-1}$,$g_{ij}(t)$为 t时刻出发时边 (v_i, v_j)的行走时间;α, β体现信息素和启发信息对蚂蚁决策.

4) 如果 j不是目的节点,继续查找下一个节点,转 2);如果 j是目的节点,蚂蚁 k搜索到一条从出发点 s到目的地的一条路径,保存路径及其花费时间 L_k.

5) 当 m只蚂蚁搜索完后,则求得 m条路径,全局更新边上的信息量:

$$\tau_{ij}(t+1) = [(1-\rho)\tau_{ij}(t) + \rho\Delta\tau_{ij}] \cdot \lambda_{ij} \tag{18}$$

$$\lambda_{ij} = g_{ij}^{\text{old}} / g_{ij}^{\text{new}} \tag{19}$$

g_{ij}^{old}和 g_{ij}^{new}分别表示 (i, j)上的原行走时间及更新后的行走时间.

$$\Delta\tau_{ij} = \sum_{k=1}^{m}\Delta\tau_{ij}^{k} \tag{20}$$

$$\Delta\tau_{ij}^{k} = \begin{cases} \dfrac{Q}{L_k} & \text{当第 } k\text{只蚂蚁在时刻 } t\text{和 } t+1\text{之间经过 } ij\text{时} \\ 0 & \text{否则} \end{cases} \tag{21}$$

ρ表示信息素的消逝程度,$0<\rho<1$;$\Delta\tau_{ij}^{k}$是蚂蚁 k在边 (v_i, v_j)上的信息素增量;Q为常数;L_k为蚂

蚁 k从出发点到目的地所经历的时间.

6) 扫描得到的 m 个解, 进行交叉计算;计算杂交后的路径时间,若小于杂交前的路径时间,则用新路径取代原路径,同时更新相应路径上每条边的信息素,否则不进行更新.

7) 按式 (22)在 m 个解中求取局部最优解.

$$L_{k_{\min}} = \min(L_k) \quad k = 1, \cdots, m \tag{22}$$

8) 到当前迭代次数为止,所建立的所有局部最优解中值最小的解作为当前迭代次数的全局最优解.

$$L_{\min} = \min(L_{k_{\min(l)}}) \quad l = 1, 2, \cdots, \text{当前迭代次数} \tag{23}$$

9) 如果到达设置的迭代次数, 则退出程序; 否则迭代次数加 1,转步骤 2).

5 仿真结果

为验证算法,建立了 40个城市的拓扑网络,分别采用本文提出的 TDNACO算法和 ACO算法进行计算.为模拟实际时变网络中权值的变化情况,在程序中建立独立线程,在计算过程中根据设定的权值变化频率和权值变化系数用程序来随机挑选并修改路网中部分边的权值.设置 8只蚂蚁同时进行计算,取系数 $\alpha = 2, \beta = 2, \rho = 0.3, Q = 1$,分别取权值变化系数为 1, 2, 5,采用 TDNACO算法和 ACO算法分别计算 200次,结果如表 1所示.

表 1 仿真结果

仿真试验比较项目	仿真试验一		仿真试验二		仿真试验三	
	TDNACO	ACO	TDNACO	ACO	TDNACO	ACO
权值变化系数	1	1	2	2	5	5
权值变化频率 /ms	5	5	5	5	5	5
最优解概率 /%	100	91	100	88	98	83
平均求解时间 /ms	59.06	99.74	66.8	126.5	72	198.08

仿真结果表明,当边上权值变化较大及权值变化频率较高时, ACO算法的收敛速度远低于 TDNACO算法,且得到最优解的概率较低.而 TDNACO算法由于信息素更新策略和边上权值实时变化情况相关,因而收敛速度和得到最优解的概率均明显高于 ACO算法.

6 结 论

本文给出的时变网络中蚁群算法的信息素更新策略,使边上残留信息素能够如实正确地反映时变网络中边上权值的变化情况,同时也能真实反映本次最优解的路径选择情况.在此基础上提出的改进蚁群算法能够有效求解时变网络中最短路径问题.仿真结果表明,在迭代次数相同时 TDNACO比 ACO算法得到最优解的概率更大,且 TDNACO的收敛速度要高于 ACO算法. TDNACO算法可应用于智能交通系统,计算机网络通信等众多领域,相关工作还需要进一步深入研究.

参考文献 (References)

[1] Hall R W. The fastest path through a network with random time-dependent travel time [J]. Transportation Science, 1986, 20(3): 182 - 188

[2] 谭国真,高文. 时间依赖网络中最小时间路径算法 [J]. 计算机学报, 2002, 25(2): 165 - 172
Tan Guozhen, Gao Wen. Shortest path algorithm in time-dependent networks[J]. Chinese Journal of Computers, 2002, 25(2): 165 - 172 (in Chinese)

[3] 刘坚强,刘粉林. 动态网络最佳路径的遗传算法求解 [J]. 信息工程大学学报, 2004, 5(3): 14 - 18
Liu Jianqiang, Liu Fenlin. The optimum route algorithm for dynamic network based on genetic algorithms[J]. Journal of Information Engineering University, 2004, 5(3): 14 - 18 (in Chinese)

[4] He Hong, Zhu Daming, Ma Shaohan. A new algorithm for the shortest path compution by neural networks on time-dependent Networks[J]. Journal of Fudan University (Natural Science), 2004, 43(5): 714 - 716

[5] 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, 29(5): 81 - 82
Chen Lichao, Liu Jia, Lü Yanan. An ant colony algorithm with crossover operators for the shortest path problem in dynamic networks[J]. Computer Engineering & Science, 2007, 29(5): 81 - 82 (in Chinese)

[6] Dorigo M, Caro G D, Gambardella L M. Ant algorithms for discrete optimization [J]. Artificial Life, 1999, 5(2): 137 - 172

(编 辑:娄 嘉)