

DOI: 10.13718/j.cnki.xdzk.2014.09.029

基于多头绒泡菌网络模型的蚁群算法优化^①

刘玉欣, 张自力, 高超, 钱涛, 吴雨横

西南大学 智能软件与软件工程重点实验室, 重庆 400715

摘要: 基于多头绒泡菌网络模型在求解迷宫问题时所展现的“重点管道重点培养”特性, 设计一种优化的蚁群算法. 该优化算法在更新信息素矩阵时考虑蚂蚁释放的信息素和多头绒泡菌网络中流通的信息素. 通过对重点管道信息素浓度的加强, 提高路径寻优过程中重点管道的被选概率, 从而提高蚁群算法对最优解的开发力度. 针对旅行商问题的对比实验验证了该优化算法可提高传统蚁群算法的寻优能力, 并具有更高的鲁棒性.

关键词: 多头绒泡菌网络模型; 蚁群算法; 旅行商问题

中图分类号: TP301.6 **文献标志码:** A **文章编号:** 1673-9868(2014)9-0182-06

蚁群算法(ant colony optimization, ACO)最早由 Dorigo 等人于 1991 年提出, 称为蚂蚁系统(ant system, AS), 并成功应用于求解旅行商问题(traveling salesman problem, TSP)^[1]. 随后, 研究者们提出了多种改进算法^[2-5], 如蚁群系统(ant colony system, ACS)^[2]、最大-最小蚂蚁系统(max-min ant system, MMAS)^[3]和最优-最差蚂蚁系统(best-worst ant system, BWAS)^[4]. 然而在蚁群算法运算初期, 由于非全局最优路径的干扰, 会出现早熟、停滞及陷入局部最优解等现象^[6]. 本文基于多头绒泡菌网络(physarum network, PN)模型在生成高效网络过程中体现的“重点管道重点培养”特性^[7-8], 提出一种优化的蚁群算法(PNACO 算法). PNACO 算法引入一种直接的优化信息素矩阵更新策略, 通过额外叠加多头绒泡菌网络中流通的信息素, 增强“重点管道”的被选概率, 从而获得更优的最短路径, 并提高算法鲁棒性.

1 基于蚁群算法的 TSP 问题求解

TSP 问题的目标是寻找一条经过所有城市一次且仅一次并最终返回起点的最短哈密尔顿回路. 设有 n 个城市 $V = \{i \mid i = 1, 2, \dots, n\}$, m 只蚂蚁分别从各城市出发, 每只蚂蚁具有记忆能力, 对于自己已经访问过的城市将不会再次访问. t 时刻蚂蚁 k 在城市 i 选择未访问城市 j 的概率为 $P_{ij}^k(t)$, 如公式(1)^[1-3]所示, 式中 $\tau_{ij}(t)$ 表示 t 时刻连接城市 i 和城市 j 路径上的信息素浓度, N_i 表示所有未访问城市的集合, $\eta_{ij}(t) = \frac{1}{d_{ij}}$ 表示从城市 i 移动到城市 j 的期望程度, d_{ij} 表示城市 i 与城市 j 之间的距离.

$$P_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}^a(t) \times \eta_{ij}^b(t)}{\sum \tau_{iu}^a(t) \times \eta_{iu}^b(t)} & j, u \in N_i \\ 0 & j, u \notin N_i \end{cases} \quad (1)$$

① 收稿日期: 2013-02-02

基金项目: 国家科技支撑计划资助项目(2012BAD35B08); 重庆市自然科学基金资助项目(cstc2012jjA40013, cstc2013jcyjA40022); 高等学校博士学科点专项科研基金新教师基金资助(20120182120016); 中央高校基本科研业务费专项资金基金资助(XDJK2012B016, XDJK2012C018, XDJK2013D017).

作者简介: 刘玉欣(1990-), 女, 山东聊城人, 硕士生, 主要从事智能软件工程的研究.

通信作者: 张自力, 教授, 博士生导师.

在蚂蚁完成对所有城市访问后,各路径的信息素浓度根据公式(2)^[1-3]进行更新调整.

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t) \quad (2)$$

其中: $\rho(0 < \rho < 1)$ 代表信息素的挥发率. $\Delta\tau_{ij}(t)$ 的取值由具体算法确定;如AS算法采用公式(3)^[1],式中 F 为常数,表示每只蚂蚁释放的信息素量, $S_k(t)$ 为 t 时刻蚂蚁 k 走完全程的路径长度;ACS算法采用公式(4)^[2],式中 $S_{\text{best}}(t)$ 表示 t 时刻蚂蚁行走的最短路径长度;MMAS算法采用公式(5)^[3]更新信息素矩阵.

$$\Delta\tau_{ij}(t) = \sum_{k=1}^m \frac{F}{S_k(t)} \quad (3)$$

$$\Delta\tau_{ij}(t) = \frac{\rho}{S_{\text{best}}(t)} \quad (4)$$

$$\Delta\tau_{ij}(t) = \frac{F}{S_{\text{best}}(t)} \quad (5)$$

经过多次信息素矩阵更新迭代后,算法逐渐收敛至一条最短哈密尔顿回路 Ω ,也即TSP问题的最短路径.用最优解 S_{min} 表示最短路径长度,取值如公式(6)所示, Ω_i 表示哈密尔顿回路序列中的第 i 个城市,且 $\Omega_i \in V$.

$$S_{\text{min}} = \min(\sum_{i=1}^{n-1} d_{\Omega_i \Omega_{i+1}} + d_{\Omega_n \Omega_1}) \quad (6)$$

本文采用以下评价参数对比优化前后算法的寻优能力和鲁棒性:

1) 最优解 S_{min} 以及获得最优解的平均迭代步数 $\text{Steps}_{\text{average}}$.

2) 平均值 S_{average} 和方差 S_{variance} :算法经过 C 次重复计算后,所得最优解的平均值和方差,例如 S_{average} 可

定义为 $S_{\text{average}} = \frac{\sum_{i=1}^C S_{\text{min}}^i}{C}$,其中 S_{min}^i 表示第 i 次运算所得的最优解.

2 基于PN模型优化蚁群算法

本节首先介绍基本的PN模型以及对PN模型的改进,其次提出基于改进PN模型的蚁群算法优化基本思想.

2.1 PN模型

基本的PN模型用于求解两点间最短路径,所以又称为单入口单出口PN模型,文献[8]给出了其求解过程.如图1所示,设网络中的边为有水流量的管道, N_{in} 表示注水口, N_{out} 表示出水口.连接节点 i 和节点 j 的管道传导性 D_{ij} 与管道粗细有关,管道越粗,传导性越高,管道中水流流量 Q_{ij} 越大.传导性和流量的关系用公式(7)表示,其中 p_i 表示节点 i 的压强, L_{ij} 表示管道 (i,j) 的长度.

$$Q_{ij} = \frac{D_{ij}}{L_{ij}}(p_i - p_j) \quad (7)$$

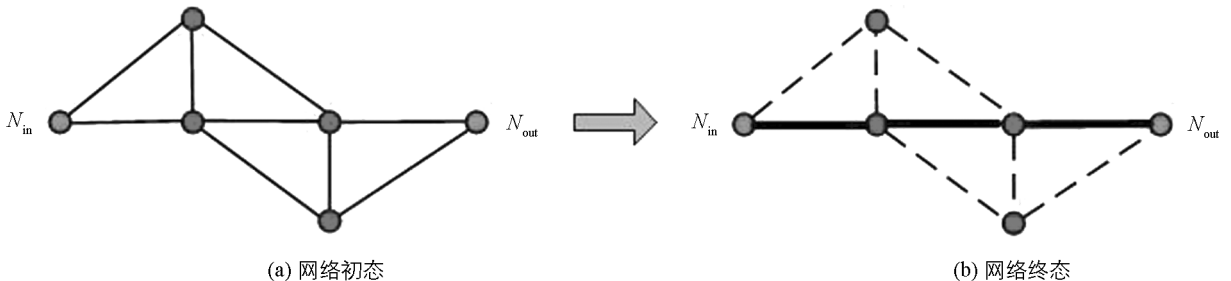


图1 单入口单出口PN模型

设初始流量为 I_0 .根据基尔霍夫定律,网络中流入和流出同一个节点的流量相等,因此由公式(8)可计算网络中各节点压强.

$$\sum_i Q_{ij} = \begin{cases} -I_0 & \text{如果 } j = N_{ij} \\ I_0 & \text{如果 } j = N_{out} \\ 0 & \text{其余} \end{cases} \quad (8)$$

然后, 利用公式(7)可计算出此时网络中各管道流量. 在水流不断注入的过程中, $t+1$ 时刻管道传导性将按公式(9)变化. 如此完成一个计算时间步, 并将 $t+1$ 时刻各管道传导性反馈给公式(8), 再次循环迭代执行上述流程.

$$\frac{d}{dt} D_{ij} = \frac{|Q_{ij}|}{1 + |Q_{ij}|} - D_{ij} \quad (9)$$

如果把迭代过程视为“培养”管道的过程, 发现某些管道在“培养”过程中慢慢变粗, 传导性逐渐增强, 这种培养模式称为“重点培养”. 当 $t+1$ 时刻的传导性 $D_{ij}(t+1)$ 和 t 时刻的传导性 $D_{ij}(t)$ 满足

$$|D_{ij}(t+1) - D_{ij}(t)| \leq 10^{-6}$$

时, 视为迭代结束, 得到最终网络. 此时, 留下的位于最短路径上的管道称为“重点管道”, 如图 1(b) 中实线所示.

然而, 在求解 TSP 问题时, 仅用两个点分别作为入口和出口并不能解释整个网络的流量流通情况, 所以首先将单入口单出口 PN 模型改进为多入口多出口 PN 模型. 具体修改如下. 设网络中管道条数为 M . 在 t 时刻, 将网络中每一条管道两端节点分别作为入口和出口, 利用公式(7)和公式(8)计算各节点压强及管道流量 $Q_{ij}^{(m)}$, 初始流量定义为 $\frac{I_0}{M}$. 然后利用公式(10)求得管道中平均流量 $\overline{Q_{ij}}$, 将其代入公式(9)计算 $t+1$ 时刻各管道的传导性

$$\overline{Q_{ij}} = \frac{1}{M} \sum_{m=1}^M |Q_{ij}^{(m)}| \quad (10)$$

2.2 PNACO 算法信息素矩阵更新策略

在 PNACO 算法中, 假设多头绒泡菌网络中流通的是信息素, 为综合考虑其对蚁群算法的影响, 我们在蚁群算法更新信息素矩阵时额外添加多头绒泡菌网络中流通的信息素, 该策略称为多头绒泡菌网络信息素矩阵更新策略(以下简称 PN 策略). 以第 1 节介绍的 3 种蚁群算法为例, 其信息素矩阵更新公式中 $\Delta \tau_{ij}(t)$ 的取值由公式(3)–(5)分别优化为公式(11)–(13).

$$\Delta \tau_{ij}(t) = \sum_{k=1}^m \frac{F}{S_k(t)} + \epsilon \frac{\overline{Q_{ij}(t)} \times M}{F} \quad (11)$$

$$\Delta \tau_{ij}(t) = \rho \left[\frac{1}{S_{best}(t)} + \epsilon \frac{\overline{Q_{ij}(t)} \times M}{F} \right] \quad (12)$$

$$\Delta \tau_{ij}(t) = \frac{F}{S_{best}(t)} + \epsilon \frac{\overline{Q_{ij}(t)} \times M}{F} \quad (13)$$

其中:

$$\epsilon = 1 - \frac{1}{1 + \lambda \frac{\text{tempsteps} - (t+1)}{2}} \quad (14)$$

表示多头绒泡菌网络中流通的信息素对信息素总量的影响因子; $\overline{Q_{ij}(t)}$ 表示 t 时刻管道 (i, j) 流通的平均信息素浓度; tempsteps 表示算法运算过程中 PN 策略影响的总步数. 根据文献[9]对 PN 模型的数学分析可知, PN 模型演化速率较快, 在 100 ~ 300 迭代步之间即演化结束, 因此 tempsteps 取 [100, 300] 为宜.

2.3 PNACO 算法描述

如图 2 所示, 对蚁群算法分别应用 PN 策略进行优化, 优化后的算法统称为 PNACO 算法, 具体描述见算法 1.

算法 1(PNACO 算法)

步骤 1: 初始化信息素矩阵及管道传导性矩阵为全 1 矩阵, 迭代计数器 $N=0$;

步骤 2: 将 m 只蚂蚁放置在各城市, 并按公式(1)计算移动概率, 选择下一个未经过的城市, 直至返回起点;

- 步骤 3: 比较所有蚂蚁走过的路径长度, 记录其中最优解 $S_{\min}$;
- 步骤 4: 利用公式(7)、公式(8)和公式(10)得到多头绒泡菌网络中各管道平均信息素浓度, 并使用公式(9)计算下一时刻管道传导性;
- 步骤 5: 利用公式(11)(或公式(12)或公式(13))和公式(2)更新信息素矩阵, 且 $N=N+1$;
- 步骤 6: 如果 $N<\text{totalsteps}$ (totalsteps 表示总迭代步数), 转步骤 2;
- 步骤 7: 输出最优解 $S_{\min}$.

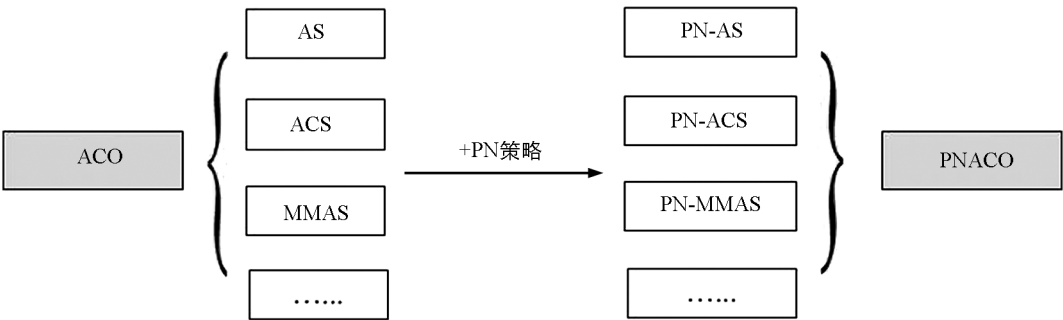


图 2 ACO 算法与 PNACO 算法关系图

3 实验验证及分析

为验证 PN 策略的有效性, 对 ACO 算法中经典的蚁群系统(ACS)^[2]、最大-最小蚂蚁系统(MMAS)^[3]以及文献[5]中改进的突变蚁群系统(M-ACS)和突变最大-最小蚂蚁系统(M-MMAS)添加 PN 策略得到 PN-ACS 算法、PN-MMAS 算法、PN-M-ACS 算法以及 PN-M-MMAS 算法. 以 TSPLIB 中 4 组标准数据集为基础, 通过第 1 节介绍的 4 种评价参数衡量 ACO 算法与 PNACO 算法在求解 TSP 问题时的寻优能力及鲁棒性.

ACO 算法和 PNACO 算法的基本参数设置为: $\alpha=1$, $\beta=2$, $F=20$, $m=n$, $\lambda=1.05$, tempsteps=300. 每个算法实例重复运算 20 次, 统计结果如表 1 所示. 实验中 M-ACS 算法、PN-M-ACS 算法、M-MMAS 算法和 PN-M-MMAS 算法中突变城市个数均设置为 2 个, 同时由于突变操作的随机性, 无法准确衡量该类算法运行结果的方差和获得最优解的平均迭代步数.

表 1 标准 TSP 数据集运行结果

数据集	算 法	参数配置	$S_{\min}$	S_{average}	S_{variance}	$\text{Steps}_{\text{average}}$
gr24	ACS	$\rho=0.7$	1 330.00	1 402.50	65.85	40.60
	PN-ACS	totalsteps=300	1 272.00	1 280.00	8.69	164.00
	M-ACS	$\rho=0.7$	1 272.00	1 275.60	—	—
	PN-M-ACS	totalsteps=300	1 272.00	1 273.20	—	—
	MMAS	$\rho=0.7$	1 272.00	1 300.60	27.31	155.20
	PN-MMAS	totalsteps=300	1 272.00	1 296.90	18.27	283.70
	M-MMAS	$\rho=0.7$	1 272.00	1 293.00	—	—
	PN-M-MMAS	totalsteps=300	1 272.00	1 288.55	—	—
gr48	ACS	$\rho=0.7$	5 236.00	5 332.90	69.39	196.60
	PN-ACS	totalsteps=500	5 221.00	5 310.50	44.98	367.00
	M-ACS	$\rho=0.7$	5 122.00	5 303.70	—	—
	PN-M-ACS	totalsteps=500	5 095.00	5 297.30	—	—
	MMAS	$\rho=0.7$	5 104.00	5 204.50	62.82	409.40
	PN-MMAS	totalsteps=500	5 083.00	5 191.65	61.33	365.00
	M-MMAS	$\rho=0.7$	5 076.00	5 211.70	—	—
	PN-M-MMAS	totalsteps=500	5 063.00	5 177.85	—	—

数据集	算 法	参数配置	$S_{\min}$	S_{average}	S_{variance}	$\text{Steps}_{\text{average}}$
eil51	ACS	$\rho=0.1$	435.39	447.82	8.13	542.90
	PN-ACS	totalsteps=1 000	433.05	440.77	3.30	476.20
	M-ACS	$\rho=0.1$	434.16	441.88	—	—
	PN-M-ACS	totalsteps=1 000	431.75	441.26	—	—
	MMAS	$\rho=0.1$	431.96	440.95	6.39	710.15
	PN-MMAS	totalsteps=1 000	430.86	439.17	5.54	748.90
	M-MMAS	$\rho=0.1$	430.75	439.35	—	—
	PN-M-MMAS	totalsteps=1 000	429.48	437.89	—	—
eil76	ACS	$\rho=0.1$	559.87	568.29	5.26	731.10
	PN-ACS	totalsteps=1 000	555.85	566.07	5.09	797.50
	M-ACS	$\rho=0.1$	556.64	568.61	—	—
	PN-M-ACS	totalsteps=1 000	554.80	564.37	—	—
	MMAS	$\rho=0.1$	558.27	571.66	8.67	834.50
	PN-MMAS	totalsteps=1 000	556.87	567.54	6.88	757.85
	M-MMAS	$\rho=0.1$	556.80	569.40	—	—
	PN-M-MMAS	totalsteps=1 000	553.39	568.31	—	—

表 1 实验结果显示, PNACO 算法的最优解 $S_{\min}$ 和平均值 S_{average} 均优于相应的 ACO 算法, 而获得最优解的平均迭代步数 $\text{Steps}_{\text{average}}$ 略大于 ACO 算法, 说明 PN 策略有效避免了 ACO 算法早熟及收敛至局部最优的缺点, 使优化后算法的求解结果更优. 同时 PNACO 算法的方差 S_{variance} 小于相应的 ACO 算法, 说明 PNACO 算法具有较高的鲁棒性.

4 总 结

传统蚁群算法的优化大多通过提高精英蚂蚁的信息素贡献率来提高算法对最优解的探索力度, 但并不能避免在迭代运算初期非最优路径对最终解的影响. 本文利用 PN 模型“重点管道重点培养”的特性, 设计了 PNACO 算法, 通过对“重点管道”信息素浓度的加强, 提高“重点管道”(即最优解)的被选概率, 从而提高蚂蚁对最优解的开发力度. 通过 TSP 问题标准数据集测试, 验证了 PNACO 算法明显优于 ACO 算法, 不仅能获得更优的最短路径, 而且具有更高的鲁棒性. PN 模型自提出以来大多应用于复杂网络分析与设计^[10-12]. 本文首次提出将 PN 模型与蚁群算法结合求解组合优化问题, 这是 PN 模型在应用领域上的又一次拓展, 我们的研究成果将为 PN 模型的发展与应用提供新思路.

参考文献:

[1] COLORNI A, DORIGO M, MANIEZZO V. Distributed Optimization by Ant Colonies [C] // Proceedings of the First European Conference on Artificial Life. Massachusetts: MIT Press, 1991, 142: 134-142.

[2] DORIGO M L, GAMBARDELLA M. Ant Colony System: a Cooperative Learning Approach to the Traveling Salesman Problem [J]. IEEE Transactions on Evolutionary Computation, 1997, 1(1): 53-66.

[3] STUTZLE T, HOOS H H. Max-Min Ant System [J]. Future Generation Computer System, 2000, 16(8): 889-914.

[4] GARCÍA O C, DE VIANA I F, TRIGUERO F H. Analysis of the Best-Worst Ant System and Its Variants on the TSP [J]. Mathware & Soft Computing, 2002, 9(3): 177-192.

[5] ZHAO N, WU Z, ZHAO Y Q, et al. Ant Colony Optimization Algorithm with Mutation Mechanism and Its Applications [J]. Expert Systems with Applications, 2010, 37(7): 4805-4810.

[6] BLUM C. Ant Colony Optimization: Introduction and Recent Trends [J]. Physics of Life Reviews, 2005, 2(4): 353-373.

[7] TERO A, KOBAYASHI R, NAKAGAKI T. Physarum Solver: a Biologically Inspired Method of Road-Network Navigation [J]. Physica A: Statistical Mechanics and Its Applications, 2006, 363(1): 115-119.

[8] TERO A, KOBAYASHI R, NAKAGAKI T. A Mathematical Model for Adaptive Transport Network in Path Finding by True Slime Mold [J]. Journal of Theoretical Biology, 2007, 244(4): 553-564.

- [9] MIYAJI T, OHNISHI I. Mathematical Analysis to an Adaptive Network of the Plasmodium System [J]. Hokkaido Mathematical Journal, 2007, 36(2): 445—465.
- [10] TERO A, TAKAGI S, SAIGUSA T, et al. Rules for Biologically Inspired Adaptive Network Design [J]. Science Signaling, 2010, 327(5964): 439—442.
- [11] HOUBRAKEN M, DEMEYER S, STAESSENS D, et al. Fault Tolerant Network Design Inspired by Physarum Polycephalum [J]. Natural Computing, 2013, 12(2): 277—289.
- [12] WATANABE S, TAKAMATSU A. Transportation Network with Fluctuating Input/Output Designed by the Bio-Inspired Physarum Algorithm [J]. PloS ONE, 2014, 9(2): 1—11.

An Optimized Ant Colony Optimization Algorithm Based on the *Physarum* Network Model

LIU Yu-xin, ZHANG Zi-li, GAO Chao,
QIAN Tao, WU Yu-heng

Key Laboratory of Intelligent Software and Software Engineering, Southwest University, Chongqing 400715, China

Abstract: The *Physarum* network (PN) model exhibits a unique feature that the critical pipelines are reserved with the evolution of network during the process of solving the maze problem. In the present study, drawing on this feature, an optimized ant colony optimization (ACO) algorithm denoted as PNACO algorithm is proposed based on the PN model. When the pheromone matrix is updated, the PNACO algorithm updates both the pheromone released by ants and the flowing pheromone in the *Physarum* network. By adding extra pheromones in the *Physarum* network, the critical pipelines are the shortest routes that have a higher opportunity to be selected when ants travel cities. Hence, the exploitation of the optimal solution will be promoted. Experimental results show that the solutions of PNACO algorithm are better than those of ACO algorithm for solving the traveling salesman problem (TSP). In addition, the PNACO algorithm is more robust than the ACO algorithm.

Key words: *Physarum* network model; ant colony optimization (ACO) algorithm; traveling salesman problem

责任编辑 张 枸

