

基于改进蚁群算法的虚拟网络映射优化

谢永浩 高嵩峰 代明竹

(北京建筑大学机电与车辆工程学院 北京 100044)

摘要 优化了基于改进蚁群算法的虚拟网络映射结果。以最优化应用底层网络的资源,提升虚拟网络映射底层网络的资源利用效率为研究目标,在不需要支持路径分裂底层网络的情况下,提出一种新的基于改进蚁群算法的虚拟网络映射。通过引入高斯过程模型,加快蚁群优化算法的收敛速度,满足实际应用的实时性要求;并且以映射开销作为适应度函数,最终解决虚拟网络映射问题。实验结果表明,在满足相同准确度的前提下,该算法显著地降低了算法的求解时间,发挥了积极影响。

关键词 虚拟网络映射,改进蚁群算法,高斯算法

中图分类号 TP393 **文献标识码** A

Virtual Network Mapping Optimization Based on Improved Ant Colony Algorithm

XIE Yong-hao GAO Song-feng DAI Ming-zhu

(School of Mechanical-Electronic and Vehicle Engineering, Beijing University of Civil Engineering and Architecture, Beijing 100044, China)

Abstract In virtual network mapping, the virtual network mapping results based on improved ant colony algorithm were optimized. Aiming at the optimal utilization efficiency of the underlying network resources, a new virtual network mapping algorithm based on improved ant colony algorithm was proposed. By introducing the Gauss process model, the convergence speed of ant colony optimization algorithm is accelerated, and the real-time requirement of practical application is satisfied. The results show that the algorithm can significantly reduce the solution time and play a positive role on the premise of satisfying the same accuracy.

Keywords Virtual network mapping, Improved ant colony algorithm, Gauss algorithm

在网络虚拟化技术中,可以通过网络映射的方式共享物理网络基础设施,从而提升虚拟网络存储性能,优化构建多重异构的虚拟网络^[1-3]。在实际计算机领域内,虚拟化作为一种广义术语,表示在虚拟基础上的计算机元件,而不是真实运行的元件^[4]。同时,虚拟化也是一种可以验证软件质量效益的技术^[5],在同一台物理机上可以针对不同的虚拟机运行不同的操作系统,也可以同时运行多个不同的应用程序。同样,虚拟化又是一个抽象层,它将物理硬件与操作系统分开,从而提供更高的IT资源利用率和灵活性。利用虚拟基础架构可以在整个基础架构范围内共享多台计算机的物理资源^[6-9]。利用虚拟机可以在多台虚拟机之间共享单台物理计算机的资源以实现效率最高。资源在多个虚拟机和应用程序之间共享。在实际中,通过分析底层网络的可编程特性,在一个特定逻辑拓扑结构中运行虚拟网络,可以在不影响现有网络的前提下,在不同节点、拓扑结构联合链路资源约束的条件下,满足构建虚拟网络映射的请求,从而为底层网络分配相应的资源,并在网络虚拟化环境中完成虚拟网络映射操作,通过改进蚁群算法,降低问题复杂度,满足实际应用中对虚拟网络请求实时性处理的要求,发挥积极影响。

1 蚁群算法的原理

事实上,蚁群算法就是通过模拟自然界中蚁群的集体行

为而构建的一种随机搜索算法^[10]。意大利的 M. Dorigo 指出,在蚁群算法中能够通过人工模拟蚁群搜索食物的过程,来求解从蚁穴找到食物源之间的最短路径^[11]。在蚁群算法中,若在初期信息素匮乏,则算法的求解速度会变慢。在蚁群算法中,假设有 M 只蚂蚁,将其放入到 N 个不同的城市内,则蚂蚁在选择下一个城市时,可以依据 $\tau_{ij}(t)$ 解决最短路径问题。算法如下:

$$P_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha(t) \eta_{ij}^\beta(t)}{\sum_{n \in allowed_k} \tau_{in}^\alpha(t) \eta_{in}^\beta(t)}, & j \in allowed_k \\ 0, & \text{else} \end{cases} \quad (1)$$

假设,在 $j \in N_i^k$ 中, α 表示残留信息相对重要的程度; β 表示期望值相对重要的程度; N_i^k 表示可能访问的目标城市; P_{ij}^k 表示蚂蚁 k 从 i 城市访问到 j 城市的发生概率。

蚂蚁在运动过程中可以感知运动方向,从而朝着物质强度高方向移动。

2 改进蚁群算法

2.1 更新信息素

在基本蚁群算法的基础上作出如下改进:对于信息素更新策略,采取全局更新的策略。考虑到每次迭代产生的较差解对搜索全局最优解的帮助不大,此时若增加较差解路径上的信息素,反而会影响到搜索最优解的速度^[12]。因此,在每次

本文受网络型多媒体汽车实训与考核系统(31068015212)资助。

谢永浩(1990—),男,硕士生,主要研究方向为管理信息系统, E-mail: 418636274@qq.com; 高嵩峰(1977—),男,副教授,硕士生导师,主要研究方向为管理信息系统; 代明竹(1992—),女,硕士生,主要研究方向为管理信息系统。

迭代结束后适当减少这些解路径上的信息素,相应地增加较优解路径上的信息素,改成如下形式:

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{L^{ave} - L_k}{L^{worst} - L^{best}} \cdot 5 \operatorname{sgn}(L^{ave} - L_k), & (i, j) \in R_k(t) \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

其中, L_k 表示第 k 只蚂蚁在本次迭代中所走路径的总长度, L^{best} 和 L^{worst} 分别表示本次迭代中得出的最短路径与最长路径。 L^{ave} 表示所有蚂蚁在本次迭代中所走路径的平均长度。

2.2 更新信息素的改进机制

由于每次迭代结束后要对路径上的信息素进行更新,因此迭代前期的较差解所在路径上的信息素量相对较少,加上挥发系数 ρ 的存在,导致这些路径上的信息素量在一定的迭代次数后会接近甚至等于 τ_{min} ,从而降低了算法的搜索能力。因此,对每次迭代后信息素的挥发策略加以改进:

$$\tau_{ij}(t+1) = (1-\rho)^{\xi(t)} \tau_{ij}(t) + \sum_{k=1}^m \Delta\tau_{ij}^k(t) \quad (3)$$

$$\xi(t) = \sqrt{\frac{NC_{max}}{25(NC_{max}-t)+1}} \quad (4)$$

其中, NC_{max} 为最大迭代次数。

在迭代前期,信息素的保留率相对于之前得到了提高,较差解所在路径被搜索的概率得到了提升,从而扩大了算法的搜索空间^[13-15]。之后,随着迭代的进行,通过不断增大 $\xi(t)$ 的值,使得算法在搜索时更倾向于搜索到较优路径,在迭代中后期加快了算法的搜索速度。

为防止部分路径上的信息素量过大或过小甚至消失,每次更新后对各条路径上的信息素量进行限制:

$$\tau_{ij}(t+1) = \begin{cases} \tau_{max}, & \tau_{ij}(t+1) > \tau_{max} \\ \tau_{ij}(t+1), & \tau_{min} \leq \tau_{ij}(t+1) \leq \tau_{max} \\ \tau_{min}, & \tau_{ij}(t+1) < \tau_{min} \end{cases} \quad (5)$$

2.3 应用高斯过程模型

在高斯过程回归中,不需指定 $f(x)$ 的具体形式, n 个训练数据的观测值 $(y_1, \dots, y_n)$ 被认为是从某个多维(n 维)的高斯分布中采样出来的一个点(n 维),而类似的 $f(x)$ 也可以认为是从高斯过程中采样得到的一个无穷维的点。

给定训练数据 $x_1, \dots, x_n$, 其对应的函数值是 $y_1, \dots, y_n$ 。

假设对观测 t 建模为某个目标函数 $y(x)$, 加上高斯噪声:

$$t = y(x) + N(0, \beta^{-1}) \quad (6)$$

目标变量 t 的联合概率分布为:

$$P(t|y) = N(t|y, \beta^{-1}|N) \quad (7)$$

根据高斯过程的定义, $p(y)$ 的边缘分布如下,其中协方差由 K 矩阵定义。

$$P(y) = N(y|0, K) \quad (8)$$

得到 $p(t)$ 的边缘分布为:

$$P(t) = \int P(t|y)P(y)dy = N(t|0, C) \quad (9)$$

3 仿真实验

3.1 实验环境

在 Windows XP MATLAB R2012b 上运行。为了检验改进蚁群算法的具体性能,针对 27 个不同城市的路径选择问题进行实验,具体坐标如表 1 所列。

表 1 城市的坐标

城市	坐标(X,Y)	城市	坐标(X,Y)	城市	坐标(X,Y)
1	(14.5,17.5)	10	(0.5,4.5)	19	(15.5,3.5)
2	(14.5,6.2)	11	(6.0,16.0)	20	(6.0,6.0)
3	(12.0,14.0)	12	(16.0,14.0)	21	(20.0,15.5)
4	(10.0,18.0)	13	(17.8,4.0)	22	(12.5,19.0)
5	(1.5,11.5)	14	(9.0,12.0)	23	(12.5,11.0)
6	(18.0,11.5)	15	(8.0,8.0)	24	(15.5,10.0)
7	(4.0,10.0)	16	(4.0,18.0)	25	(10.5,11.0)
8	(18.0,18.0)	17	(12.5,3.0)	26	(9.0,15.0)
9	(9.7,10.0)	18	(13.5,15.0)	27	(15.5,12.0)

3.2 算法的实现步骤

Step 1 参数初始化。每条路径上的信息素浓度 $\tau_{ij}(0) = c$, c 为常数。 $NC=0$, $NC_{max} = \text{MAX}(\text{最大迭代次数})$ 。

Step 2 将 m 只蚂蚁随机放到 n 个城市中,为每只蚂蚁建立禁忌表 $tabu_k$ 并把它们当前所在的第一个城市记录在禁忌表中。

Step 3 蚂蚁从其所在的第一个城市出发,依照式(1)在 $allowed_k$ 中选择下一个城市并记录在禁忌表 $tabu_k$ 中。

Step 4 重复 Step 3,直至访问完所有城市。所有蚂蚁回到起点,这样就形成了 m 条闭合的回路。计算每只蚂蚁走过的路程,得出 L^{best} , L^{worst} 和 L^{ave} 的值并将其中的最小值 L^{best} 保存。

Step 5 按照式(4)一式(7)对每条路径上的信息素进行更新,清空所有禁忌表。

Step 6 $NC=NC+1$, $t=t+1$,重复 Step 2—Step 5,当 $NC=NC_{max}$ 时,算法停止。比较每次迭代产生的最短路径,输出其中的最优解及其所对应的路径。

3.3 实验结果

实验的各参数设置为: $\alpha=1$, $\beta=5$, $\rho=0.1$, $Q=100$, $\tau_{max}=10$, $\tau_{min}=0.01$,基本蚁群算法与本文算法中各条路径上的信息素量初始值 $\tau_{ij}(0)=1$,蚂蚁的数量 $m=20$ 。每种算法运行 20 次,每次运行迭代 250 次,运行得到的结果分别如表 2、图 1 及图 2 所示,其中平均解为 20 次运行所得的解的平均值;最优解为 20 次运行所得的解的最小值;平均耗时为各次运行得到最优解的时间的平均值。

表 2 算法性能的比较

算法	平均解	最优解	平均耗时/s
本文算法	92.2156	91.2286	1.42
基本蚁群算法	95.5680	94.3929	2.83
MMAS 算法	92.5785	91.2286	2.12

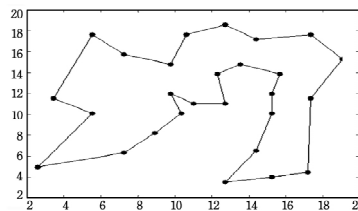


图 1 本文算法与 MMAS 算法的最优路径

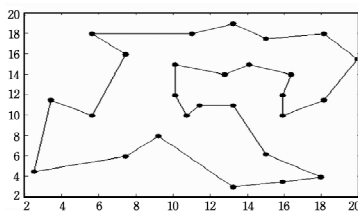


图 2 基本蚁群算法的最优路径

Com. 2011;73-84.

- [10] PACKER H S, SAMANGOOEI S, HARE J S. Event Detection using Twitter and Structured Semantic Query Expansion[C]// Proc. of ACM Workshop on Multimodal Crowd Sensing. 2012; 7-14.
- [11] LUKYANENKO R, PARSONS J. Conceptual Modeling Principles for Crowdsourcing[C]// Proc. of ACM Workshop on Multimodal Crowd Sensing. 2012; 3-6.
- [12] ZHOU P, ZHENG Y, LI M. How long to wait?: predicting bus arrival time with mobile phone based participatory sensing[C]// Proc. of ACM MobiSys. 2012; 379-392.
- [13] BALAN R K, NGUYEN K X, JIANG L. Real-time trip informa-

tion service for a large taxi fleet[C]// Proc. of ACM MobiSys. 2011; 99-112.

- [14] ZHAO D, MA H, TANG S. COUPON: cooperatively Building Sensing Maps in Mobile Opportunistic Networks[C]// Proc. of IEEE MASS. 2013; 295-303.
- [15] YANG Z, WU C, LIU Y. Locating in fingerprint space: wireless indoor localization with little human intervention[C]// Proc. of ACM MobiCom. 2012; 269-280.
- [16] GUO B, ZHANG D, WANG Z, et al. Opportunistic IoT: exploring the harmonious interaction between human and the Internet of Things[J]. Journal of Network and Computer Applications, 2013, 36(6): 1531-1539.

(上接第 313 页)

通过表 2 可以看出本文算法相对于基本蚁群算法以及 MMAS 算法在求解最优解时更具优势,具体表现在全局搜索能力与收敛速度两个方面。继续使用本文算法解决 oliver30 城市问题,参数设置:蚂蚁的数目 $m=30$,每次运行迭代 200 次;其他参数的设置与上例相同。算法运行到第 3 次时得到最优解 423.7476,得到的解进化曲线与最短路径分别如图 3 和图 4 所示。

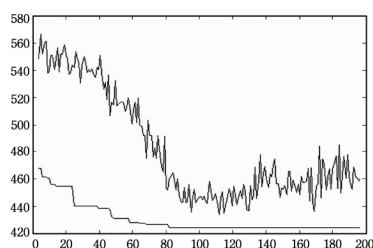


图 3 解进化曲线

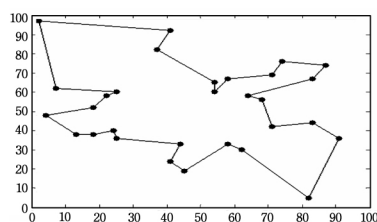


图 4 oliver30 的最优路径

3.4 结构分析

本文对基本蚁群算法进行了分析,在此基础上提出了一种新的启发式算法,具体体现在对信息素的更新规则以及挥发策略加以改进,增强了系统的适应性及鲁棒性。最后的算例实验表明,改进算法有效降低应用虚拟网络映射资源的开销。还可以在不支持路径分裂底层网络的情况下改进蚁群算法,优化虚拟网络映射,该算法根据问题模型重新定义了蚂蚁的位置,通过在高斯过程采用机器学习方法,学习蚁群寻优过程中蚂蚁信息素较好的变化方式,进行蚁群信息素的二次更新,启发式地指导蚁群运行行为,矫正蚂蚁信息素的变化方式。模拟实验结果表明,对于任意的虚拟网络资源请求,在满足同等求解精度的前提下,与已有的研究相比,改进方法降低了虚拟网络映射的求解时间,加快了求解算法的收敛速度。

结束语 本文通过改进蚁群算法,创新算法中信息素的更新策略,有效加快了算法的收敛速度,抑制了早熟现象的出现。最后通过实例证明,运用改进后的蚁群算法构建虚拟网络映射,满足实际应用需求。同时,在改进蚁群算法的过程中,还可以引入高斯过程模型,从而有助于加快蚁群算法在实际中的收敛速度,满足实际应用的实时性要求;并且以映射开销作为适应度函数,最终解决虚拟网络映射问题。

参考文献

- [1] 肖国荣. 改进蚁群算法和支持向量机的网络入侵检测[J]. 计算机工程与应用, 2014, 50(3): 75-78.
- [2] 曹文杰. 基于蚁群算法的虚拟网络映射研究[D]. 济南: 山东大学, 2015.
- [3] 张葆. 虚拟网络映射算法研究[D]. 西安: 西安电子科技大学, 2014.
- [4] 蔡进科, 顾华玺, 卢冀, 等. 基于 Openflow 网络的高可靠性虚拟网络映射算法[J]. 电子与信息学报, 2014, 36(2): 396-402.
- [5] 杨微. 低功耗片上网络映射算法研究[J]. 现代计算机, 2015(2): 10-13.
- [6] 朱强, 王慧强, 马春光, 等. 虚拟网络可生存的启发式可靠映射算法[J]. 通信学报, 2015, 36(7): 109-119.
- [7] 侯颖. 基于蚁群算法的众核嵌入式流程映射方法研究[D]. 上海: 华东师范大学, 2015.
- [8] 邱大洪. 基于混沌的蚁群算法及其应用研究[D]. 北京: 北京化工大学, 2015.
- [9] 尔雅莉. 基于蚁群算法的非结构化 P2P 资源搜索研究[D]. 太原: 太原理工大学, 2014.
- [10] 赵玉苹, 张惠珍. 带柔性时间窗车辆路径问题的混沌蚁群算法[J]. 数学理论与应用, 2016(2): 84-92.
- [11] 殷洪海. 云环境下基于改进蚁群算法的资源调度策略[D]. 成都: 电子科技大学, 2014.
- [12] 范绍聪, 刘怡俊. 基于量子蚁群算法的片上网络映射研究[J]. 计算机应用研究, 2017, 34(1): 156-159.
- [13] 李燕龙, 王俊义, 符杰林, 等. 基于改进蚁群算法的高能效 WSN 协作传输策略[J]. 电视技术, 2015, 39(11): 131-135.
- [14] 赵开新, 魏勇, 王东署. 改进蚁群算法在 P2P 网络资源搜索中的应用[J]. 火力与指挥控制, 2015, 40(5): 139-142.
- [15] 沈显庆, 崔保峰. 模拟退火改进蚁群算法的公交网络设计[J]. 黑龙江科技大学学报, 2016, 26(3): 327-331.