Computer Science and Application 计算机科学与应用, 2019, 9(10), 1823-1830
Published Online October 2019 in Hans. http://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2019.910204

Hans汉斯

Ant Colony Algorithm for Vertex Covering Problem of Weighted Graphs

Sen Sun, Jingrong Chen*, Peiwen Wu

School of Mathematics and Physics, Lanzhou Jiaotong University, Lanzhou Gansu
Email: *sa1173346987@163.com

Received: Sep. 16th, 2019; accepted: Oct. 1st, 2019; published: Oct. 8th, 2019

Abstract

Vertex covering problem is a classical NP complete problem in combinatorial optimization. At present, the problem cannot find the optimal solution in the effective polynomial time. This paper improves the early stagnation problem of ant colony algorithm. By limiting the pheromone concentration, the pheromone concentration will not continue to strengthen at the good vertex, nor will it ignore some potential search areas. This algorithm effectively avoids local optimization, improves the accuracy of the algorithm, and obtains a solution with time complexity of $O\left[(n-1)^2 \cdot n\right]$.

Keywords

Vertex Coverage, Ant Colony Algorithm, Pheromone Updating, Time Complexity

赋权图点覆盖问题的蚁群算法求解

孙　森，陈京荣*，吴佩雯

兰州交通大学，甘肃　兰州
Email: *sa1173346987@163.com

收稿日期：2019年9月16日；录用日期：2019年10月1日；发布日期：2019年10月8日

摘　要

点覆盖问题为组合优化中一个经典的NP完全问题。目前，该问题在有效的多项式时间内无法找到最优解。文章针对蚁群算法可能出现的早期停滞问题进行改进，通过对信息素浓度进行限定，信息素浓度不会在

*通讯作者。

文章引用: 孙森, 陈京荣, 吴佩雯. 赋权图点覆盖问题的蚁群算法求解[J]. 计算机科学与应用, 2019, 9(10): 1823-1830.
DOI: 10.12677/csa.2019.910204

好的顶点继续加强，也不会忽略掉潜在的一些搜索区域。该算法有效地避免了局部最优，提高了算法的准确度，得到时间复杂性为$0\left[(n-1)^2\cdot n\right]$的解决方案。

关键词

点覆盖，蚁群算法，信息素更新，时间复杂性

1. 引言

点覆盖是指给定一个无向图$G=(V,E)$，其中V为顶点集，若存在$V(G)$的一个子集S，使得G中任意一条边的两个端点至少有一个在S中，则S为G的点覆盖集。若图G中不存在点覆盖S'，使$|S'|<|S|$，则称S为G的一个最小点覆盖集，其中$|S|$和$|S'|$即集合S和S'的顶点个数[1]。

在运筹学和管理科学领域，点覆盖问题也是NP-hard最常用的问题之一[2]。在现实生活中有许多的问题都可以利用最小点覆盖问题原理进行解决。对该问题研究虽然已经有很长时间，但是人们至今没有找到一个多项式算法可以求得其精确解[3]。Niedermeier等[4]提出当每个点的权值最小为1，n个点总权值最多为k时，可以找到时间复杂度为$O(1.3788^k+kn)$的解决方案。Chvatal等[5]运用贪婪启发式算法，选取顶点权值与度之比最小的点解决集合覆盖问题。1993年参数化算法第一次用来解决该问题[6]。寇磊等[7]运用改进的Dijkstra算法解决该问题，并且得出该近似算法的时间复杂度为$O(n^3)$。王丽丽等通过线性规划松弛来实现对每个实例中参数k范围的预测，介绍了一种求解该问题实例计算成本的方法[8]。骆伟忠等将启发式操作和核心化操作相融合，给出了基于核心化技术的点覆盖问题的改进算法[9]。郝斌斌等人研究了一种基于该问题的共享单车投放点选取问题[10]。目前对于蚁群算法的研究已经不再仅仅局限于单一的范围中，近年来往往利用遗传算法结合贪婪启发式算法和基于重力的转启发式算法研究最小点覆盖问题并得到近似解。

1992年意大利学者Dorigo等人提出了蚁群算法[11]。2000年在国际顶级期刊《Nature》上发表了关于蚁群算法系统的研究综述，首次将蚁群算法推向国际领域，人们开始对蚁群算法进行广泛地研究和应用[12]。吴佩雯等人对蚁群算法解决点覆盖问题给出了简单的近似算法[13]。葛洪伟等人将SCHF启发函数与蚁群算法相结合得到了一个新的算法对集合问题进行求解[14]。

目前蚁群算法研究点覆盖问题还不完善，蚁群算法虽然具有系统性、分布式计算、自组织、正反馈等优势，但是当处理数据量巨大时其会出现求解时间长或者得不到最优解，而且蚂蚁在循环中可能会出现早期停滞问题导致算法陷入局部最优。因此为了避免上述问题的出现，引入信息素浓度限定这一概念，对信息素浓度进行限定，提出了一种改进的蚁群算法。当每个点的权值最小为1时，得到时间复杂性为$0\left[(n-1)^2\cdot n\right]$的解决方案，其中$n$为图$G$的顶点数。

2. 基本原理

蚁群算法的灵感来源于现实生活中真正蚂蚁的自然优化机制，它是一种启发式算法。蚂蚁依靠相互协作可以在洞穴和食物源之间找到一条最短路径[15]。探究发现这是由于蚂蚁在通过一条路时会留下一种

信息素，当其它蚂蚁选择路径时，会依靠信息素浓度判断并找出一条最短路径[16]。

设 m 为蚂蚁的数目，$b_i(t)$ 表示 t 时刻在节点 i 的蚂蚁数目，$m=\sum_{i=1}^{n}b_i(t)\tau_{ij}(t)$ 为某时刻 t 在路段 (i,j) 上的信息素，信息素轨迹和参数初始化，$\tau_{ij}(t_0)=C$，(C 为常数)，为使算法初期各个蚂蚁更好获得较好的解，将信息素大小设置为 $\tau_{\max}$。η_{ij} 表示路段 (i,j) 上的能见度。

计算状态转移规则，将初始节点 i 置于蚂蚁 $k(k=1,2,\cdots,m)$ 的解集 C_k 中，蚂蚁 k 按状态转移规则选择下一节点 j，并将 j 放入当前解集 C_k 中。

$$p_{ij}^{k}(t)=\begin{cases}\dfrac{\left[\tau_{ij}(t)\right]^{\alpha}\left[\eta_{ik}(t)\right]^{\beta}}{\sum_{s\in allowed_k}\left[\tau_{is}(t)\right]^{\alpha}\left[\eta_{is}(t)\right]^{\beta}}, & \text{若 } j\in allowed_k\\ 0, & \text{否则}\end{cases}\tag{1}$$

在公式(1)中 $allowed_k=\{0,1,2,\cdots,n-1\}$ 为允许访问下一节点的集合。α,β 为参数因子，由 ant-cycle 模型得出的最好的经验结果为 $0\ll\alpha\ll5$，$0\ll\beta\ll5$。

在每只蚂蚁走完一步或者完成一次环游，记录最好解 C_{best}^{t}，并进行信息素更新。

$$\tau_{ij}(t+1)=\rho\tau_{ij}(t)+\Delta\tau_{ij}^{best}\tag{2}$$

$$\Delta\tau_{ij}^{best}=\begin{cases}1/L^{best}, & \text{若}i,j\text{为最优路径}\\ 0, & \text{否则}\end{cases}\tag{3}$$

式(3)中 ρ 为信息素的挥发系数，一般 $0<\rho<1$。当达到最大迭代次数或者找寻到最优路径，则算法停止，输出全局最优解。

3. 模型建立

根据最小点覆盖的特点，蚂蚁在搜索过程中是可以从一个点转移到其他任一顶点的，故作图 G 的完全图 $G_C=(V,E_C)$，如果一个边是 G 中的边那么它的权值是 1，如果它未出现在原来的图 G 中，那么它的权值为 0。

定义 $G_{ck}=(V,E_C)$ 为 k 个顶点加入解集后的图，定义 $\Psi_k(i,j)=Value(E_{ck}(i,j))$。

定义动态启发函数

$$\eta_{jk}=\frac{\sum_{(i,j)\in E_c}\Psi_k(i,j)}{\omega(j)}\tag{4}$$

在式(4)中，$\omega(j)$ 指顶点 j 的权值，用启发函数 η_{jk} 定义状态转移概率方程

$$p_j^k=\begin{cases}1, q<q_0, j=\max_{i\in A_k}\tau_i\eta_{ik}^{\beta}\\ 0, q<q_0, j\neq\max_{i\in A_k}\tau_i\eta_{ik}^{\beta}\\ \dfrac{\tau_j\eta_{jk}^{\beta}}{\sum_{\gamma\in A_k}\tau_i\eta_{ik}^{\beta}}, q\geq q_0\end{cases}\tag{5}$$

在式(5)中，q_0 是标准参数且 q 是一个随机变量，它决定每一步选择的类型。A_k 是可用的顶点的列表。不同于 TSP 的状态转移规则，它并不取决于最后一个被选择的点，这就是为什么用 τ_i 来代替 τ_{ij}。对一个完全指定的蚂蚁群系统全球(当蚂蚁完成它们的所有路径时)，信息素更新为

$$\tau_i(t+1)=\rho\tau_i(t)+\Delta\tau_i^{best}\tag{6}$$

$$\Delta\tau_{ij}^{best}=1/L^{best} \tag{7}$$

在式(7)中的 L^{best} 为当前迭代最优解。每条路径上的信息素 $\tau_i(t)$ 都限于 $[\tau_{\min},\tau_{\max}]$ 范围中。每次迭代之后都应将信息素限定在该范围内。若 $\tau_i(t)\geq\tau_{\max}$，则 $\tau_i(t)=\tau_{\max}$；若 $\tau_i(t)\leq\tau_{\min}$，则 $\tau_i(t)=\tau_{\min}$，与此同时要保证 $\tau_{\min}>0$。

局部信息素更新的主要目的是为了减少算法陷入局部最优的状态，同时增加短路径上的信息素浓度。在每一步结束时执行局部信息素的更新，更新规则为

$$\tau_i=(1-\varphi)\tau_i+\varphi\tau_0 \tag{8}$$

式(8)中，τ_0 的值是指被选择的点的度数与权值之间的最佳比。参数 φ 是被指定的特殊的局部更新规则的值。

4. 算法设计

对点赋权图 $G=(V,E)$，做出图 G 的完全图 $G_C=(V,E_C)$。

首先，定义连接函数

$$\Psi_k(i,j)=\begin{cases}1, & i,j\in E\\ 0, & i,j\in E_C-E\end{cases} \tag{9}$$

由此函数即可确定各边的连接值。对于蚂蚁 k，当蚂蚁 k 访问到 i 点，则 $\Psi_k(i,j)=0$，其中 j 表示所有与 i 相邻的顶点。

然后计算动态启发因子

$$\eta_{jk}=\frac{\sum_{(i,j)\in E_c}\Psi_k(i,j)}{\omega(j)} \tag{10}$$

其中 $\omega(j)$ 为 j 点的权值，$\sum_{(i,j)\in E_c}\Psi_k(i,j)$ 表示顶点 j 所关联的所有的边的连接值之和。因此可求出 $\max\{\eta_{jk}^{\beta}\}$。

由此给出蚂蚁 k 接下来访问新的顶点的状态转移概率

$$P_j^k=\begin{cases}1, & j=\max\{\eta_{jk}^{\beta}\}\\ 0, & j\neq\max\{\eta_{jk}^{\beta}\}\end{cases} \tag{11}$$

对于图 G，当所有的连接值均为 0 时，算法停止。当蚂蚁完成环游，对于所有点 i，有 $\sum_{(i,j)\in E_c}\Psi_k(i,j)=0$，此时可得到一个点覆盖集 S_k。更新初始点，得到 n 个新的点覆盖集，计算 $\min\limits_{j\in S_k}\omega(j)$，则得到图的最小点覆盖集。

算法实现步骤

Step1. 初始化：做出图 G 的完全图 $G_C=(V,E_C)$，由连接函数

$$\Psi_k(i,j)=\begin{cases}1, & i,j\in E\\ 0, & i,j\in E_C-E\end{cases}$$

给定的各边连接值，令 $S_k=\{v_i\}$，$k=1$；

Step2. 用连接函数 $\Psi_k(i,j)=0$ 更新与 i 点相关联的所有边的连接值，计算

$$\eta_{jk}=\frac{\sum_{(i,j)\in E_c}\Psi_k(i,j)}{\omega(j)}$$

若有 $\eta_{jk}=0$ 则输出点覆盖，算法停止。否则，转 Step3；

Step3. 计算 $\max\left\{\eta_{jk}^{\beta}\right\}$，由

$$P_j^k=\begin{cases}1, & j=\max\left\{\eta_{jk}^{\beta}\right\}\\ 0, & j\neq\max\left\{\eta_{jk}^{\beta}\right\}\end{cases}$$

确定下一个选择的顶点，并将其并入集合 S_k，转 Step2；

Step4. 计算 $\left|S_k\right|=\min\limits_{j\in S_1}\omega(j)$，更新初始点，转 Step2；

Step5. 计算 $S=\min\left\{S_1,S_2,\cdots,S_k,\cdots,S_n\right\}$，输出最小点覆盖集 S，算法停止。由此得出算法的时间复杂度为 $0\left[(n-1)^2\cdot m\right]$，其中 n 为图 G 的顶点数，m 为蚂蚁数目，若 $m\approx n$，则蚁群算法的时间复杂度为 $0\left[(n-1)^2\cdot n\right]$。

5. 算例分析

求图 1 的点覆盖来阐释算法的具体执行过程，其中图 G 是 8 个顶点，12 条边的点赋权图且图 2 中的图 G_C 是图 G 的完全图。

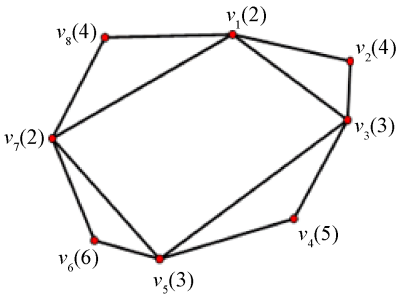

Figure 1. Graph G
图 1. 图 G

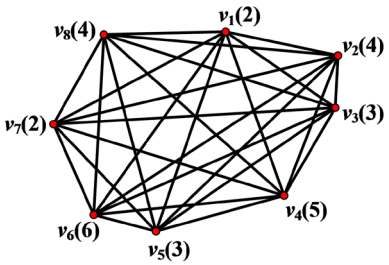

Figure 2. Graph G_C
图 2. 图 G_C

Step1. 由连接函数

$$\Psi_k(i,j)=\begin{cases}1, & i,j\in E\\ 0, & i,j\in E_C-E\end{cases}$$

给定的各边连接值，蚂蚁 1 以 v_1 为出发点，有 $S_1=\{v_1\}$，期望启发因子 β 赋值为 2；

Step2. 用连接函数 $\Psi_k(i,j)=0$ 更新与点 v_1 关联边的连接值，计算 η_{jk}^{β} 有 $\eta_{jk}^{\beta}\neq 0$，转 Step3；

Step3. 计算 $\max\left\{\eta_{jk}^{\beta}\right\}=12.25, j=7$，将其并入集合 $S_1=\{v_1,v_7\}$，转 Step2；

Step2. 用 $\Psi_k(i,j)=0$ 更新 v_7 关联边的连接值，计算 η_{jk}^{β}，有 $\eta_{jk}^{\beta}\neq 0$，转 Step3；

Step3. 计算 $\max\left\{\eta_{jk}^{\beta}\right\}=16/9, j=3$，将其并入集合 $S_1=\{v_1,v_7,v_3\}$，转 Step2；

Step2. 用连接函数 $\Psi_k(i,j)=0$ 更新与点 v_3 关联的边的连接值，计算 η_{jk}^{β}，有 $\eta_{jk}^{\beta}\neq 0$，转 Step3；Step3. 计算 $\max\left\{\eta_{jk}^{\beta}\right\}=4/9, j=5$，将其并入集合 $S_1=\{v_1,v_7,v_3,v_5\}$，转 Step2；

Step2. 用连接函数 $\Psi_k(i,j)=0$ 更新与点 v_5 关联的边的连接值，计算 η_{jk}^{β}，有 $\eta_{jk}^{\beta}=0$，输出 $S_1=\{v_1,v_2,v_3\}$；

Step4. 计算 $|S_k|=\min_{j\in S_1}\omega(j)=10$，算法停止；

Step5. 计算 $S=\min\{S_1,S_2,\cdots,S_k,\cdots,S_n\}$ 得 $S=\min\{v_1,v_3,v_5,v_7\}$，输出最小点覆盖集 $S=\{v_1,v_3,v_5,v_7\}$，算法停止。

比较表 1 中点覆盖的点覆盖数和权值，集合 $\{v_1,v_3,v_5,v_7\}$ 的点覆盖数和权值最小，点覆盖数为 4，点覆盖权值为 10。该算法点覆盖数和权值没有出现局部最优。

Table 1. Number of point overlays and point overlays of each vertex in figure *G*
表 1. 图 *G* 各顶点的点覆盖数及点覆盖值

蚂蚁	初始点	点覆盖	点覆盖数	点覆盖权值
1	v_1	$S_1=\{v_1,v_3,v_5,v_7\}$	4	10
2	v_2	$S_2=\{v_1,v_2,v_3,v_5,v_7\}$	5	14
3	v_3	$S_3=\{v_1,v_3,v_5,v_7\}$	4	10
4	v_4	$S_4=\{v_1,v_3,v_4,v_5,v_7\}$	5	15
5	v_5	$S_5=\{v_1,v_3,v_5,v_7\}$	4	10
6	v_6	$S_6=\{v_1,v_3,v_5,v_6,v_7\}$	5	16
7	v_7	$S_7=\{v_1,v_3,v_5,v_7\}$	4	10
8	v_8	$S_8=\{v_1,v_3,v_5,v_7,v_8\}$	5	14

为使该算例更具代表性，选取 $n=|V|=3r+4$，图 3 中最优覆盖有 $r+2$ 个点，为 $\{B_1,B_2,\cdots,B_{r+2}\}$，用 20 只蚂蚁对该实例求解，计算结果见表 2。

从表 2 中可知当图规模较小时，蚁群算法的早期停滞问题并没有出现。然而，随着图的复杂程度增加、点数的增多和 r 的增大，蚁群算法在计算过程中出现了停滞现象，所得结果是存在局部最优现象的。对信息素浓度的限定，有效的避免了该现象的出现，所得结果更加接近最优结果，计算精度方面得到了一定的提升。

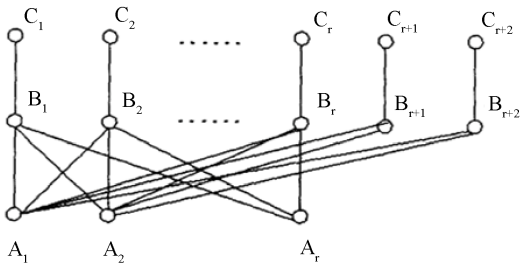

Figure 3. Numerical example
图 3. 算例

Table 2. The calculation results
表 2. 计算结果

r	n	最优点覆盖数	近似算法点覆盖数
15	49	17	18
20	64	22	23
25	79	27	30
30	99	32	35
35	109	37	41

6. 结论

蚁群算法相较各种成熟的智能计算方法，仍在实验探索阶段，但从当前的应用效果来看这种新型系统寻优思想对解决一些实际问题表现突出，解决赋权图点覆盖问题是可行的。

本文主要对蚂蚁选择下一顶点的状态转移概率进行改进，同时对信息素浓度进行限定，设置了最大值与最小值，这样可以使信息素浓度不会在好的顶点上继续加强也不会忽略掉一些潜在的搜索区域，使得解决点覆盖问题的精确度更高，运行效率更快。通过实例计算表明该近似算法提高了准确性，所得解更加趋近于最优解。

基金项目

国家自然科学基金(61463027，61463026)，甘肃省自然科学基金(1610RJZA038)。

参考文献

[1] Johnson, D.S. (1979) Approximation Algorithms for Combinatorial Problems. *Journal of Computer and System Sciences*, **9**, 256-278. https://doi.org/10.1016/S0022-0000(74)80044-9

[2] 谢政. 网络最优化[M]. 北京: 科学出版社, 2014.

[3] Balaji, S., Swaminathan, V. and Kannan, K. (2010) An Effective Algorithm for Minimum Weighted Vertex Cover Problem. *International Journal of Computational and Mathematical Sciences*, **4**, 34-38.

[4] Niedermeier, R. and Rossmanith, P. (2003) On Efficient Fixed-Parameter Algorithms for Weighted Vertex Cover. *Journal of Algorithms*, **47**, 63-77. https://doi.org/10.1016/S0196-6774(03)00005-1

[5] Chvatal, V. (1979) A Greedy Heuristic for the Set-Covering Problem. *Mathematics of Operations Research*, **4**, 233-235. https://doi.org/10.1287/moor.4.3.233

[6] Taoka, S. and Watanabe, T. (2012) Performance Comparison of Approximation Algorithms for the Minimum Weight Vertex Cover Problem. *ISCAS*, **209**, 632-635. https://doi.org/10.1109/ISCAS.2012.6272111

[7] 寇磊, 崔笑川, 陈京荣. 最小权点覆盖问题的一个近似算法[J]. 数学的实践与认识, 2015, 45(12): 201-206.

[8] 王丽丽, 崔晋川. 求解最小点覆盖问题实例的计算成本的一种度量方法[J]. 数学的实践与认识, 2017, 47(23): 142-149.

[9] 骆伟忠, 蔡昭权. 基于核心化技术的点覆盖改进算法[J]. 计算机工程与科学, 2018, 40(8): 1405-1411.

[10] 郝斌斌, 吕斌, 陈京荣. 基于最小点覆盖的共享单车投放点选取方法[J]. 交通信息安全, 2018, 36(5): 147-152.

[11] Colorni, A., Dorlgo, M. and Maniezzo, V. (1992) An Investigation of Some Properties of an Ant Algorithm. In: *Proceedings of the Parallel Problem Solving from Nature Conference*, Elsevier Publishing, Brussels, 509-520.

[12] Bonabeau, E., Dorigo, M. and Teraulaz, G. (2000) Inspiration Foe Optimzation from Social Insect Behavior. *Nature*, **406**, 39-42. https://doi.org/10.1038/35017500

[13] 吴佩雯, 陈京荣, 姬璐烨. 基于蚁群算法的赋权图点覆盖问题[J]. 应用数学进展, 2017, 6(9): 1119-1125.

[14] 葛洪伟, 高阳. 基于蚁群算法的集合覆盖问题[J]. 计算机工程与应用, 2007, 43(4): 49-50, 255.

[15] 范辉, 华臻, 李晋江, 原达. 点覆盖问题的蚂蚁算法求解[J]. 计算机工程与应用, 2004, 40(23): 71-73.

[16] Bouamama, S., Blum, C. and Boukerram, A. (2012) A Population-Based Iterated Greedy Algorithm for the Minimum Weight Vertex Cover Problem. *Applied Soft Computing Journal*, **12**, 1632-1639. https://doi.org/10.1016/j.asoc.2012.02.013