

文章编号:1672-6197(2017)04-0013-06

基于栅格的移动机器人区域分解遍历算法

朱宝艳, 李彩虹, 王小宇, 宋 莉

(山东理工大学 计算机科学与技术学院, 山东 淄博 255049)

摘 要: 针对不同类型障碍物提出基于栅格的区域分解方法使移动机器人实现全覆盖遍历. 算法包含区域分解、子区域连接和子区域内遍历三部分. 区域分解是按照凹型障碍物边缘和工作环境边界将栅格区域分解成若干个子区域. 区域内遍历按照障碍物不同类型, 对存在凸型障碍物的区域采用内螺旋方法, 对于凹型障碍物区域采用梳状遍历方法. 子区域之间通过两点法求最短路径, 然后按照逆时针方向形成遍历连通图. 通过 MATLAB 对算法进行仿真, 结果验证了该算法的可行性和有效性.

关键词: 移动机器人; 栅格; 区域分解; 遍历规划

中图分类号: TP242

文献标志码: A

DOI:10.13367/j.cnki.sdgc.2017.04.004

Cellular decomposition algorithm of coverage path planning based on the grid map for the mobile robot

ZHU Bao-yan, LI Cai-hong, WANG Xiao-yu, SONG Li

(School of Computer Science and Technology, Shandong University of Technology, Zibo 255049, China)

Abstract: According to the different types of obstacles, this paper presents a cellular decomposition algorithm of the coverage path planning based on the grid map for the mobile robot. The algorithm consists of three parts, including the cellular decomposition, the sub-regions connections and the coverage of the sub-regions inside. The cellular decomposition is to divide the feasible regions of the grid map into several sub-regions according to the boundaries of the concave obstacles and workplace. According to the different types of obstacles in the sub-regions, an internal spiral method is adopted for the area with convex obstacles and a back-and-forth method is adopted for the area with concave obstacles. Two-points method is used to find the shortest path between the adjacent sub-regions. Then a complete connected graph is formed according to the counter clockwise direction. The feasibility and effectiveness of the algorithm are verified by the simulation of MATLAB.

Key words: mobile robot; grid; cellular decomposition; coverage path planning

近年来,伴随着智能服务机器人^[1]迅速发展的同时,智能服务机器人研究领域的一个重要分支——全覆盖路径规划技术也逐渐受到众多研究者的关注.与通常所说的点到点的路径规划不同,全覆盖遍历路径规划的最终目标是使机器人能够按照一

定的标准,在具有障碍物的环境中,无碰撞的合理而高效的遍历除障碍物以外的全部区域.移动机器人全覆盖遍历算法的研究分为机器人环境建模方法和路径搜索算法两个方面,对于未知环境建模可以采用边走边建模的方法,也可以通过 SLAM(Simulta-

收稿日期: 2016-07-02

基金项目:国家自然科学基金项目(61473179);山东省自然科学基金项目(ZR2014FM007)

作者简介:朱宝艳,女, zby0682@126.com;通信作者:李彩虹,女, lich@sdu.edu.cn

neous Localization and Mapping)^[2] 转换成已知环境,对于已知环境建模的方法主要有栅格表示法^[3]、可视图法^[4](几何表示法)和拓扑图法^[5]等.对于全覆盖路径搜索算法主要有基于栅格表示法^[6]的路径规划、基于行为的路径规划^[7]、基于势场栅格地图的生物激励法、模板模型法^[8]、单元分解法^[9-10]等.

基于栅格地图常用的搜索算法是区域分解法.该方法将机器人所要遍历的区域根据环境中的障碍物或其它方法分为若干子区域,通过对各子区域的遍历实现对整个区域的遍历.这种思想在很大的程度上降低了全局覆盖实现的难度,因此对区域分解法的研究是近年来的主要趋势.本文采用基于栅格区域分解的方法对环境建模,机器人依次遍历每个子区域并通过局部路径规划方法实现各个子区域的高效连接.对每个子区域的遍历按照障碍物的不同类型,对凸型障碍物存在的区域采用内螺旋的方式,对凹型障碍物存在的区域采用梳状遍历的方式.

1 环境建模

环境建模主要是机器人通过传感器获取实时环境信息并将环境进行栅格量化,这里将地图环境按照机器人大小进行栅格化处理,建立栅格地图,以此为基础进行路径规划算法研究.


1.1 建立量化栅格的环境地图

将工作环境栅格化后,整个区域分成 $n \times n$ 个栅格,每个栅格生成相应坐标 $cell(x, y)$, 其中, $1 \leq x \leq n$, $1 \leq y \leq n$. 然后针对每个栅格设定属性值:

$$cell(x, y).block = \begin{cases} 0 & \text{自由栅格} \\ 1 & \text{障碍物栅格} \end{cases} \quad (1)$$

当栅格属性 $cell(x, y).block = 0$ 时,说明该栅格为自由栅格,此时该栅格可以被覆盖,可以被覆盖的栅格还有一个覆盖属性 $cell(x, y).visited$, 其初始值设置为 0, 栅格每被覆盖一次,其覆盖属性值依次累加 1, 即

$$cell(x, y).visited = cell(x, y).visited + 1 \quad (2)$$

图 1 画出了量化后的栅格地图. 其中  表示机器人,黑色表示障碍物,空白栅格表示自由栅格,机器人的任务就是要遍历所有的自由栅格.

1.2 环境的障碍物模型

障碍物种类可以分为凹型障碍物和凸型障碍物两种,凹型障碍物一般是半封闭型,比如 U 型、V 型等,凹型障碍物以外的障碍物归类为凸型障碍物.

凸型障碍物,如图 2 所示.

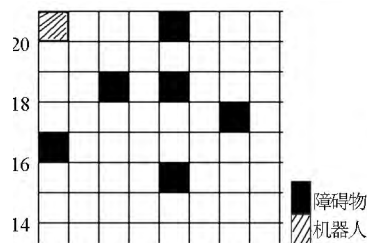


图 1 栅格地图

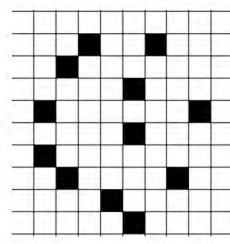


图 2 凸型障碍物

凹型障碍物,比如 U 型障碍物,如图 3 所示.

在环境地图中,既有凹型障碍物又有凸型障碍物,如图 4 所示.

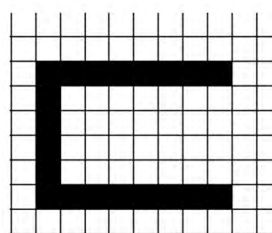


图 3 凹型障碍物

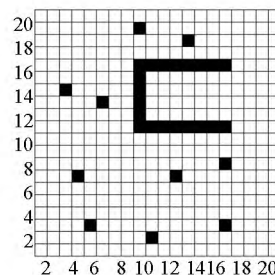


图 4 混合类型障碍物

2 基于栅格区域分解法的算法设计

基于栅格的区域分解法包含区域分解、子区域连接和子区域遍历三部分.在含有混合障碍物的环境里,首先以凹型障碍物边缘和环境地图边界为标准进行区域划分,然后通过局部路径规划算法求相邻区域的最短路径,最后将子区域按照逆时针方向连接起来,形成遍历连通图.子区域遍历根据障碍物的不同类型采用不同的遍历方法.

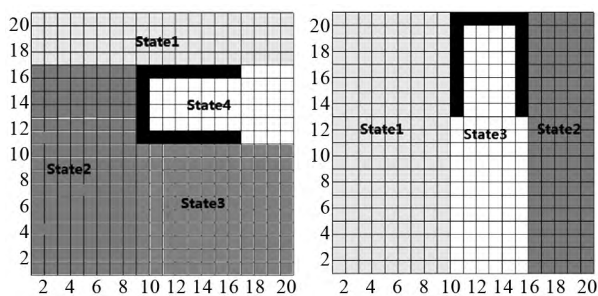
2.1 栅格的区域分解

在含有混合障碍物环境中,这里以含有 U 型障碍物为例说明,在可行区域按照 U 型障碍物边缘和环境边界进行分解,根据 U 型障碍物在环境地图的不同位置,如图 5 所示,有以下几种分解情况:

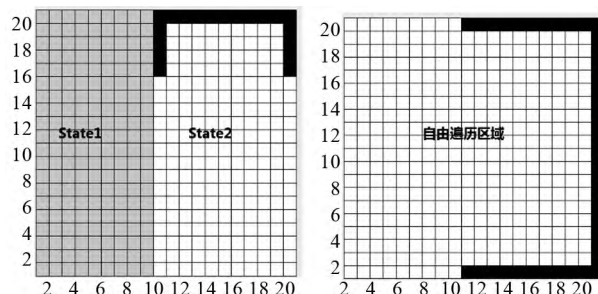
1) 当 U 型障碍物三边与栅格地图边界均没有接触时为一般情况,可以分成四部分,分别为 State1, State2, State3, State4, 如图 5(a).

2) 当 U 型障碍物一边与栅格地图边界接触时,可以分成三部分,分别为 State1, State2, State3, 如图 5(b).

3) 当 U 型障碍物两边与栅格地图边界接触时,可以分成两部分,分别为 State1, State2, 如图 5(c).



(a) 一般 U 型障碍物单元分解图示 (b) 障碍物一边靠在栅格地图边界



(c) 障碍物两边靠在栅格地图边界 (d) 障碍物三边均在栅格地图边界

图5 栅格地图的区域分解

4) 当 U 型障碍物三边都与栅格地图边界接触时,除 U 型三边外,将整个自由栅格区域看做 U 型内部区域,如图 5(d).

以图 5(a)为例说明算法设计过程.

2.2 子区域之间连接

子区域之间的连接包括子区域起点、终点位置确定,相邻子区域连接方法设计和子区域连通图设计三部分.

1) 子区域起点、终点位置确定

以图 5(a)为例,一共有四个子区域,分别为 State1、State2、State3、State4,相邻两个子区域连接的起点在子区域的边界. State1 作为遍历开始的子区域,设置机器人的起点为 cell(1, n),遍历结束点为 cell(x1, y1). State2 起点有两个, cell(1, j) 或者 cell(i, j),通过计算结束点 cell(x1, y1) 到 cell(1, j) 或 cell(i, j) 的距离,取最短距离的点为 State2 的起点, State2 遍历结束点为 cell(x2, y2). State3 起点也有两个, cell(i, 1) 或者 cell(i, j2),通过计算结束点 cell(x2, y2) 到 cell(i, 1) 或 cell(i, j2) 的距离,取最短距离的点为 State3 起点, State3 遍历结束点为 cell(x3, y3). State4 起点是确定的,为 U 型障碍物的右下角,设 U 型障碍物上下边缘长度为 m,则起点为 cell(i+m, j2),遍历结束点为 cell(i+m, j).

State1 结束点到 State2 起点的最短距离:

$$d_1 = \sqrt{(x_1 - i)^2 - (y_1 - j)^2} \quad (3)$$

$$d_2 = \sqrt{(x_1 - 1)^2 - (y_1 - j)^2} \quad (4)$$

通过 $\min(d_1, d_2)$ 确定 State2 的起点.

State2 结束点到 State3 起点的最短距离:

$$d_3 = \sqrt{(x_2 - i)^2 - (y_2 - 1)^2} \quad (5)$$

$$d_4 = \sqrt{(x_2 - i)^2 - (y_2 - j_2)^2} \quad (6)$$

通过 $\min(d_3, d_4)$ 确定 State3 的起点.

2) 相邻子区域连接方法设计

根据确定的相邻两个子区域之间起点与结束点,按照两点法搜索策略进行局部路径规划,求出相对最短路径.两点法搜索策略简述为:移动机器人向目标栅格直线移动时,若遇到障碍物,就改变方向、移动到没有障碍物的位置,并再次确定目标栅格的方向、感知障碍物信息.如此反复,直至到达目标栅格.两点法搜索策略局部路径规划仿真如图 6 所示,⊞表示起点,⊞表示终点,黑色表示障碍物,⊞表示规划的路径.

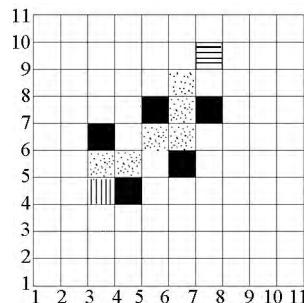


图6 两点法局部路径规划

3) 子区域连通图设计

四个子区域之间两两相邻,因此直接按照逆时针(顺时针)方向形成一张完全连通图,即移动机器人按照 State1 → State2 → State3 → State4 顺序依次对每个子区域完成遍历.

2.3 子区域遍历

子区域遍历根据障碍物的不同类型,在凸型障碍物区域采用内螺旋遍历方式,在凹型障碍物区域采用梳状遍历方式.

1) 内螺旋遍历算法

在凸型障碍物的环境地图中,判断程序结束的覆盖率用 coverage 表示,最大覆盖率用 max 表示.

根据设置的属性值判断机器人的动作目标:

G1 准目标栅格(即为前进方向的下一个栅格)

G2 障碍物栅格

G3 自由栅格(目标栅格相关的栅格)

具体算法如下:

Step1 初始化机器人位置坐标 $cell(1, n)$, 最大覆盖率 $\max = 100\%$, 机器人运行方向为顺时针方向.

Step2 判断覆盖率 $\text{coverage} < \max$ 是否成立, 否则转移到 Step6, 是则继续执行.

Step3 判断准目标栅格 $G1$ 的属性值 $cell(i, j). \text{block} = 1$ 时, 转移到 Step5, 否则继续执行.

Step4 该栅格为自由栅格 $G3$, $cell(x, y). \text{visited} = cell(x, y). \text{visited} + 1$, 转移到 Step2.

Step5 判断障碍物栅格 $G2$ 内侧的准目标栅格 $G1$ 属性值 $cell(i, j). \text{block} = 0$ 时, 转移到 Step4, 否则继续执行 Step5.

Step6 退出循环, 算法结束.

程序流程图如图 7 所示.

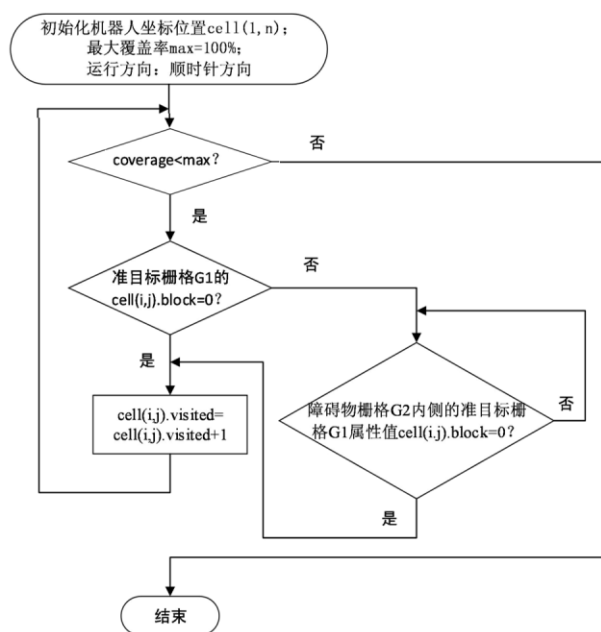
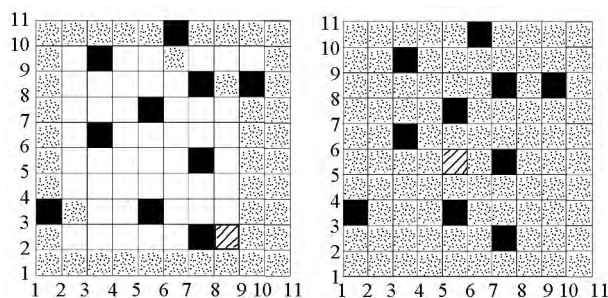


图 7 内螺旋程序流程图

算法仿真过程如图 8 所示. 图 8(a) 为遍历中间过程, 图 8(b) 是遍历结束, 黑色表示障碍物, 表示机器人遍历过程中的位置和遍历结束的位置, 表示遍历路径.

2) 梳状遍历算法

在 U 型障碍物栅格地图中, (i, j_2) 为 U 型障碍物左下角栅格坐标, (i, j) 为 U 型障碍物左上角栅格坐标. U 型区域遍历的初始位置横坐标表示为 $x = i + m$, 纵坐标表示为 $y = j_2$, 根据机器人起始点位置可知 U 型区域下边缘为起始行, 定义其为奇数行, 运行方向为横坐标增大的方向, 相反的, 在偶数行, 运行方向为横坐标减小的方向. 用 r 表示奇



(a) 遍历中间过程 (b) 遍历结束

图 8 内螺旋算法仿真

数行或者偶数行的计算, 用 r_0 表示奇偶数计算结果. r 的计算方法如式 (7).

$$r = \frac{j - j_2}{2} \quad (7)$$

具体算法如下:

Step1 初始化 U 型区域的起始坐标为 $cell(x, y)$, $r_0 = 0$, 初始运行方向为横坐标增大的方向.

Step2 判断 $y = j$, 是则转移到 Step7, 否则继续执行.

Step3 判断 $r = r_0$, 是则转移到 Step9, 否则继续执行.

Step4 机器人沿着当前方向横坐标 $x = x + 1$, 纵坐标不变, $cell(x, y). \text{visited} = cell(x, y). \text{visited} + 1$.

Step5 判断 $x = n$, 否则转移到 Step4, 是则继续执行.

Step6 纵坐标 $y = y + 1$, 并且转移到 Step2.

Step7 判断 $x = i + m$, 是则转移到 Step12, 否则机器人沿着当前方向横坐标 $x = x - 1$, 纵坐标不变, $cell(x, y). \text{visited} = cell(x, y). \text{visited} + 1$, 并继续执行 Step7.

Step8 判断 $x = i + 1$ 并且 $y = j - 1$, 是则转移到 Step4, 否则继续执行.

Step9 机器人沿着当前方向横坐标 $x = x - 1$, 纵坐标不变, $cell(x, y). \text{visited} = cell(x, y). \text{visited} + 1$.

Step10 判断 $x = i + 1$, 否则转移到 Step6, 是则继续执行.

Step11 判断 $y = j - 1$, 是则转移到 Step4, 否则转移到 Step6.

Step12 退出循环, 算法结束.

程序流程图如图 9 所示.

算法仿真过程如图 10 所示. 其中 10(a) 为遍历开始, 10(b) 是遍历结束, 黑色为障碍物, 表示分别为机

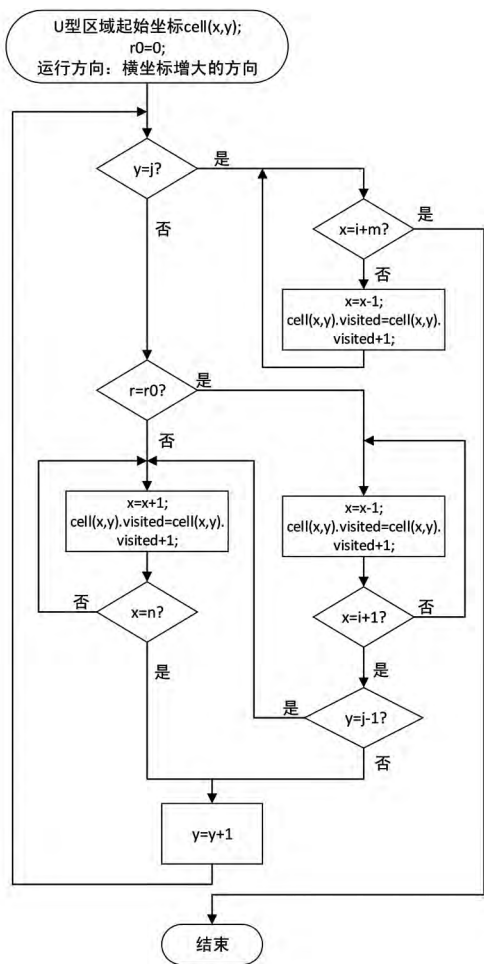
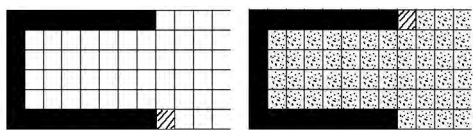


图 9 梳状遍历的流程图

器人在 U 型遍历的起点和终点, 是遍历路径.



(a) 遍历开始

(b) 遍历结束

图 10 梳状遍历仿真

3 算法评价

根据移动机器人全覆盖遍历路径规划的要求, 一是提高覆盖率, 二是降低重复率, 通过计算得到覆盖率和重复率的值, 判断该算法的可行性和有效性.

覆盖率计算:

$$coveragerate = \left(\frac{num}{n \times n - ObsNumber} \right) \times 100\% \quad (8)$$

重复率计算:

$$repeat = \left(\frac{numb}{n \times n - ObsNumber} \right) \times 100\% \quad (9)$$

式中: num 为覆盖栅格个数; $numb$ 为重复覆盖栅格个数; n 为环境区域边长; $ObsNumber$ 为障碍物个数.

4 实验仿真和分析

在 MATLAB 上设计一个仿真实验平台, 仿真的栅格区域大小为 20×20 , 初始化机器人坐标 $cell(1, 20)$, 障碍物栅格量化后用黑色表示. 仿真结果如图 11 所示.

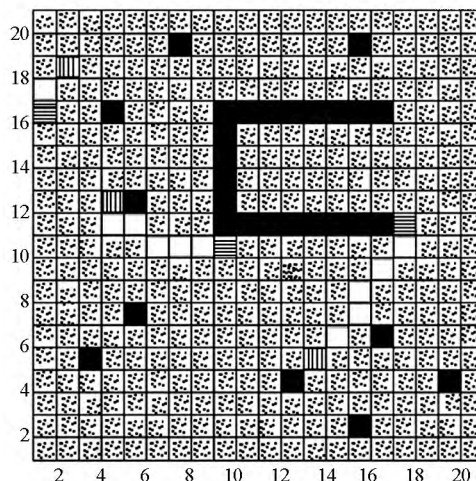


图 11 仿真结果

仿真环境中表示机器人全局路径规划遍历路径, 表示子区域连接的结束点, 表示子区域连接的起点, 空白表示相邻子区域连接路径. 从仿真实验结果可知, 机器人对自由区域的覆盖率达到 100% 时重复率为 5.68%, 采用混合式算法达到同样覆盖率的情况下, 重复率为 14.45%, 说明所设计算法是可行, 有效的, 可以提高覆盖率, 降低重复率.

5 结束语

本文采用基于栅格的区域分解的方法实现了移动机器人全覆盖遍历, 对栅格地图按照凹型障碍物边缘和环境边界分成了几个子区域, 用两点法搜索算法将这几个子区域按逆时针方向形成一张完全连通图, 每个子区域根据障碍物不同类型对凸型障碍物采用内螺旋遍历, 对凹型障碍物采用梳状遍历. 通过 MATLAB 仿真平台模拟了遍历过程, 仿真结果表明, 该算法在实现移动机器人全覆盖遍历的过程

具有可行性和有效性.

基于栅格的移动机器人区域分解遍历算法,能高效的完成全覆盖遍历的要求,但也存在一些不足,如进行区域分解时如果遇到复杂的凹型和凸型障碍物如何进行子区域分解才能达到重复率最低,进行局部路径规划时,尚不能保证遍历路径绝对最短等问题,这是今后要继续研究的课题.

参考文献:

- [1] 赵晓东, 鲍方. 清洁机器人路径规划算法研究综述[J]. 机电工程, 2013, 30(11): 1 440-1 444.
- [2] 季秀才, 郑志强, 张辉. SLAM 问题中机器人定位误差分析与控制[J]. 自动化学报, 2008, 34(3): 323-330.
- [3] 刘松, 李志蜀, 李奇. 机器人全覆盖最优路径规划的改进遗传算法[J]. 计算机工程与应用, 2009, 45(31): 245-248.
- [4] 艾海舟, 张钹. 基于拓扑的路径规划问题的图形解法[J]. 机器人. 1990, 12(5): 20-24.

- [5] 纪晴, 段培永, 李连防. 移动机器人全覆盖路径规划算法综述[J]. 山东建筑大学学报, 2007, 22(4): 355-359.
- [6] 陈鹏, 李彩虹. 移动机器人全遍历覆盖路径规划研究[J]. 山东理工大学学报(自然科学版), 2013, 27(5): 22-27.
- [7] 陈炜峰, 薛冬, 周旺平. 基于行为模糊控制的移动机器人路径规划[J]. 计算机测量与控制, 2014, 22(11): 3 600-3 602.
- [8] 刘海, 郭小勤, 余得贵. 清洁机器人全覆盖路径规划算法综述[J]. 机电产品开发与创新, 2008, 21(6): 36-38.
- [9] CHOSSET H. Coverage of known spaces: The boustrophedon cellular decomposition[J]. Autonomous Robots, 2000, 5(3): 247 - 253.
- [10] FABIO M M. A directional diffusion algorithm on cellular automata for robot path-planning[J]. Future Generation Computer Systems, 2002, 18(7): 982-994.

(编辑: 刘宝江)

(上接第 12 页)

参考文献:

- [1] 邹佳顺, 张永胜. ABAC 中基于前缀标记运算的策略检索方法[J]. 计算机工程与设计, 2015(11): 2 943-2 947.
- [2] LIM L, MARIE P, CONAN D, et al. Enhancing context data distribution for the internet of things using qoc-awareness and attribute-based access control[J]. Annals of Telecommunications, 2015, 71 (3): 1-12.
- [3] ZHU Y, LI J, ZHANG Q. General attribute based RBAC model for web services[J]. 武汉大学学报(自然科学英文版), 2008, 13 (1): 81-86.
- [4] JIN X, KRISHNAN R, SANDHU R. A unified attribute-based access control model covering DAC, MAC and RBAC[C]// Ifip Wg 11.3 Conference on Data and Applications Security and Privacy, 2012: 41-55.

- [5] ZHANG X, LI Y, NALLA D. An attribute-based access matrix model [C]// ACM Symposium on Applied Computing. 2005: 359-363.
- [6] 穆玲玲, 张韶松. 扩展的基于属性的访问矩阵模型[J]. 计算机工程与设计, 2012, 33(10): 3 797-3 800.
- [7] QIU B Z, LI X L. Grid-Based clustering algorithm based on intersecting partition and density estimation[J]. Proc of PAKDD. Berlin: Springer, 2007, 21(5): 368-377.
- [8] 谢娟英, 王艳娥. 最小方差优化初始聚类中心的 K-means 算法[J]. 计算机工程, 2014, 40(8): 205-211.
- [9] 汪中, 刘贵全, 陈恩红. 一种优化初始中心点的 K-means 算法[J]. 模式识别与人工智能, 2009, 22(2): 299-304.
- [10] FRANK A, ASUNCION A. UCI machine learning repository [D]. Irvine, USA: University of California, 2010.

(编辑: 刘宝江)