

流向图回溯法及传输网络 From-To 解算方法和技术研究^①

冯治东¹, 张培元²

¹(榆林学院 信息工程学院, 榆林 719000)

²(中国神华神东 锦界煤矿, 榆林 719000)

通讯作者: 冯治东, E-mail: fanston@126.com

摘要: 针对传输网络中流体“从哪里来, 到哪里去”的确定问题, 基于图回溯法, 提出了一种基于流向图的传输网络 From-To 解算方法. 根据传输网络中的驱动点、管道、闸阀和出口各属性状态, 将整个网络转化为初始流向图拓扑结构, 根据图回溯原理, 逐步累积计算管道中流体的来源和去向, 直到全部管道回溯结束, 得出最终流向图. 在此基础上, 研发了基于 Observer 设计模式的“From-To 解算”通用组件接口, 并被应用于某大型煤矿的复杂排水管网的计算机仿真平台中, 应用效果较好.

关键词: 传输网络; 流向图; From-To 解算; 回溯法; 矿井排水

引用格式: 冯治东, 张培元. 流向图回溯法及传输网络 From-To 解算方法和技术研究. 计算机系统应用, 2018, 27(6): 225-230. <http://www.c-s-a.org.cn/1003-3254/6404.html>

Research on Transport Network From-To Calculation Based on Flow Graph

FENG Zhi-Dong¹, ZHANG Pei-Yuan²

¹(School of Information Engineering, Yulin University, Yulin 719000, China)

²(Jinjie Colliery, Shenhua Shendong Coal Group Co. Ltd., Yulin 719000, China)

Abstract: An approach called “From-To Calculation Algorithm Based on Flow Graph” is presented. It is used to calculate the “from-position” and “to-position” of flow in complex piping network. By representing the piping network with graph topology, using the principle of maze backtracking, the direction of all the aspects can be determined in the piping system. A general component of the algorithm is developed using C# computer language based on generic parameter, interface technology and delegated observer mode. The algorithm is then applied to a mine drainage system. By taking the water pumps & joints as nodes, and pipes as edges, all flow “From-To” can be automatically calculated with “flow graph algorithm”. Application results show that it can be an effective solution to calculate where a flow is from and where it will go.

Key words: transport network; flow graph; From-To calculation; backtracking; mine drainage

本文的传输网络是指为将相关物质从源点输送到目标点所建立的传输网络, 如供电网络、通风系统和排水网络等. 通常情况下, 传输网络中包括了多个源点、输送路径和目标点. 人们往往希望达到这样一个目标: 任意指定传输网络中的一段路径, 求出该路径所有可能的源点和目标输送点. 当传输网络规模较大时, 问题求解便显得较为困难. 例如地下矿产资源开采过

程中, 井下采掘工作面、巷道低洼地段和储水仓等区域积聚了大量地下水, 尤其在突水事件发生时, 该类积水迅速积聚蔓延, 严重时甚至淹没巷道, 造成严重的人员生命危险和财产损失. 为了将水排出地面, 人们在井下部署了大量的排水设备, 组成排水网络, 随着积水点的增多, 排水工程师往往会将更多的排水设备接入, 导致排水网络越来越复杂, 如我国某大型地下煤矿排水

^① 基金项目: 陕西省教育厅专项计划项目 (16JK1896); 榆林学院博士科研启动基金 (16GK23)

收稿时间: 2017-10-11; 修改时间: 2017-11-03; 采用时间: 2017-11-17; csa 在线出版时间: 2018-05-28

网络中的水泵数量达五百多个, 闸阀达一千多个, 各类管道总长达两万多米, 导致人们面临的如下两个问题越来越突出: (1) 排水工程师在接入新的设备时, 确定已有的管道水流方向十分困难; (2) 在远程指挥和调度控制过程中, 决策人员很难在现有图纸基础上, 根据闸阀的开关状态, 动态掌握管道网络中各环节水流的来龙去脉, 影响决策效率. 排水网络中水流“从哪里来, 到哪里去”的确定问题成为迫切需要解决的问题. 有必要研究一种方法迅速解算排水网络的水流流向辅助决策, 这是本文的主要研究背景.

国内外的大多研究是借助三维辅助观察、手动记录或物联感知等方法确定流体的来源和去向, 或是借助一些数学方法解决路径优化或规划问题^[1-7], 尚无文献或软件平台涉及专门的传输网络来源去向自动解算问题, 鲜有学者以组合数学的观点, 在已知传输网络部署和工作状态情况下, 给出与实际流动情况一致的解算结果. 实际上, 图论是解算“流向”问题的有力工具之一, 图论在解决工程问题时, 方法十分灵活, 通常一种问题对应一种专有解法, 针对复杂线路网络中流体流向解算问题, 文献[8]提出了“树形探测法”计算流体方向, 但该方法受限于树状拓扑, 且仅标识了路径方向, 并不能解算来源和去向, 为适应复杂多样的传输网络, 不失一般性, 本文将传输网络拓扑推广到图, 并将排水网络流向解算问题抽象为流向图的 From-To 解算问题, 可应用于大多运输网络, 如通风系统、给排水系统和电路系统等.

1 From-To 解算基本思想及相关概念

本文的 From 指运输物体的来源点 (从哪里来), To 是指运输问题的目标去向 (到哪里去), 整个“From-To 解算”是指求解管路中物质的来源和去向 (从哪里来到哪里去).

在相关图论原理基础上^[9], 现对相关概念和思想扩展解释如下:

定义 流向图. 对于有向图, 若其顶点集合 V 与边集合 E 满足:

$$\begin{cases} G = (V, E) \\ V = \{v | v \in \{vd, vo, vs, ve\}\} \\ E = \{ \langle v_i, v_j \rangle | P(v_i, v_j) \wedge \langle v_i, v_j \rangle \in \{ef\} \wedge v_i, v_j \in V \} \end{cases} \quad (1)$$

其中, vd 表示驱动节点, vo 表示通节点, vs 表示闭节点,

ve 表示出口节点; $P(v_i, v_j)$ 表示存在边; $\{ef\}$ 表示流向边集合, 称该图为流向图. 相关术语解释如下:

驱动节点 (vd): 能够促使流体从该类节点向远方流动, 通常情况下 vd 处于叶子节点位置, 表示法如图 1(a) 所示.

通节点 (vo): 流体可通过该节点流通, 表达符号如图 1(b) 所示;

闭节点 (vs): 阻塞流体流通的节点, 表达符号如图 1(c) 所示;

出口节点 (ve): 流体的最终出口节点, 表达符号如图 1(d) 所示;

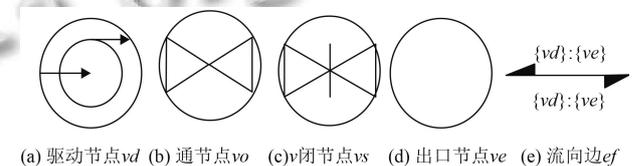


图 1 流向图节点及边的符号示意图

流向边 (ef): 两个节点之间流体流动情况, 包括两个方向, 通常地, 若规定一个方向为正流, 则其反方向为逆流. 每个方向上标识了该方向上的流源 (流体来自于哪些驱动节点) 和流目标 (流体流向哪些出口节点). 其语义为: $\{\text{流源}\}:\{\text{流目标}\}$, 例如 $\{9, 13\}:\{12\}$ 可理解为该流向上的流体来自于 9、13 号驱动节点, 将从 12 号出口节点流出. 由上述驱动节点、通节点、闭节点、出口节点和流向边构成的图称为流向图如图 2 所示, 本文规定计算之前的流向图为初始流向图 (此时流源和流目标为空集), 计算完毕后的流向图为最终流向图.

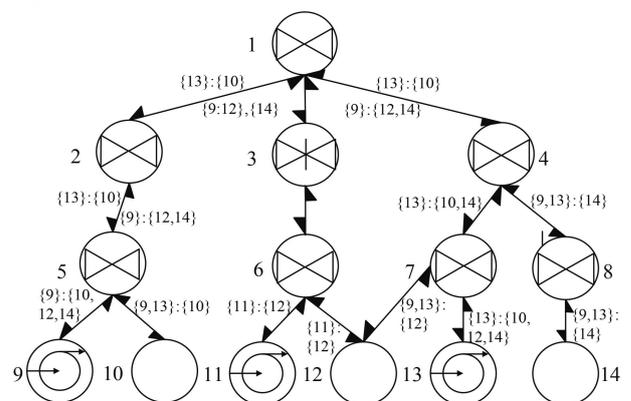


图 2 流向图

基于流向图的 From-To 解算原理: 流向图中, 根据各节点类型, 确定各流向边流源和流目标集合. 具体地,

每个驱动节点需要一个解算周期, 流体从驱动节点 vd 出发, 根据“回溯探测法”, 遇通节点 vo 则前行, 遇闭节点 vs 则标记相应流向边为闭合路径并回退, 遇出口节点 ve 则记为一条可行通路, 在该条可行路径中各流向边的流源集合中加入该驱动节点 vd , 流目标集合中加入出口节点 ve , 然后逐级回退探寻下一可行通路, 直到所有路径探寻完毕, 进入下一个驱动节点的流向解算周期, 直到所有驱动节点解算完成。

2 传输网络的 From-To 解算

根据上述基本原理及相关概念, 可将传输网络看作上述流向图, 在流向图基础上解算出各边的 From-To 信息, 最后将整个流向图的状态信息反馈到传输网络上. 传输网络到流向图的转换方法: 可将正在运转的来源点转化为驱动节点; 停止运转的来源点转化为闭节点; 开启的闸阀、关口转化为通节点; 关闭的闸阀、关口转化为闭节点; 网络中的最终出口转化为出口节点; 路径转化为流向边。

2.1 基本步骤和算法

步骤 1. 按照上述转化方法将传输网络转化为流向图; 设定图中所有驱动节点有且唯一的相邻通节点, 所有边的流源和流目标初始状态为空集且访问标志为“未访问”, 同时对该流向图中的每个节点进行编号;

步骤 2. 找出未被流向解算过的驱动节点 $vd_x \in vd$, 现从从其唯一的相邻节点 v_0 出发, 探寻 vd_x 的所有可行通路;

步骤 3. 将 v_0 节点入栈;

步骤 4. 检查栈内是否存在节点, 若不存在则探寻完毕, 得出该驱动节点所驱动流体的所有可行路径, 该驱动节点 vd_x 的解算周期结束, 转步骤 2 继续执行; 若存在, 设栈顶节点为 v_b , 则进行下一步;

步骤 5. 找出栈顶节点 v_b , 执行如下分支:

步骤 5.1. 若 v_b 是闭节点, 则回溯, 即将 v_b 出栈, 并将 v_b 到新的栈顶节点所形成的流向边标记为“已访问”, 转步骤 4;

步骤 5.2. 若 v_b 是出口节点, 则顺序从栈底到栈顶节点序列所形成的路径即为一条可行通路, 在路径上所有边的流源集合中加入该路径上的驱动节点, 所有边的流目标集合中加入该路径上的出口节点, 然后回溯, v_b 出栈, 并将节点 v_b 与新的栈顶节点所形成的流向边标记为“已访问”, 转步骤 4;

步骤 5.3. 若 v_b 是通节点, 则执行如下分支:

步骤 5.3.1. 若 v_b 存在“非栈内”且“未访问”的相邻节点 (“非栈内”指尚未入栈, “未标记”为相邻边尚未标记为“已访问”), 若存在, 则从满足条件的相邻节点中找出任一节点, 将该相邻节点入栈, 转步骤 4;

步骤 5.3.2. 若不存在, 则回溯, 首先将 v_b 的所有邻边重置为“未访问”, 然后将 v_b 出栈, 并将 v_b 到新的栈顶节点所形成的边标记为“已访问”, 转步骤 4;

步骤 6. 若所有驱动节点的解算周期完毕, 则表示, 找寻到所有的路径, 从而得出各边的流源和流目标, 算法结束。

2.2 实例验证

这里举例说明如图 3 所示的流向图 From_To 解算过程。

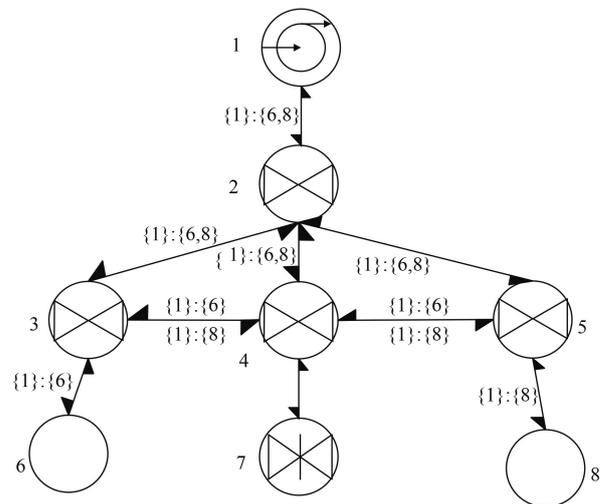


图 3 简单流向图实例

首先将所有边的流源和流目标初始化为空集, 现从驱动节点 1 出发, 按照 From_To 解算的基本步骤和原理进行如下:

(1) 将驱动节点 1 入栈, 此时栈内节点序列由底到顶为: {1};

(2) 找出栈顶节点为 1, 检查其节点类型为驱动节点, 此时初始的驱动节点等同于通节点, 则按照“非栈内”和“未访问”原则, 寻找其任一相邻节点 2, 压入栈内, 此时栈内节点序列由底到顶为: {1,2};

(3) 找出栈顶节点为 2, 检查其节点类型为通节点, 则按照“非栈内”和“未访问”原则, 寻找其任一相邻节点 3, 压入栈内, 此时栈内节点序列由底到顶为: {1,2,3};

(4) 找出栈顶节点为 3, 检查其节点类型为通节点, 则按照“非栈内”和“未访问”原则, 寻找其任一相邻节

点4, 压入栈内, 此时栈内节点序列由底到顶为: {1,2,3,4};

(5) 找出栈顶节点4, 检查到其类型为通节点, 则按照“非栈内”和“未访问”原则, 找出其任一相邻节点7, 压入栈内, 此时栈内节点序列由底到顶为: {1,2,3,4,7};

(6) 找出栈顶节点7, 检查到其节点类型为闭节点, 由于是闭节点, 移出栈外, 并将7到新的栈顶节点4所形成的边标记为“已访问”, 此时栈内节点序列由底到顶为: {1,2,3,4};

(7) 找出栈顶节点4, 检查到其节点类型为通节点, 则按照“非栈内”和“未访问”原则, 找出其任一相邻节点5, 压入栈内, 此时栈内序列: {1,2,3,4,5};

(8) 找出栈顶节点5, 检查到其节点类型为通节点, 则按照“非栈内”和“未访问”原则, 找出其任一相邻节点8, 压入栈内, 此时栈内节点序列: {1,2,3,4,5,8};

(9) 取出栈顶节点8, 发现8是出口节点, 已到达终点, 则栈内节点序列所形成的路径: 1->2->3->4->5->8便为一条可行通路, 同时将1-2、2-3、3-4、4-5、5-8方向上的流源集合加入1号节点, 流目标集合加入8号节点, 然后将8出栈, 然后将8到新的顶节点5标记为已访问;

(10) 找出栈顶节点5, 检查到其节点类型为通节点, 则按照“非栈内”和“未访问”原则, 找不到其相邻节点, 需要进行回溯, 首先将其全部相邻边重置为“未访

问”, 然后将5出栈, 并将节点5和新的栈顶节点4所形成的边标记为“已访问”, 此时栈内序列: {1,2,3,4};

(11) 找出栈顶节点4, 检查到4为通节点, 同理按照条件找不到4的相邻节点, 同样回溯, 此时栈内: {1,2,3};

(12) 找出栈顶节点3, 根据条件寻找到相邻节点6, 压入栈内, 此时栈内序列: {1,2,3,6};

(13) 找出栈顶节点6, 发现6为出口节点, 找出可行通路 1->2->3->6, 进行记录, 然后回溯;

(14) 同理, 探寻出可行通路 1->2->5->4->3->6 和 1->2->5->8, 进行记录和回溯;

(15) 直到此栈内节点数量为0, 此时寻出的所有可行通路, 从而得出各边的流源和流目标, 如图3所示, 验证结果符合预期目标。

3 计算机程序实现及应用

3.1 From-To 解算应用接口设计

本文借助 C#.NET 技术及相关软件工程开发方法学^[10-13], 在原树形探测法接口的基础上, 扩展了 From-To 解算接口, 如图4所示, 其中 DrainageGraph 类表达运输网络, 并与实现了 INode 和 IEdge 接口的节点和边进行关联, GraphDetect 为 From-To 解算类, 可直接调用 Detect() 方法进行解算。

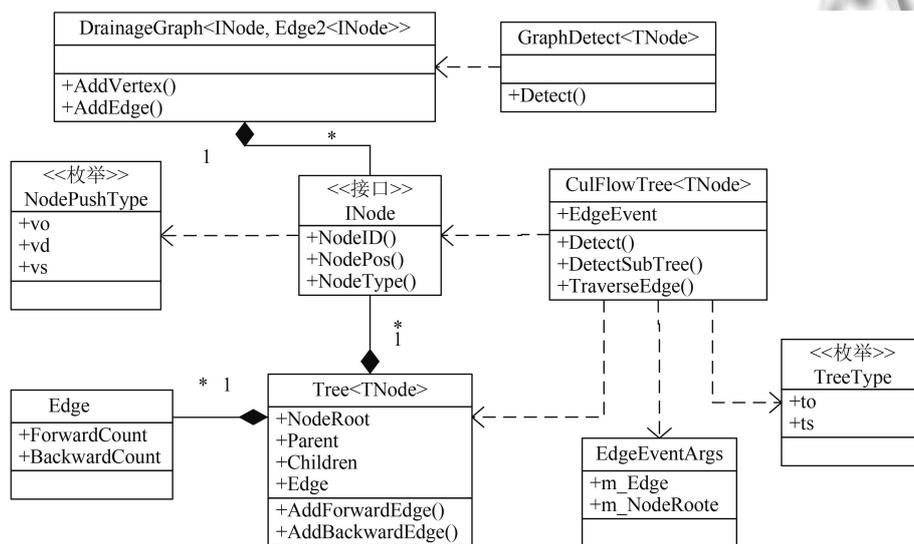


图4 图探测法的类图结构

3.2 在矿井排水仿真系统中的应用

某大型地下煤矿因地质降水渗水较多, 加之周边

有3个水库, 涌水量高达 3800 m³/h, 属于特大型涌水矿井. 为保证矿井安全回采, 该矿在回采前通过探放水

等方案提前放水,形成了5200方/小时的外排能力,并设置中央排水泵房5个、采区临时泵房2个,安设管路40余万米、出水点18处,水泵240个,闸阀总计530个,总排水能力9300 m³/h. 本文将该矿井下排水线路网络进行了三维仿真和From-To解算,系统能够根据排水网络中各水泵、闸阀等工作状态,自动计算各管道的水流来源和去向,解算过程中仅需将整个排水网络转化为流向图,然后按照本文的From-To解算方法进行解算. 具体地,首先将开启时的水泵看作驱动节点 vd ,关闭时的水泵看作闭节点 vs ;打开的闸阀看作通节点 vo ,关闭的闸看作闭节点 vs ;最终出口看作出口节点 ve ;并将信息持久化存储为如图5、图6所示的关

系表中,同时将这些数据动态加载到DrainageGraph对象中,然后利用GraphDetect类进行From-To解算,效果如图7所示.

节点ID	节点类型	工作状态	x	y	z	大小比例	角度
node130505160826	接头	1	2.265	-6.536	0.000	0.500	0.000
node130505160913	接头	1	-1.468	-7.866	0.000	0.500	0.000
node130505161006	接头	1	3.381	-6.536	0.000	0.500	0.000
node130505161106	接头	1	5.464	-6.536	0.000	0.500	0.000
node130510152355	接头	1	1.396	-6.536	0.000	0.350	0.000
node130510152807	接头	1	1.557	-6.536	0.000	0.500	0.000
node130510153639	接头	1	1.934	-6.536	0.000	0.500	0.000
node130510154033	接头	1	3.820	-7.414	0.000	0.500	0.000
node130510154434	接头	1	4.892	-5.742	0.000	0.500	0.000
node130604154628	接头	1	-2.365	1.002	0.000	0.500	0.000

图5 排水系统中节点集合的部分数据截图

编号	源点ID	汇点ID	管道类型	接头1长度	接头2长度	RGB
node130614081420node130624144224	node130614081420	node130624144224	108.000	0.050	0.100	0,255,0
node130614081549node130614092336	node130614081549	node130614092336	108.000	0.050	0.100	0,255,0
node130614081622node130702085408	node130614081622	node130702085408	108.000	0.050	0.100	0,255,0
node130614081651node130702085109	node130614081651	node130702085109	108.000	0.050	0.100	0,255,0
node130615100005node130716101232	node130615100005	node130716101232	108.000	0.050	0.100	0,255,0
node130615100420node130716143329	node130615100420	node130716143329	108.000	0.050	0.100	0,255,0
node130615100648node130716101550	node130615100648	node130716101550	108.000	0.050	0.100	0,255,0
node130615100744node130716101920	node130615100744	node130716101920	108.000	0.050	0.100	0,255,0
node130615100859node130716100708	node130615100859	node130716100708	108.000	0.050	0.100	0,255,0
node130615100952node130716100742	node130615100952	node130716100742	108.000	0.050	0.100	0,255,0

图6 排水系统中管道集合的部分数据截图

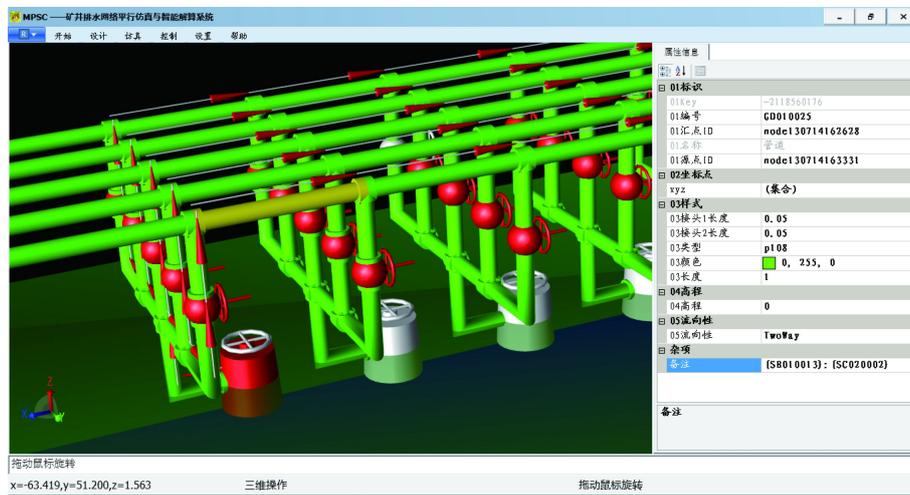


图7 From-To解算应用效果

图5所示表结构中存储的是排水网络转换为流向图的节点数据,包括节点ID、节点类型、工作状态、x、y和z坐标等信息,这些节点主要指接头、水泵和闸阀等对象.图6所示表结构中存储的是排水网络转换为流向图的边信息,包括编号、源点ID(边的来源节点ID)、汇点ID(边的去向节点ID)和管道类型等信

息,这些边主要指排水管道.

图7中,红色的水泵和闸阀指示正在工作和打开状态,白色的水泵和闸阀指示停止工作和关闭状态,水泵和闸阀之间的绿色边为排水管道,红色箭头为解算后的水流方向,管道边的属性信息中自动记录了管道的From-To信息,如对于图中的黄色管道其From-

To 信息为: {SB010013};{SC020002}. 解算效果较好.

4 结论

主要研究了复杂传输网络中流体来源去向的 From-To 解算问题, 相关结论如下:

(1) 相对于文献[8]所提出的“树形探测法”而言, “From-To 解算法”不仅使得“流向计算”不再受限于树结构, 让“普通图”的解算成为可能, 且其解算结果带有更为详细的 From 和 To 信息, 可较为完美地辅助决策支持, 其原理严谨可靠, 时间复杂度为 $O(n)-O(n!)$, 执行速度较快.

(2) 该算法在某大型地下煤矿排水网络调度中发挥了重要作用, 应用效果较好, 实际上该方法可推广到大多传输网络的智能解算中, 如通风、油气、车流甚至供电等.

参考文献

- 1 何杨, 王建中, 鲁仁全. 基于两个节点排水管网水力学模型的新方法. 化工学报, 2010, 61(8): 1901-1904.
- 2 王东云, 徐艳平, 瞿博阳. 基于改进蜂群算法的机器人路径规划. 计算机系统应用, 2017, 26(2): 145-150.
- 3 Zhang JX, Liang MG, Wang ZW. Efficient algorithm for finding multi-paths in telecommunication networks. Key Engineering Materials, 2011, 474-476: 2274-2278. [doi: 10.4028/www.scientific.net/KEM.474-476]
- 4 Kumar CN, Satyanarayana N. Multipath QoS routing for traffic splitting in MANETs. Procedia Computer Science, 2015, 48: 414-426. [doi: 10.1016/j.procs.2015.04.115]
- 5 赵德群, 段建英, 陈鹏宇, 等. 基于 A*算法的三维地图最优路径规划. 计算机系统应用, 2017, 26(7): 146-152.
- 6 Qureshi AH, Ayaz Y. Potential functions based sampling heuristic for optimal path planning. Autonomous Robots, 2016, 40(6): 1079-1093. [doi: 10.1007/s10514-015-9518-0]
- 7 Feng RJ, Li TL, Wu YF, et al. Reliable routing in wireless sensor networks based on coalitional game theory. IET Communications, 2016, 10(9): 1027-1034. [doi: 10.1049/iet-com.2015.0884]
- 8 冯治东, 卢才武, 张培元. 树形探测法及其在矿井排水智能流向计算中的应用. 计算机工程与应用, 2014, 50(10): 227-232. [doi: 10.3778/j.issn.1002-8331.1309-0288]
- 9 胡运权. 运筹学教程. 4 版. 北京: 清华大学出版社, 2012.
- 10 Weiss MA. Data Structures and Algorithm Analysis in C++. 3rd ed. New Jersey: Addison Wesley, 2007.
- 11 Meier JD, Homer A, Hill D, et al. Application architecture guide 2.0. <http://msdn.microsoft.com/en-us/library/dd673617.aspx>, 2009.
- 12 Stellman A, Greene J. Head First C#. New York: O'Reilly Media Inc., 2007.
- 13 Pelikhan. QuickGraph, graph data structures and algorithms for .NET. <http://quickgraph.codeplex.com/documentation>, 2009.