



(12)发明专利

(10)授权公告号 CN 105139685 B

(45)授权公告日 2017.06.06

(21)申请号 201510451828.3

审查员 沈紫乐

(22)申请日 2015.07.24

(65)同一申请的已公布的文献号

申请公布号 CN 105139685 A

(43)申请公布日 2015.12.09

(73)专利权人 合肥工业大学

地址 230009 安徽省合肥市包河区屯溪路  
193号

(72)发明人 郭丹 胡学钢 倪武 吴信东

(74)专利代理机构 安徽省合肥新安专利代理有  
限责任公司 34101

代理人 陆丽莉 何梅生

(51)Int.Cl.

G08G 1/0968(2006.01)

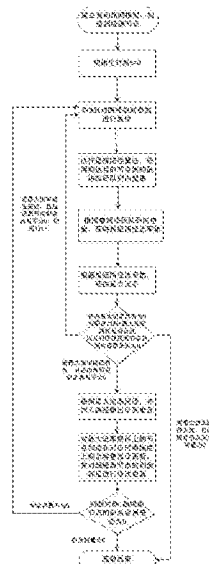
权利要求书6页 说明书17页 附图3页

(54)发明名称

一种基于最大流率路径优先的路网疏散规划方法

(57)摘要

本发明公开了一种基于最大流率路径优先的路网疏散规划方法,是应用于模拟有向路网结构中,其特征是如下步骤进行:1、构造超级源节点并与有向路网结构进行关联;2路网疏散规划参数定义;3疏散规划的路径选择。本发明综合考虑了所求疏散路径的疏散时间和疏散容量两方面影响,以求在每次路径选择时,总是能选择出在最短疏散时间内疏散最多容量的疏散路径,从而能提高该路网疏散规划的疏散效率。



1. 一种基于最大流率路径优先的路网疏散规划方法,是应用于模拟有向路网结构 $G(V, E)$ 中,在所述模拟有向路网结构 $G(V, E)$ 中, $V$ 表示节点集,并有 $V = \{V_s, V_z, V_d\}$ ;  $V_s$ 表示源节点集,  $V_z$ 表示中间节点集;  $V_d$ 表示目的节点集; 并有:  $V_s = \{V_1, V_2, \dots, V_i, \dots, V_a\}$ ;  $V_i$ 表示所述源节点集 $V_s$ 中第 $i$ 个源节点,且 $V_i = \{V_{i, cap}, V_{i, occ}\}$ ,  $V_{i, cap}$ 表示所述源节点集 $V_s$ 中第 $i$ 个源节点 $V_i$ 的节点通行容量,  $V_{i, occ}$ 表示所述源节点集 $V_s$ 中第 $i$ 个源节点 $V_i$ 的初始待疏散容量,  $1 \leq i \leq a$ ;  $V_z = \{V_{a+1}, \dots, V_j, \dots, V_b\}$ ,  $V_j$ 表示所述中间节点集 $V_z$ 中第 $j$ 个中间节点,且 $V_j = \{V_{j, cap}\}$ ,  $V_{j, cap}$ 表示所述中间节点集 $V_z$ 中第 $j$ 个中间节点 $V_j$ 的节点通行容量,  $a+1 \leq j \leq b$ ;  $V_d = \{V_{b+1}, \dots, V_k, \dots, V_c\}$ ,  $V_k$ 表示所述目的节点集 $V_d$ 中第 $k$ 个目的节点,且 $V_k = \{V_{k, cap}, V_{k, occ}\}$ ,  $V_{k, cap}$ 表示所述目的节点集 $V_d$ 中第 $k$ 个目的节点 $V_k$ 的节点通行容量,  $V_{k, occ}$ 表示所述目的节点集 $V_d$ 中第 $k$ 个目的节点 $V_k$ 的可容纳疏散容量,  $b+1 \leq k \leq c$ ;  $E$ 表示边集, 并有 $E = \{e_1, e_2, \dots, e_h, \dots, e_H\}$ ,  $e_h$ 表示所述边集 $E$ 中第 $h$ 条边, 假设所述第 $h$ 条边 $e_h$ 是由所述节点集 $V$ 中第 $n_h$ 个节点 $V_{n_h}$ 出发到与其相连的第 $n_h'$ 个节点 $V_{n_h'}$ 结束所生成的边, 并有  $e_h = \left\{ e_{cap}^{(V_{n_h}, V_{n_h'})}, e_{tra}^{(V_{n_h}, V_{n_h'})} \right\}$ ,  $e_{cap}^{(V_{n_h}, V_{n_h'})}$ 表示第 $h$ 条边 $e_h$ 的通行容量,  $e_{tra}^{(V_{n_h}, V_{n_h'})}$ 表示通过第 $h$ 条边 $e_h$ 的通行时间; 其特征是, 所述路网疏散规划方法按如下步骤进行:

步骤1、构造超级源节点 $V_0$ 并与所述有向路网结构 $G(V, E)$ 进行关联:

步骤1.1、构建超级源节点 $V_0$ , 并有 $V_0 = \{V_{0, cap}\}$ ,  $V_{0, cap} = \sum_{i=1}^a V_{i, occ}$ 表示所述超级源节点 $V_0$ 的疏散容量;

步骤1.2、将所述超级源节点 $V_0$ 分别与所述源节点集 $V_s$ 中的每个源节点相连, 从而生成 $a$ 条边, 记为 $E_{V_0} = \{e'_1, e'_2, \dots, e'_i, \dots, e'_a\}$ ;  $e'_i$ 表示所述超级源节点 $V_0$ 与所述源节点集 $V_s$ 中第 $i$ 个源节点 $V_i$ 相连所生成的边, 并有:  $e'_i = \left\{ e_{cap}^{(V_0, V_i)}, e_{tra}^{(V_0, V_i)} \right\}$ ,  $e_{cap}^{(V_0, V_i)}$ 表示所述超级源节点 $V_0$ 与所述源节点集 $V_s$ 中第 $i$ 个源节点 $V_i$ 相连所生成的边 $e'_i$ 的通行容量,  $e_{tra}^{(V_0, V_i)}$ 表示通过所述超级源节点 $V_0$ 与所述源节点集 $V_s$ 中第 $i$ 个源节点 $V_i$ 相连所生成的边 $e'_i$ 的通行时间;

步骤1.3、将所述源节点集 $V_s$ 中第 $i$ 个源节点 $V_i$ 的初始待疏散容量 $V_{i, occ}$ 赋值给 $e_{cap}^{(V_0, V_i)}$ , 并令 $e_{tra}^{(V_0, V_i)} = 0$ , 实现所述超级源节点 $V_0$ 与所述源节点集 $V_s$ 中第 $i$ 个源节点 $V_i$ 的关联; 从而实现所述超级源节点 $V_0$ 与所述模拟有向路网结构 $G(V, E)$ 的关联;

步骤2、路网疏散规划参数定义:

步骤2.1、路径规划参数定义:

定义时刻为 $t$ , 并用于记录所述疏散规划的时间进程, 初始化 $t=0$ ;

定义带时间维度的路径遍历有向路网结构为 $G(V, E)'$ , 并用于求解疏散最短路径;

定义 $t$ 时刻的路径代价向量为 $D_t$ , 并有 $D_t = \{D_{t, b+1}, D_{t, b+2}, \dots, D_{t, k}, \dots, D_{t, c}\}$ ,  $D_{t, k}$ 表示所述路径代价向量 $D_t$ 中的第 $k$ 个分量, 并用于存储 $t$ 时刻从所述超级源节点 $V_0$ 到所述目的节点集 $V_d$ 中第 $k$ 个目的节点 $V_k$ 所花费的最小时间代价;

定义t时刻的路径选择队列向量为 $Path_t$ ,并有 $Path_t = \{P_{t,b+1}, P_{t,b+2}, \dots, P_{t,k}, \dots, P_{t,c}\}$ ;  $P_{t,k}$ 表示所述路径选择队列向量 $Path_t$ 中的第k个分量,并用于存储t时刻从所述超级源节点 $V_0$ 到所述目的节点集 $V_d$ 中第k个目的节点 $V_k$ 所花费的最小时间代价 $D_{t,k}$ 所对应的路径,并记为最短路径;

定义t时刻的路径流率向量为 $Rate_t$ ,并有 $Rate_t = \{R_{t,b+1}, R_{t,b+2}, \dots, R_{t,k}, \dots, R_{t,c}\}$ ,  $R_{t,k}$ 表示所述路径流率向量 $Rate_t$ 中的第k个分量,并用于存储t时刻所述路径选择队列向量 $Path_t$ 中的第k个分量 $P_{t,k}$ 上容纳的最大通行容量与通行时间的比率,即从所述超级源节点 $V_0$ 到所述目的节点集 $V_d$ 中第k个目的节点 $V_k$ 的最短路径 $P_{t,k}$ 的流率;

定义t时刻的路径流率向量 $Rate_t$ 中最大分量为最大流率 $R_{t,max}$ ;

定义t时刻由所述路径选择队列向量 $Path_t$ 中的最大流率 $R_{t,max}$ 所对应的最短路径,为最大流率路径,记为 $P_{t,max}$ ;

定义路径输出队列集合为 $OutPut\_Path$ ,并初始化所述路径输出队列集合 $OutPut\_Path$ 为空;

步骤2.2、路网状态参数定义并初始化:

定义t时刻通行容量节点集为 $V_{cap}^{(t)}$ ,用来表示所述节点集 $V$ 中所有节点在t时刻的剩余通行容量,并有 $V_{cap}^{(t)} = \bigcup_{1 \leq m \leq c} V_{cap}^{(V_m, t)}$ ,  $V_{cap}^{(V_m, t)}$ 表示所述节点集 $V$ 中第m个节点 $V_m$ 在t时刻的剩余通行容量;

当 $t=0$ 时,初始化 $V_{cap}^{(V_m, 0)} = V_{m, cap}$ ,  $V_{m, cap}$ 表示所述节点集 $V$ 中第m个节点 $V_m$ 的通行容量,从而获得 $t=0$ 时刻的通行容量节点集 $V_{cap}^{(0)}$ ;

定义剩余容量时间扩展节点集为 $V_{cap}$ ,将 $t=0$ 时刻的通行容量节点集 $V_{cap}^{(0)}$ 存入所述剩余容量时间扩展节点集 $V_{cap}$ 中;

合并所述边集 $E$ 和边集 $E_{V_0}$ 中所有边,构成边全集 $E_{union}$ ;

定义t时刻通行容量边集为 $E_{cap}^{(t)}$ ,用于表示所述边全集 $E_{union}$ 中各条边在t时刻的剩余通行容量,并有 $E_{cap}^{(t)} = \bigcup_{1 \leq r \leq R} e_{cap}^{(r, t)}$ , 且 $R=H+a$ ;  $e_{cap}^{(r, t)} = \{e_{cap}^{(V_{n_r}, V_{n_r'})^{(t)}}\}$ 表示所述边全集 $E_{union}$ 中第r条边在t时刻的剩余通行容量,假设第r条边的剩余通行容量 $e_{cap}^{(r, t)}$ 是由所述节点集 $V$ 和所述超级源节点 $V_0$ 中第 $n_r$ 个节点 $V_{n_r}$ 出发到与其相连的第 $n_r'$ 个节点 $V_{n_r'}$ 结束所生成的边在t时刻的剩余通行容量,并有 $V_{n_r}, V_{n_r'} \in V \cup V_0$ ;

$t=0$ 时刻时,将所述边全集 $E_{union}$ 中从第 $n_r$ 个节点 $V_{n_r}$ 出发到与其相连的第 $n_r'$ 个节点 $V_{n_r'}$ 结束所生成的边的通行容量 $e_{cap}^{(V_{n_r}, V_{n_r'})}$ 赋值给 $e_{cap}^{(V_{n_r}, V_{n_r'})^{(0)}}$ ,从而获得 $t=0$ 时刻的通行容量边集 $E_{cap}^{(0)}$ ;

定义剩余容量时间扩展边集为 $E_{cap}$ ,将 $t=0$ 时刻的通行容量边集 $E_{cap}^{(0)}$ 存入所述剩余容量

时间扩展边集 $E_{cap}$ 中;

步骤3、疏散规划的路径选择:

步骤3.1、在 $t$ 时刻,进行所述路径规划参数复位操作:

步骤3.1.1、参数复位:

设置 $t$ 时刻的路径代价向量 $D_t = \{D_{t,b+1}, D_{t,b+2}, \dots, D_{t,k}, \dots, D_{t,c}\}$ 中各分量值均为正无穷大;设置 $t$ 时刻的路径选择队列向量 $Path_t = \{P_{t,b+1}, P_{t,b+2}, \dots, P_{t,k}, \dots, P_{t,c}\}$ 中各向量值均为空;设置 $t$ 时刻的路径流率向量 $Rate_t = \{R_{t,b+1}, R_{t,b+2}, \dots, R_{t,k}, \dots, R_{t,c}\}$ 中各分量值均为空;设置 $t$ 时刻的最大流率 $R_{t,max} = 0$ ;设置 $t$ 时刻的最大流率路径 $P_{t,max}$ 为空;

步骤3.1.2、带时间维度的路径遍历有向路网结构 $G(V,E)'$ 复位:

将所述有向路网结构 $G(V,E)$ 赋值给 $G(V,E)'$ ;

根据所述剩余容量时间扩展节点集 $V_{cap}$ 中剩余通行容量为0的节点获得所对应的的时间维度,并判定在 $G(V,E)'$ 中剩余通行容量为0的节点在所对应的的时间维度上为无效;

根据所述剩余容量时间扩展边集 $E_{cap}$ 中剩余通行容量为0的边获得所对应的的时间维度,并判定在 $G(V,E)'$ 中剩余通行容量为0的边在所对应的的时间维度上为无效;从而获得带时间维度的路径遍历有向路网结构 $G(V,E)'$ ;

步骤3.2、根据所述带时间维度的路径遍历有向路网结构 $G(V,E)'$ 中节点和边是否为无效,遍历所述带时间维度的路径遍历有向路网结构 $G(V,E)'$ 中从所述超级源节点 $V_0$ 到所述目的节点集 $V_d$ 中第 $k$ 个目的节点 $V_k$ 所经过的所有路径;

选取到达第 $k$ 个目的节点 $V_k$ 所经过的所有路径中花费时间代价最小的路径作为第 $k$ 个目的节点 $V_k$ 的最短路径并分别存入 $t$ 时刻的路径选择队列向量 $Path_t$ 的第 $k$ 个分量 $P_{t,k}$ 中;从而将所述目的节点集 $V_d$ 中所有目的节点的最短路径分别存入所述路径选择队列向量 $Path_t$ 相应的分量中;

步骤3.3、将 $t$ 时刻路径选择队列向量 $Path_t$ 中第 $k$ 个分量 $P_{t,k}$ 所花费的时间代价存入所述路径代价向量 $D_t$ 的第 $k$ 个分量 $D_{t,k}$ 中,从而将所述目的节点集 $V_d$ 中所有目的节点的最短路径所花费时间代价分别存入所述路径代价向量 $D_t$ 相应的分量中;

步骤3.4、获得 $t$ 时刻路径选择队列向量 $Path_t$ 中第 $k$ 个分量 $P_{t,k}$ 所表示的最短路径上允许通行的最大疏散容量 $F_{P_{t,k}}$ ;从而获得在 $t$ 时刻从所述超级源节点 $V_0$ 到所述目的节点集 $V_d$ 中所有目的节点的最短路径上的最大疏散容量集 $\{F_{P_{t,b+1}}, F_{P_{t,b+2}}, \dots, F_{P_{t,k}}, \dots, F_{P_{t,c}}\}$ ;

步骤3.4.1、获得 $t$ 时刻路径选择队列向量 $Path_t$ 中第 $k$ 个分量 $P_{t,k}$ 所表示的最短路径,记为 $P_{t,k} = \{V_0^{P_{t,k}}, V_1^{P_{t,k}}, \dots, V_{n_{t,k}-1}^{P_{t,k}}, V_{n_{t,k}}^{P_{t,k}}, \dots, V_{N_{t,k}}^{P_{t,k}}\}$ ,并有 $V_0^{P_{t,k}} = V_0, V_{n_{t,k}}^{P_{t,k}} \in V$ ;  $N_{t,k}$ 表示所述最短路径 $P_{t,k}$ 中除超级源节点 $V_0$ 外所经过的节点总数;  $V_{n_{t,k}}^{P_{t,k}}$ 表示所述最短路径 $P_{t,k}$ 中所经过的第 $n_{t,k}$ 个节点,  $1 \leq n_{t,k} \leq N_{t,k}$ ;

步骤3.4.2、利用式(1)获得所述最短路径 $P_{t,k}$ 中所经过第0个节点 $V_0^{P_{t,k}}$ 以及经过的第 $n_{t,k}$ 个节点 $V_{n_{t,k}}^{P_{t,k}}$ 的到达时间 $T_{n_{t,k}}^{P_{t,k}}$ ;从而获得所述最短路径 $P_{t,k}$ 中所经过的节点的到达时间集 $\{T_0^{P_{t,k}}, T_1^{P_{t,k}}, \dots, T_{n_{t,k}}^{P_{t,k}}, \dots, T_{N_{t,k}}^{P_{t,k}}\}$ ;

$$\begin{cases} T_0^{P_{t,k}} = t, \\ T_{n_{t,k}}^{P_{t,k}} = T_{n_{t,k}-1}^{P_{t,k}} + e_{tra}^{(V_{n_{t,k}-1}^{P_{t,k}}, V_{n_{t,k}}^{P_{t,k}})}, \quad \forall n_{t,k}, 1 \leq n_{t,k} \leq N_{t,k} \end{cases} \quad (1)$$

式(1)中,  $e_{tra}^{(V_{n_{t,k}-1}^{P_{t,k}}, V_{n_{t,k}}^{P_{t,k}})}$  表示在所述边全集  $E_{union}$  中, 由所述最短路径  $P_{t,k}$  中从第  $n_{t,k}-1$  个节点  $V_{n_{t,k}-1}^{P_{t,k}}$  出发到与其相连的第  $n_{t,k}$  个节点  $V_{n_{t,k}}^{P_{t,k}}$  结束所生成的边的通行时间;

步骤3.4.3、更新路网状态参数:

步骤3.4.3.1、根据所述最短路径  $P_{t,k}$  中所经过第  $n_{t,k}$  个节点  $V_{n_{t,k}}^{P_{t,k}}$  的到达时间  $T_{n_{t,k}}^{P_{t,k}}$ , 查询所述剩余容量时间扩展节点集  $V_{cap}$  中是否包含到达时间  $T_{n_{t,k}}^{P_{t,k}}$  的通行容量节点集  $V_{cap}^{(T_{n_{t,k}}^{P_{t,k}})}$ , 若不包含, 则令  $V_{cap}^{(T_{n_{t,k}}^{P_{t,k}})}$  为空并存入所述剩余容量时间扩展节点集  $V_{cap}$  中; 否则, 维持  $V_{cap}^{(T_{n_{t,k}}^{P_{t,k}})}$  不变;

查询所述通行容量节点集  $V_{cap}^{(T_{n_{t,k}}^{P_{t,k}})}$  中是否包含所述第  $n_{t,k}$  个节点  $V_{n_{t,k}}^{P_{t,k}}$  在到达时间  $T_{n_{t,k}}^{P_{t,k}}$  时的剩余通行节点容量  $V_{cap}^{(V_{n_{t,k}}^{P_{t,k}}, T_{n_{t,k}}^{P_{t,k}})}$ , 若包含, 则维持  $V_{cap}^{(V_{n_{t,k}}^{P_{t,k}}, T_{n_{t,k}}^{P_{t,k}})}$  不变; 否则, 将所述最短路径  $P_{t,k}$  中所经过第  $n_{t,k}$  个节点  $V_{n_{t,k}}^{P_{t,k}}$  在所述节点集  $V$  中所对应节点的通行容量  $V_{n_{t,k}, cap}$  赋值给  $V_{cap}^{(V_{n_{t,k}}^{P_{t,k}}, T_{n_{t,k}}^{P_{t,k}})}$ , 将  $V_{cap}^{(V_{n_{t,k}}^{P_{t,k}}, T_{n_{t,k}}^{P_{t,k}})}$  存入所述通行容量节点集  $V_{cap}^{(T_{n_{t,k}}^{P_{t,k}})}$  中; 从而实现将所述剩余通行节点容量  $V_{cap}^{(V_{n_{t,k}}^{P_{t,k}}, T_{n_{t,k}}^{P_{t,k}})}$  存入所述剩余容量时间扩展节点集  $V_{cap}$  中;

步骤3.4.3.2、重复执行步骤3.4.3.1, 从而获得所述最短路径  $P_{t,k}$  中所经过的节点在各自到达时间维度上的剩余节点容量, 并依次存入所述剩余容量时间扩展节点集  $V_{cap}$  中;

步骤3.4.3.3、根据所述最短路径  $P_{t,k}$  中所经过从第  $n_{t,k}-1$  个节点  $V_{n_{t,k}-1}^{P_{t,k}}$  出发到与其相连的第  $n_{t,k}$  个节点  $V_{n_{t,k}}^{P_{t,k}}$  结束所生成的边的到达时间  $T_{n_{t,k}-1}^{P_{t,k}}$ , 查询所述剩余容量时间扩展边集  $E_{cap}$  中是否包含到达时间  $T_{n_{t,k}-1}^{P_{t,k}}$  时的通行容量边集  $E_{cap}^{(T_{n_{t,k}-1}^{P_{t,k}})}$ , 若不包含, 令  $E_{cap}^{(T_{n_{t,k}-1}^{P_{t,k}})}$  为空并存入所述剩余容量时间扩展边集  $E_{cap}$  中; 否则, 维持  $E_{cap}^{(T_{n_{t,k}-1}^{P_{t,k}})}$  不变;

查询所述通行容量边集  $E_{cap}^{(T_{n_{t,k}-1}^{P_{t,k}})}$  中是否包含所述最短路径  $P_{t,k}$  中从第  $n_{t,k}-1$  个节点  $V_{n_{t,k}-1}^{P_{t,k}}$  出发到与其相连的第  $n_{t,k}$  个节点  $V_{n_{t,k}}^{P_{t,k}}$  结束所生成的边在到达时间  $T_{n_{t,k}-1}^{P_{t,k}}$  时的剩余通行边容量  $e_{cap}^{(V_{n_{t,k}-1}^{P_{t,k}}, V_{n_{t,k}}^{P_{t,k}}), T_{n_{t,k}-1}^{P_{t,k}})}$ , 若包含, 则维持  $e_{cap}^{(V_{n_{t,k}-1}^{P_{t,k}}, V_{n_{t,k}}^{P_{t,k}}), T_{n_{t,k}-1}^{P_{t,k}})}$  不变; 否则, 将所述边全集  $E_{union}$  中由所述最短路径  $P_{t,k}$  中从第  $n_{t,k}-1$  个节点  $V_{n_{t,k}-1}^{P_{t,k}}$  出发到与其相连的第  $n_{t,k}$  个节点  $V_{n_{t,k}}^{P_{t,k}}$  结束所生成的边的通行容量  $e_{cap}^{(V_{n_{t,k}-1}^{P_{t,k}}, V_{n_{t,k}}^{P_{t,k}}), T_{n_{t,k}-1}^{P_{t,k}})}$  赋值给  $e_{cap}^{(V_{n_{t,k}-1}^{P_{t,k}}, V_{n_{t,k}}^{P_{t,k}}), T_{n_{t,k}-1}^{P_{t,k}})}$ , 将  $e_{cap}^{(V_{n_{t,k}-1}^{P_{t,k}}, V_{n_{t,k}}^{P_{t,k}}), T_{n_{t,k}-1}^{P_{t,k}})}$  存入所述通行容量边集  $E_{cap}^{(T_{n_{t,k}-1}^{P_{t,k}})}$  中,

从而实现将所述剩余通行边容量  $e_{cap}^{(V_{n_{t,k}-1}^{P_{t,k}}, V_{n_{t,k}}^{P_{t,k}}) T_{n_{t,k}-1}^{P_{t,k}}}$  存入所述剩余容量时间扩展边集  $E_{cap}$  中；

步骤3.4.3.4、重复执行步骤3.4.3.3,从而获得所述最短路径  $P_{t,k}$  中所经过的边在各自到达时间维度上的剩余边容量,并依次存入所述剩余容量时间扩展边集  $E_{cap}$  中；

步骤3.4.4、根据式(2)获得所述路径选择队列向量  $Path_t$  中的第  $k$  个分量  $P_{t,k}$  所代表的最短路径上允许通行的最大疏散容量  $F_{P_{t,k}}$ ：

$$F_{P_{t,k}} = \min \begin{cases} V_{0,cap} \\ V_{cap}^{(V_{n_{t,k}}^{P_{t,k}}, T_{n_{t,k}}^{P_{t,k}})} \\ e_{cap}^{(V_{n_{t,k}-1}^{P_{t,k}}, V_{n_{t,k}}^{P_{t,k}}) T_{n_{t,k}-1}^{P_{t,k}}} \end{cases} \quad (2)$$

s.t.  $\forall n_{t,k}, 1 \leq n_{t,k} \leq N_{t,k}$

式(2)中,  $V_{cap}^{(V_{n_{t,k}}^{P_{t,k}}, T_{n_{t,k}}^{P_{t,k}})}$  表示所述最短路径  $P_{t,k}$  上的第  $n_{t,k}$  个节点  $V_{n_{t,k}}^{P_{t,k}}$  在到达时间  $T_{n_{t,k}}^{P_{t,k}}$  时的剩余通行容量； $e_{cap}^{(V_{n_{t,k}-1}^{P_{t,k}}, V_{n_{t,k}}^{P_{t,k}}) T_{n_{t,k}-1}^{P_{t,k}}}$  表示从所述最短路径  $P_{t,k}$  上的第  $n_{t,k}-1$  个节点  $V_{n_{t,k}-1}^{P_{t,k}}$  出发到与其相连的第  $n_{t,k}$  个节点  $V_{n_{t,k}}^{P_{t,k}}$  结束所生成的边在到达时间  $T_{n_{t,k}-1}^{P_{t,k}}$  时的剩余通行容量；

步骤3.5、利用式(3)获得  $t$  时刻从所述超级源节点  $V_0$  到所述目的节点集  $V_d$  中第  $k$  个目的节点  $V_k$  的最短路径的流率  $R_{t,k}$ ,从而获得  $t$  时刻从所述超级源节点  $V_0$  到所述目的节点集  $V_d$  中所有目的节点的最短路径的流率向量  $Rate_t = \{R_{t,b+1}, R_{t,b+2}, \dots, R_{t,k}, \dots, R_{t,c}\}$ ；并获得所述  $t$  时刻的路径流率向量  $Rate_t$  中最大流率  $R_{t,max}$ ：

$$R_{t,k} = F_{P_{t,k}} / D_{t,k} \quad (3)$$

步骤3.6、判断  $R_{t,max}=0$  和  $V_{0,cap}=0$  是否成立,若都成立,则执行步骤3.10；

若只有  $R_{t,max}=0$  成立,则表示  $t$  时刻路径选择队列向量  $Path_t$  无效,且在  $t$  时刻不能再次进行疏散,将  $t+1$  的值赋给  $t$ ,并返回步骤3.1重新计算疏散规划路径；

若都不成立,则表示时刻路径选择队列向量  $Path_t$  有效,并从所述路径选择队列向量  $Path_t$  中将最大流率  $R_{t,max}$  所对应的最短路径选出,存入所述最大流率路径  $P_{t,max}$ ,并记为

$P_{t,max} = \{V_0^{P_{t,max}}, V_1^{P_{t,max}}, \dots, V_{n'_{t,max}-1}^{P_{t,max}}, V_{n'_{t,max}}^{P_{t,max}}, \dots, V_{N'_{t,max}}^{P_{t,max}}\}$ ,以及所述最大流率路径  $P_{t,max}$  中所经过节点的到达时间集记为  $\{T_0^{P_{t,max}}, T_1^{P_{t,max}}, \dots, T_{n'_{t,max}-1}^{P_{t,max}}, T_{n'_{t,max}}^{P_{t,max}}, \dots, T_{N'_{t,max}}^{P_{t,max}}\}$ ,并有  $V_0^{P_{t,max}} = V_0, V_{n'_{t,max}}^{P_{t,max}} \in V$ ；

$N'_{t,max}$  表示所述最大流率路径  $P_{t,max}$  中除超级源节点  $V_0$  外所经过的节点总数,  $V_{n'_{t,max}}^{P_{t,max}}$  表示所述最大流率路径  $P_{t,max}$  中所经过的第  $n'_{t,max}$  个节点,  $1 \leq n'_{t,max} \leq N'_{t,max}$ ；记所述最大流率路径  $P_{t,max}$  所对应的最大疏散容量为  $F_{t,max}$ ；将所述最大流率路径  $P_{t,max}$  中除所述超级源节点  $V_0$  外路径所经过的节点存入所述路径输出队列集合  $OutPut\_Path$  中,并记录最大疏散容量和各节点的到达时间；并执行步骤3.7；

步骤3.7、利用式(4)分别对所述最大流率路径  $P_{t,max}$  中所经过的节点和边的剩余通行容量进行修改,将修改值重新存入所述剩余容量时间扩展节点集  $V_{cap}$  和所述剩余容量时间扩

展边集 $E_{cap}$ 中,从而实现所述剩余容量时间扩展节点集 $V_{cap}$ 和所述剩余容量时间扩展边集 $E_{cap}$ 的更新;

$$\begin{cases} e_{cap}^{(V_{n'_{t,max}-1}^{P_{t,max}}, V_{n'_{t,max}}^{P_{t,max}}), T_{n'_{t,max}-1}^{P_{t,max}}} = e_{cap}^{(V_{n'_{t,max}-1}^{P_{t,max}}, V_{n'_{t,max}}^{P_{t,max}}), T_{n'_{t,max}-1}^{P_{t,max}}} - F_{t,max} \\ V_{cap}^{(V_{n'_{t,max}}^{P_{t,max}}, T_{n'_{t,max}}^{P_{t,max}})} = V_{cap}^{(V_{n'_{t,max}}^{P_{t,max}}, T_{n'_{t,max}}^{P_{t,max}})} - F_{t,max} \end{cases} \quad (4)$$

s. t.  $n'_{t,max} \leq n'_{t,max} \leq N'_{t,max}$

式(4)中,  $e_{cap}^{(V_{n'_{t,max}-1}^{P_{t,max}}, V_{n'_{t,max}}^{P_{t,max}}), T_{n'_{t,max}-1}^{P_{t,max}}}$  表示所述最大流率路径 $P_{t,max}$ 上的第 $n'_{t,max}-1$ 个节点 $V_{n'_{t,max}-1}^{P_{t,max}}$  出发到与其相连的第 $n'_{t,max}$ 个节点 $V_{n'_{t,max}}^{P_{t,max}}$  结束所生成的边在到达时间 $T_{n'_{t,max}-1}^{P_{t,max}}$  时的剩余通行容量;  $V_{cap}^{(V_{n'_{t,max}}^{P_{t,max}}, T_{n'_{t,max}}^{P_{t,max}})}$  表示所述最大流率路径 $P_{t,max}$ 上的第 $n'_{t,max}$ 个节点 $V_{n'_{t,max}}^{P_{t,max}}$  在到达时间 $T_{n'_{t,max}}^{P_{t,max}}$  时的剩余通行容量;

步骤3.8、利用式(5)获得所述超级源节点 $V_0$ 在 $t$ 时刻的剩余通行容量 $V_{0,cap}$ :

$$V_{0,cap} = V_{0,cap} - F_{t,max} \quad (5)$$

步骤3.9、判断 $V_{0,cap} = 0$ 是否成立,若成立,则执行步骤3.10;否则,执行步骤3.1;

步骤3.10、所述疏散规划结束,获得所述存储了最大流率路径的路径输出队列集合 $OutPut\_Path$ 的最终结果。

## 一种基于最大流率路径优先的路网疏散规划方法

### 技术领域

[0001] 本发明属于路径规划、信息处理、地理信息学领域,并涉及到图论的应用,具体地说是一种基于最大流率路径优先的路网疏散规划方法。

### 背景技术

[0002] 当今社会,大城市的人口密度越来越大,城市交通拥堵的情况也越来越突出,此外,城镇人口的密集化带来了在灾害事件发生时能否及时疏散来降低危害影响的问题,因此,路网疏散问题逐渐成为人们亟待解决的一个课题。如何能够进行科学的路径规划,来指导和控制待疏散区域内人口的行动路径,以尽可能大的疏散效率和尽可能小的疏散时间将受灾区域的群众安全地转移到指定的区域,具有重要的现实意义。

[0003] 针对这种情况,人们提出了很多种算法模型来解决这类问题,例如传统的线性规划和模拟仿真等方法。后来,在学术界也出现了一种名为启发式算法的方法,来解决这类疏散问题。一时间很多种相类似的启发式算法云涌而出,例如贪婪算法,回溯算法,逆流算法等等层出不穷。前几年,卢青松教授在MRCCP算法的基础之上提出了一种CCRP的启发式算法思想,在限制疏散容量的情况下每次总是让疏散路径短的路径优先进行疏散,实践证明,CCRP的效率比MRCCP的效率有了一定的提高。后来,有学者在CCRP的基础之上,提出了多种改进的算法,有的是长路径优先算法,有的提出智能减载算法。然而,这些算法思想所生成的疏散路径总是因为不能充分利用路径上剩余容量的资源,延长了每次疏散路径选择的时间和疏散时间,因此不可避免的会影响到总的疏散时间,从而导致路网疏散规划的疏散效率低下。

### 发明内容

[0004] 本发明为克服现有技术的不足之处,提出了一种基于最大流率路径优先的路网疏散规划方法,综合考虑了所求疏散路径的疏散时间和疏散容量两方面影响,以求在每次路径选择时,总是能选择出在最短疏散时间内疏散最多容量的疏散路径,从而能提高该路网疏散规划的疏散效率。

[0005] 本发明为解决以上技术问题采用如下技术方案:

[0006] 本发明一种基于最大流率路径优先的路网疏散规划方法,是应用于模拟有向路网结构 $G(V,E)$ 中,在所述模拟有向路网结构 $G(V,E)$ 中, $V$ 表示节点集,并有 $V = \{V_s, V_z, V_d\}$ ;  $V_s$ 表示源节点集,  $V_z$ 表示中间节点集;  $V_d$ 表示目的节点集; 并有:  $V_s = \{V_1, V_2, \dots, V_i, \dots, V_a\}$ ;  $V_i$ 表示所述源节点集 $V_s$ 中第 $i$ 个源节点,且 $V_i = \{V_{i,cap}, V_{i,occ}\}$ ,  $V_{i,cap}$ 表示所述源节点集 $V_s$ 中第 $i$ 个源节点 $V_i$ 的节点通行容量,  $V_{i,occ}$ 表示所述源节点集 $V_s$ 中第 $i$ 个源节点 $V_i$ 的初始待疏散容量,  $1 \leq i \leq a$ ;  $V_z = \{V_{a+1}, \dots, V_j, \dots, V_b\}$ ,  $V_j$ 表示所述中间节点集 $V_z$ 中第 $j$ 个中间节点,且 $V_j = \{V_{j,cap}\}$ ,  $V_{j,cap}$ 表示所述中间节点集 $V_z$ 中第 $j$ 个中间节点 $V_j$ 的节点通行容量,  $a+1 \leq j \leq b$ ;  $V_d = \{V_{b+1}, \dots, V_k, \dots, V_c\}$ ,  $V_k$ 表示所述目的节点集 $V_d$ 中第 $k$ 个目的节点,且 $V_k = \{V_{k,cap}, V_{k,occ}\}$ ,  $V_{k,cap}$ 表示所述目的节点集 $V_d$ 中第 $k$ 个目的节点 $V_k$ 的节点通行容量,  $V_{k,occ}$ 表示所述目的节点集 $V_d$ 中第 $k$ 个目



的节点 $V_k$ 的可容纳疏散容量,  $b+1 \leq k \leq c$ ;  $E$ 表示边集, 并有  $E = \{e_1, e_2, \dots, e_h, \dots, e_H\}$ ,  $e_h$ 表示所述边集 $E$ 中第 $h$ 条边, 假设所述第 $h$ 条边 $e_h$ 是由所述节点集 $V$ 中第 $n_h$ 个节点 $V_{n_h}$ 出发到与其相连的第 $n_h'$ 个节点 $V_{n_h'}$ 结束所生成的边, 并有  $e_h = \left\{ e_{cap}^{(V_{n_h}, V_{n_h'})}, e_{tra}^{(V_{n_h}, V_{n_h'})} \right\}$ ,  $e_{cap}^{(V_{n_h}, V_{n_h'})}$ 表示第 $h$ 条边 $e_h$ 的通行容量,  $e_{tra}^{(V_{n_h}, V_{n_h'})}$ 表示通过第 $h$ 条边 $e_h$ 的通行时间; 其特点是, 所述路网疏散规划方法按如下步骤进行:

[0007] 步骤1、构造超级源节点 $V_0$ 并与所述有向路网结构 $G(V, E)$ 进行关联:

[0008] 步骤1.1、构建超级源节点 $V_0$ , 并有  $V_0 = \{V_{0, cap}\}$ ,  $V_{0, cap} = \sum_{i=1}^a V_{i, occ}$ 表示所述超级源节点 $V_0$ 的疏散容量;

[0009] 步骤1.2、将所述超级源节点 $V_0$ 分别与所述源节点集 $V_s$ 中的每个源节点相连, 从而生成 $a$ 条边, 记为  $E_{V_0} = \{e'_1, e'_2, \dots, e'_i, \dots, e'_a\}$ ;  $e'_i$ 表示所述超级源节点 $V_0$ 与所述源节点集 $V_s$ 中第 $i$ 个源节点 $V_i$ 相连所生成的边, 并有:  $e'_i = \left\{ e_{cap}^{(V_0, V_i)}, e_{tra}^{(V_0, V_i)} \right\}$ ,  $e_{cap}^{(V_0, V_i)}$ 表示所述超级源节点 $V_0$ 与所述源节点集 $V_s$ 中第 $i$ 个源节点 $V_i$ 相连所生成的边 $e'_i$ 的通行容量,  $e_{tra}^{(V_0, V_i)}$ 表示通过所述超级源节点 $V_0$ 与所述源节点集 $V_s$ 中第 $i$ 个源节点 $V_i$ 相连所生成的边 $e'_i$ 的通行时间;

[0010] 步骤1.3、将所述源节点集 $V_s$ 中第 $i$ 个源节点 $V_i$ 的初始待疏散容量 $V_{i, occ}$ 赋值给  $e_{cap}^{(V_0, V_i)}$ , 并令  $e_{tra}^{(V_0, V_i)} = 0$ , 实现所述超级源节点 $V_0$ 与所述源节点集 $V_s$ 中第 $i$ 个源节点 $V_i$ 的关联; 从而实现所述超级源节点 $V_0$ 与所述模拟有向路网结构 $G(V, E)$ 的关联;

[0011] 步骤2、路网疏散规划参数定义:

[0012] 步骤2.1、路径规划参数定义:

[0013] 定义时刻为 $t$ , 并用于记录所述疏散规划的时间进程, 初始化 $t=0$ ;

[0014] 定义带时间维度的路径遍历有向路网结构为 $G(V, E)'$ , 并用于求解疏散最短路径;

[0015] 定义 $t$ 时刻的路径代价向量为 $D_t$ , 并有  $D_t = \{D_{t, b+1}, D_{t, b+2}, \dots, D_{t, k}, \dots, D_{t, c}\}$ ,  $D_{t, k}$ 表示所述路径代价向量 $D_t$ 中的第 $k$ 个分量, 并用于存储 $t$ 时刻从所述超级源节点 $V_0$ 到所述目的节点集 $V_d$ 中第 $k$ 个目的节点 $V_k$ 所花费的最小时间代价;

[0016] 定义 $t$ 时刻的路径选择队列向量为 $Path_t$ , 并有  $Path_t = \{P_{t, b+1}, P_{t, b+2}, \dots, P_{t, k}, \dots, P_{t, c}\}$ ;  $P_{t, k}$ 表示所述路径选择队列向量 $Path_t$ 中的第 $k$ 个分量, 并用于存储 $t$ 时刻从所述超级源节点 $V_0$ 到所述目的节点集 $V_d$ 中第 $k$ 个目的节点 $V_k$ 所花费的最小时间代价 $D_{t, k}$ 所对应的路径, 并记为最短路径;

[0017] 定义 $t$ 时刻的路径流率向量为 $Rate_t$ , 并有  $Rate_t = \{R_{t, b+1}, R_{t, b+2}, \dots, R_{t, k}, \dots, R_{t, c}\}$ ,  $R_{t, k}$ 表示所述路径流率向量 $Rate_t$ 中的第 $k$ 个分量, 并用于存储 $t$ 时刻所述路径选择队列向量 $Path_t$ 中的第 $k$ 个分量 $P_{t, k}$ 上容纳的最大通行容量与通行时间的比率, 即从所述超级源节点 $V_0$ 到所述目的节点集 $V_d$ 中第 $k$ 个目的节点 $V_k$ 的最短路径 $P_{t, k}$ 的流率;

[0018] 定义 $t$ 时刻的路径流率向量 $Rate_t$ 中最大分量为最大流率 $R_{t, max}$ ;

[0019] 定义 $t$ 时刻由所述路径选择队列向量 $Path_t$ 中的最大流率 $R_{t, max}$ 所对应的最短路径,

为最大流率路径,记为 $P_{t,\max}$ ;

[0020] 定义路径输出队列集合为 $OutPut\_Path$ ,并初始化所述路径输出队列集合 $OutPut\_Path$ 为空;

[0021] 步骤2.2、路网状态参数定义并初始化:

[0022] 定义t时刻通行容量节点集为 $V_{cap}^{(t)}$ ,用来表示所述节点集V中所有节点在t时刻的剩余通行容量,并有 $V_{cap}^{(t)} = \bigcup_{1 \leq m \leq c} V_{cap}^{(V_m, t)}$ ,  $V_{cap}^{(V_m, t)}$ 表示所述节点集V中第m个节点 $V_m$ 在t时刻的剩余通行容量;

[0023] 当 $t=0$ 时,初始化 $V_{cap}^{(V_m, 0)} = V_{m, cap}$ ,  $V_{m, cap}$ 表示所述节点集V中第m个节点 $V_m$ 的通行容量,从而获得 $t=0$ 时刻的通行容量节点集 $V_{cap}^{(0)}$ ;

[0024] 定义剩余容量时间扩展节点集为 $V_{cap}$ ,将 $t=0$ 时刻的通行容量节点集 $V_{cap}^{(0)}$ 存入所述剩余容量时间扩展节点集 $V_{cap}$ 中;

[0025] 合并所述边集E和边集 $E_{V_0}$ 中所有边,构成边全集 $E_{union}$ ;

[0026] 定义t时刻通行容量边集为 $E_{cap}^{(t)}$ ,用于表示所述边全集 $E_{union}$ 中各条边在t时刻的剩余通行容量,并有 $E_{cap}^{(t)} = \bigcup_{1 \leq r \leq R} e_{cap}^{(r, t)}$ ,且 $R=H+a$ ;  $e_{cap}^{(r, t)} = \{e_{cap}^{(V_{n_r}, V_{n_r'})} \}$ 表示所述边全集 $E_{union}$ 中第r条边在t时刻的剩余通行容量,假设第r条边的剩余通行容量 $e_{cap}^{(r, t)}$ 是由所述节点集V和所述超级源节点 $V_0$ 中第 $n_r$ 个节点 $V_{n_r}$ 出发到与其相连的第 $n_r'$ 个节点 $V_{n_r'}$ 结束所生成的边在t时刻的剩余通行容量,并有 $V_{n_r}, V_{n_r'} \in V \cup V_0$ ;

[0027]  $t=0$ 时刻时,将所述边全集 $E_{union}$ 中从第 $n_r$ 个节点 $V_{n_r}$ 出发到与其相连的第 $n_r'$ 个节点 $V_{n_r'}$ 结束所生成的边的通行容量 $e_{cap}^{(V_{n_r}, V_{n_r'})}$ 赋值给 $e_{cap}^{(V_{n_r}, V_{n_r'})}^{(0)}$ ,从而获得 $t=0$ 时刻的通行容量边集 $E_{cap}^{(0)}$ ;

[0028] 定义剩余容量时间扩展边集为 $E_{cap}$ ,将 $t=0$ 时刻的通行容量边集 $E_{cap}^{(0)}$ 存入所述剩余容量时间扩展边集 $E_{cap}$ 中;

[0029] 步骤3、疏散规划的路径选择:

[0030] 步骤3.1、在t时刻,进行所述路径规划参数复位操作:

[0031] 步骤3.1.1、参数复位:

[0032] 设置t时刻的路径代价向量 $D_t = \{D_{t, b+1}, D_{t, b+2}, \dots, D_{t, k}, \dots, D_{t, c}\}$ 中各分量值均为正无穷大;设置t时刻的路径选择队列向量 $Path_t = \{P_{t, b+1}, P_{t, b+2}, \dots, P_{t, k}, \dots, P_{t, c}\}$ 中各向量值均为空;设置t时刻的路径流率向量 $Rate_t = \{R_{t, b+1}, R_{t, b+2}, \dots, R_{t, k}, \dots, R_{t, c}\}$ 中各分量值均为空;设置t时刻的最大流率 $R_{t, \max} = 0$ ;设置t时刻的最大流率路径 $P_{t, \max}$ 为空;

[0033] 步骤3.1.2、带时间维度的路径遍历有向路网结构 $G(V, E)$ '复位:

[0034] 将所述有向路网结构 $G(V, E)$ 赋值给 $G(V, E)'$ ;

[0035] 根据所述剩余容量时间扩展节点集 $V_{cap}$ 中剩余通行容量为0的节点获得所对应的

时间维度,并判定在 $G(V,E)'$ 中剩余通行容量为0的节点在所对应的的时间维度上为无效;

[0036] 根据所述剩余容量时间扩展边集 $E_{cap}$ 中剩余通行容量为0的边获得所对应的的时间维度,并判定在 $G(V,E)'$ 中剩余通行容量为0的边在所对应的的时间维度上为无效;从而获得带时间维度的路径遍历有向路网结构 $G(V,E)'$ ;

[0037] 步骤3.2、根据所述带时间维度的路径遍历有向路网结构 $G(V,E)'$ 中节点和边是否为无效,遍历所述带时间维度的路径遍历有向路网结构 $G(V,E)'$ 中从所述超级源节点 $V_0$ 到所述目的节点集 $V_d$ 中第 $k$ 个目的节点 $V_k$ 所经过的所有路径;

[0038] 选取到达第 $k$ 个目的节点 $V_k$ 所经过的所有路径中花费时间代价最小的路径作为第 $k$ 个目的节点 $V_k$ 的最短路径并分别存入 $t$ 时刻的路径选择队列向量 $Path_t$ 的第 $k$ 个分量 $P_{t,k}$ 中;从而将所述目的节点集 $V_d$ 中所有目的节点的最短路径分别存入所述路径选择队列向量 $Path_t$ 相应的分量中;

[0039] 步骤3.3、将 $t$ 时刻路径选择队列向量 $Path_t$ 中第 $k$ 个分量 $P_{t,k}$ 所花费的时间代价存入所述路径代价向量 $D_t$ 的第 $k$ 个分量 $D_{t,k}$ 中,从而将所述目的节点集 $V_d$ 中所有目的节点的最短路径所花费时间代价分别存入所述路径代价向量 $D_t$ 相应的分量中;

[0040] 步骤3.4、获得 $t$ 时刻路径选择队列向量 $Path_t$ 中第 $k$ 个分量 $P_{t,k}$ 所表示的最短路径上允许通行的最大疏散容量 $F_{Pt,k}$ ;从而获得在 $t$ 时刻从所述超级源节点 $V_0$ 到所述目的节点集 $V_d$ 中所有目的节点的最短路径上的最大疏散容量集 $\{F_{Pt,b+1}, F_{Pt,b+2}, \dots, F_{Pt,k}, \dots, F_{Pt,c}\}$ ;

[0041] 步骤3.4.1、获得 $t$ 时刻路径选择队列向量 $Path_t$ 中第 $k$ 个分量 $P_{t,k}$ 所表示的最短路径,记为 $P_{t,k} = \{V_0^{P_{t,k}}, V_1^{P_{t,k}}, \dots, V_{n_{t,k}-1}^{P_{t,k}}, V_{n_{t,k}}^{P_{t,k}}, \dots, V_{N_{t,k}}^{P_{t,k}}\}$ ,并有 $V_0^{P_{t,k}} = V_0$ ,  $V_{n_{t,k}}^{P_{t,k}} \in V$ ;  $N_{t,k}$ 表示所述最短路径 $P_{t,k}$ 中除超级源节点 $V_0$ 外所经过的节点总数;  $V_{n_{t,k}}^{P_{t,k}}$ 表示所述最短路径 $P_{t,k}$ 中所经过的第 $n_{t,k}$ 个节点,  $1 \leq n_{t,k} \leq N_{t,k}$ ;

[0042] 步骤3.4.2、利用式(1)获得所述最短路径 $P_{t,k}$ 中所经过第0个节点 $V_0^{P_{t,k}}$ 以及经过的第 $n_{t,k}$ 个节点 $V_{n_{t,k}}^{P_{t,k}}$ 的到达时间 $T_{n_{t,k}}^{P_{t,k}}$ ;从而获得所述最短路径 $P_{t,k}$ 中所经过的节点的到达时间集 $\{T_0^{P_{t,k}}, T_1^{P_{t,k}}, \dots, T_{n_{t,k}}^{P_{t,k}}, \dots, T_{N_{t,k}}^{P_{t,k}}\}$ ;

$$[0043] \quad \begin{cases} T_0^{P_{t,k}} = t, \\ T_{n_{t,k}}^{P_{t,k}} = T_{n_{t,k}-1}^{P_{t,k}} + e_{\langle V_{n_{t,k}-1}^{P_{t,k}}, V_{n_{t,k}}^{P_{t,k}} \rangle}, \quad \forall n_{t,k}, 1 \leq n_{t,k} \leq N_{t,k} \end{cases} \quad (1)$$

[0044] 式(1)中,  $e_{\langle V_{n_{t,k}-1}^{P_{t,k}}, V_{n_{t,k}}^{P_{t,k}} \rangle}$ 表示在所述边全集 $E_{union}$ 中,由所述最短路径 $P_{t,k}$ 中从第 $n_{t,k}-1$ 个节点 $V_{n_{t,k}-1}^{P_{t,k}}$ 出发到与其相连的第 $n_{t,k}$ 个节点 $V_{n_{t,k}}^{P_{t,k}}$ 结束所生成的边的通行时间;

[0045] 步骤3.4.3、更新路网状态参数:

[0046] 步骤3.4.3.1、根据所述最短路径 $P_{t,k}$ 中所经过第 $n_{t,k}$ 个节点 $V_{n_{t,k}}^{P_{t,k}}$ 的到达时间 $T_{n_{t,k}}^{P_{t,k}}$ ,查询所述剩余容量时间扩展节点集 $V_{cap}$ 中是否包含到达时间 $T_{n_{t,k}}^{P_{t,k}}$ 的通行容量节点集 $V_{cap}^{(T_{n_{t,k}}^{P_{t,k}})}$ ,

若不包含,则令  $V_{cap}^{(T_{n_t,k}^{P_{t,k}})}$  为空并存入所述剩余容量时间扩展节点集  $V_{cap}$  中;否则,维持  $V_{cap}^{(T_{n_t,k}^{P_{t,k}})}$  不变;

[0047] 查询所述通行容量节点集  $V_{cap}^{(T_{n_t,k}^{P_{t,k}})}$  中是否包含所述第  $n_{t,k}$  个节点  $V_{n_{t,k}}^{P_{t,k}}$  在到达时间  $T_{n_{t,k}}^{P_{t,k}}$  时的剩余通行节点容量  $V_{cap}^{(V_{n_{t,k}}^{P_{t,k}}, T_{n_{t,k}}^{P_{t,k}})}$ ,若包含,则维持  $V_{cap}^{(V_{n_{t,k}}^{P_{t,k}}, T_{n_{t,k}}^{P_{t,k}})}$  不变;否则,将所述最短路径  $P_{t,k}$  中所经过第  $n_{t,k}$  个节点  $V_{n_{t,k}}^{P_{t,k}}$  在所述节点集  $V$  中所对应节点的通行容量  $V_{n_{t,k},cap}$  赋值给  $V_{cap}^{(V_{n_{t,k}}^{P_{t,k}}, T_{n_{t,k}}^{P_{t,k}})}$ ,将  $V_{cap}^{(V_{n_{t,k}}^{P_{t,k}}, T_{n_{t,k}}^{P_{t,k}})}$  存入所述通行容量节点集  $V_{cap}^{(T_{n_t,k}^{P_{t,k}})}$  中;从而实现将所述剩余通行节点容量  $V_{cap}^{(V_{n_{t,k}}^{P_{t,k}}, T_{n_{t,k}}^{P_{t,k}})}$  存入所述剩余容量时间扩展节点集  $V_{cap}$  中;

[0048] 步骤3.4.3.2、重复执行步骤3.4.3.1,从而获得所述最短路径  $P_{t,k}$  中所经过的节点在各自到达时间维度上的剩余节点容量,并依次存入所述剩余容量时间扩展节点集  $V_{cap}$  中;

[0049] 步骤3.4.3.3、根据所述最短路径  $P_{t,k}$  中所经过从第  $n_{t,k-1}$  个节点  $V_{n_{t,k-1}}^{P_{t,k}}$  出发到与其相连的第  $n_{t,k}$  个节点  $V_{n_{t,k}}^{P_{t,k}}$  结束所生成的边的到达时间  $T_{n_{t,k-1}}^{P_{t,k}}$ ,查询所述剩余容量时间扩展边集  $E_{cap}$  中是否包含到达时间  $T_{n_{t,k-1}}^{P_{t,k}}$  时的通行容量边集  $E_{cap}^{(T_{n_{t,k-1}}^{P_{t,k}})}$ ,若不包含,令  $E_{cap}^{(T_{n_{t,k-1}}^{P_{t,k}})}$  为空并存入所述剩余容量时间扩展边集  $E_{cap}$  中;否则,维持  $E_{cap}^{(T_{n_{t,k-1}}^{P_{t,k}})}$  不变;

[0050] 查询所述通行容量边集  $E_{cap}^{(T_{n_{t,k-1}}^{P_{t,k}})}$  中是否包含所述最短路径  $P_{t,k}$  中从第  $n_{t,k-1}$  个节点  $V_{n_{t,k-1}}^{P_{t,k}}$  出发到与其相连的第  $n_{t,k}$  个节点  $V_{n_{t,k}}^{P_{t,k}}$  结束所生成的边在到达时间  $T_{n_{t,k-1}}^{P_{t,k}}$  时的剩余通行边容量  $e_{cap}^{(V_{n_{t,k-1}}^{P_{t,k}}, V_{n_{t,k}}^{P_{t,k}}; T_{n_{t,k-1}}^{P_{t,k}})}$ ,若包含,则维持  $e_{cap}^{(V_{n_{t,k-1}}^{P_{t,k}}, V_{n_{t,k}}^{P_{t,k}}; T_{n_{t,k-1}}^{P_{t,k}})}$  不变;否则,将所述边全集  $E_{union}$  中由所述最短路径  $P_{t,k}$  中从第  $n_{t,k-1}$  个节点  $V_{n_{t,k-1}}^{P_{t,k}}$  出发到与其相连的第  $n_{t,k}$  个节点  $V_{n_{t,k}}^{P_{t,k}}$  结束所生成的边的通行容量  $e_{cap}^{(V_{n_{t,k-1}}^{P_{t,k}}, V_{n_{t,k}}^{P_{t,k}})}$  赋值给  $e_{cap}^{(V_{n_{t,k-1}}^{P_{t,k}}, V_{n_{t,k}}^{P_{t,k}}; T_{n_{t,k-1}}^{P_{t,k}})}$ ,将  $e_{cap}^{(V_{n_{t,k-1}}^{P_{t,k}}, V_{n_{t,k}}^{P_{t,k}}; T_{n_{t,k-1}}^{P_{t,k}})}$  存入所述通行容量边集  $E_{cap}^{(T_{n_{t,k-1}}^{P_{t,k}})}$  中,从而实现将所述剩余通行边容量  $e_{cap}^{(V_{n_{t,k-1}}^{P_{t,k}}, V_{n_{t,k}}^{P_{t,k}}; T_{n_{t,k-1}}^{P_{t,k}})}$  存入所述剩余容量时间扩展边集  $E_{cap}$  中;

[0051] 步骤3.4.3.4、重复执行步骤3.4.3.3,从而获得所述最短路径  $P_{t,k}$  中所经过的边在各自到达时间维度上的剩余边容量,并依次存入所述剩余容量时间扩展边集  $E_{cap}$  中;

[0052] 步骤3.4.4、根据式(2)获得所述路径选择队列向量  $Path_t$  中的第  $k$  个分量  $P_{t,k}$  所代表的最短路径上允许通行的最大疏散容量  $F_{P_{t,k}}$ ;

$$\begin{aligned}
 [0053] \quad F_{P_{t,k}} = \min & \begin{cases} V_{0,cap} \\ V_{cap}^{(V_{n_{t,k}}^{P_{t,k}}, T_{n_{t,k}}^{P_{t,k}})} \\ e_{cap}^{(\{V_{n_{t,k-1}}^{P_{t,k}}, V_{n_{t,k}}^{P_{t,k}}\}, T_{n_{t,k-1}}^{P_{t,k}})} \end{cases} \\
 \text{s. t.} \quad \forall n_{t,k}, & 1 \leq n_{t,k} \leq N_{t,k}
 \end{aligned} \quad (2)$$

[0054] 式(2)中,  $V_{cap}^{(V_{n_{t,k}}^{P_{t,k}}, T_{n_{t,k}}^{P_{t,k}})}$  表示所述最短路径 $P_{t,k}$ 上的第 $n_{t,k}$ 个节点 $V_{n_{t,k}}^{P_{t,k}}$ 在到达时间 $T_{n_{t,k}}^{P_{t,k}}$ 时的剩余通行容量;  $e_{cap}^{(\{V_{n_{t,k-1}}^{P_{t,k}}, V_{n_{t,k}}^{P_{t,k}}\}, T_{n_{t,k-1}}^{P_{t,k}})}$ 表示从所述最短路径 $P_{t,k}$ 上的第 $n_{t,k-1}$ 个节点 $V_{n_{t,k-1}}^{P_{t,k}}$ 出发到与其相连的第 $n_{t,k}$ 个节点 $V_{n_{t,k}}^{P_{t,k}}$ 结束所生成的边在到达时间 $T_{n_{t,k-1}}^{P_{t,k}}$ 时的剩余通行容量;

[0055] 步骤3.5、利用式(3)获得t时刻从所述超级源节点 $V_0$ 到所述目的节点集 $V_d$ 中第k个目的节点 $V_k$ 的最短路径的流率 $R_{t,k}$ ,从而获得t时刻从所述超级源节点 $V_0$ 到所述目的节点集 $V_d$ 中所有目的节点的最短路径的流率向量  $Rate_t = \{R_{t,d1}, R_{t,d2}, \dots, R_{t,k}, \dots, R_{t,c}\}$ ;并获得所述t时刻的路径流率向量 $Rate_t$ 中最大流率 $R_{t,max}$ :

$$[0056] \quad R_{t,k} = F_{P_{t,k}} / D_{t,k} \quad (3)$$

[0057] 步骤3.6、判断 $R_{t,max}=0$ 和 $V_{0,cap}=0$ 是否成立,若都成立,则执行步骤3.10;

[0058] 若只有 $R_{t,max}=0$ 成立,则表示t时刻路径选择队列向量 $Path_t$ 无效,且在t时刻不能再次进行疏散,将t+1的值赋给t,并返回步骤3.1重新计算疏散规划路径;

[0059] 若都不成立,则表示时刻路径选择队列向量 $Path_t$ 有效,并从所述路径选择队列向量 $Path_t$ 中将最大流率 $R_{t,max}$ 所对应的最短路径选出,存入所述最大流率路径 $P_{t,max}$ ,并记为

$P_{t,max} = \{V_0^{P_{t,max}}, V_1^{P_{t,max}}, \dots, V_{n_{t,max}^{P_{t,max}}-1}^{P_{t,max}}, V_{n_{t,max}^{P_{t,max}}}^{P_{t,max}}, \dots, V_{N_{t,max}^{P_{t,max}}}^{P_{t,max}}\}$ ,以及所述最大流率路径 $P_{t,max}$ 中所经过节点的到达时间集记为  $\{T_0^{P_{t,max}}, T_1^{P_{t,max}}, \dots, T_{n_{t,max}^{P_{t,max}}-1}^{P_{t,max}}, T_{n_{t,max}^{P_{t,max}}}^{P_{t,max}}, \dots, T_{N_{t,max}^{P_{t,max}}}^{P_{t,max}}\}$ ,并有  $V_0^{P_{t,max}} = V_0, V_{n_{t,max}^{P_{t,max}}}^{P_{t,max}} \in V$ ;

$N_{t,max}^{P_{t,max}}$ 表示所述最大流率路径 $P_{t,max}$ 中除超级源节点 $V_0$ 外所经过的节点总数,  $V_{n_{t,max}^{P_{t,max}}}^{P_{t,max}}$ 表示所述最大流率路径 $P_{t,max}$ 中所经过的第 $n_{t,max}^{P_{t,max}}$ 个节点,  $1 \leq n_{t,max}^{P_{t,max}} \leq N_{t,max}^{P_{t,max}}$ ;记所述最大流率路径 $P_{t,max}$ 所对应的最大疏散容量为 $F_{t,max}$ ;将所述最大流率路径 $P_{t,max}$ 中除所述超级源节点 $V_0$ 外路径所经过的节点存入所述路径输出队列集合 $OutPut\_Path$ 中,并记录最大疏散容量和各节点的到达时间;并执行步骤3.7;

[0060] 步骤3.7、利用式(4)分别对所述最大流率路径 $P_{t,max}$ 中所经过的节点和边的剩余通行容量进行修改,将修改值重新存入所述剩余容量时间扩展节点集 $V_{cap}$ 和所述剩余容量时间扩展边集 $E_{cap}$ 中,从而实现所述剩余容量时间扩展节点集 $V_{cap}$ 和所述剩余容量时间扩展边集 $E_{cap}$ 的更新;

$$\begin{cases}
 e_{cap}^{(V_{n_{t,max}^{i-1}}^{t,P_{t,max}}, V_{n_{t,max}^{i-1}}^{t,P_{t,max}})T_{n_{t,max}^{i-1}}^{t,P_{t,max}}} = e_{cap}^{(V_{n_{t,max}^{i-1}}^{t,P_{t,max}}, V_{n_{t,max}^{i-1}}^{t,P_{t,max}})T_{n_{t,max}^{i-1}}^{t,P_{t,max}}} - F_{t,max} \\
 V_{cap}^{(V_{n_{t,max}^{i-1}}^{t,P_{t,max}}, T_{n_{t,max}^{i-1}}^{t,P_{t,max}})} = V_{cap}^{(V_{n_{t,max}^{i-1}}^{t,P_{t,max}}, T_{n_{t,max}^{i-1}}^{t,P_{t,max}})} - F_{t,max} \\
 \text{s.t. } n_{t,max}^{i-1} \leq n_{t,max}^i \leq N_{t,max}^i
 \end{cases} \quad (4)$$

[0062] 式(4)中,  $e_{cap}^{(V_{n_{t,max}^{i-1}}^{t,P_{t,max}}, V_{n_{t,max}^{i-1}}^{t,P_{t,max}})T_{n_{t,max}^{i-1}}^{t,P_{t,max}}}$  表示所述最大流率路径 $P_{t,max}$ 上的第 $n_{t,max}^{i-1}$ 个节点 $V_{n_{t,max}^{i-1}}^{t,P_{t,max}}$ 出发到与其相连的第 $n_{t,max}^i$ 个节点 $V_{n_{t,max}^i}^{t,P_{t,max}}$ 结束所生成的边在到达时间 $T_{n_{t,max}^{i-1}}^{t,P_{t,max}}$ 时的剩余通行容量; $V_{cap}^{(V_{n_{t,max}^{i-1}}^{t,P_{t,max}}, T_{n_{t,max}^{i-1}}^{t,P_{t,max}})}$ 表示所述最大流率路径 $P_{t,max}$ 上的第 $n_{t,max}^{i-1}$ 个节点 $V_{n_{t,max}^{i-1}}^{t,P_{t,max}}$ 在到达时间 $T_{n_{t,max}^{i-1}}^{t,P_{t,max}}$ 时的剩余通行容量;

[0063] 步骤3.8、利用式(5)获得所述超级源节点 $V_0$ 在 $t$ 时刻的剩余通行容量 $V_{0,cap}$ :  $V_{0,cap} = V_{0,cap} - F_{t,max}$  (5)

[0064] 步骤3.9、判断 $V_{0,cap} = 0$ 是否成立,若成立,则执行步骤3.10;否则,执行步骤3.1;

[0065] 步骤3.10、所述疏散规划结束,获得所述存储了最大流率路径的路径输出队列集合OutPut\_Path的最终结果。

[0066] 与已有技术相比,本发明的有益效果体现在:

[0067] 1、本发明在已有的CCRP启发式算法思想的基础上,撇开了选择最长疏散路径和最短疏散路径,加入时间维度的概念来动态维护网络结构的状态,减小有效最短路径的选择难度;采用路径流率的度量方法,综合利用疏散路径上的疏散容量和疏散路径所花费的疏散时间,以疏散路径的流率来评价该疏散路径的实用性,争取能够在每一个时刻实现最大的疏散容量;

[0068] 2、本发明在选取从源节点到目的节点的最短路径中,加入时间维度来动态维持最短路径集,在选取目标节点时,优先剔除在对应时间维度下无效的节点,无效的边,减少路径代价。当所有路径遍历完成时,便可以得到一个最优的最短路径集,从而实现了路径代价最少的效果;

[0069] 3、本发明在最短路径的基础之上,利用所求的最短路径在某一时刻的疏散容量和花费时间的比例关系,即流率值,来判断此最短路径的平均疏散能力,选取其中流率值最大的最短路径作为该时刻的疏散路径,实现了路径平均疏散量最大的效果;

[0070] 4、本发明所选的疏散路径是依据路径的疏散容量和花费时间的比例关系,因此路网结构的密集程度对本发明的结果影响程度没有其对已有的疏散规划算法的影响那么大,因此本发明具有更好的适应性和稳定性。

## 附图说明

[0071] 图1为本发明路网模型图;

[0072] 图2为本发明构造的含有超级源节点的路网模型图;

[0073] 图3为本发明的方法流程图。

### 具体实施方式

[0074] 本实施例中,一种基于最大流率路径优先的路网疏散规划方法,是应用于模拟有向路网结构 $G(V,E)$ 中,在模拟有向路网结构 $G(V,E)$ 中, $V$ 表示节点集,并有 $V = \{V_s, V_z, V_d\}$ ,构造结果如图1所示; $V_s$ 表示源节点集, $V_z$ 表示中间节点集; $V_d$ 表示目的节点集;并有: $V_s = \{V_1, V_2, \dots, V_i, \dots, V_a\}$ ;  $V_i$ 表示源节点集 $V_s$ 中第 $i$ 个源节点,且 $V_i = \{V_{i,cap}, V_{i,occ}\}$ ,  $V_{i,cap}$ 表示源节点集 $V_s$ 中第 $i$ 个源节点 $V_i$ 的节点通行容量,  $V_{i,occ}$ 表示源节点集 $V_s$ 中第 $i$ 个源节点 $V_i$ 的初始待疏散容量,  $1 \leq i \leq a$ ,在路网疏散中,源节点集代表的是发生事故且人口比较密集的地点,诸如办公大楼,超市,学校等地点;中间节点集代表的是在疏散过程中遇到的十字交叉口,拐弯地点等;目的节点集代表的是路网疏散中疏散人群的最后聚集点,诸如广场,防空洞等;在本发明中将这地点均模型化为相应的节点;如图1所示,源节点 $V_2 = \{50, 15\}$ ,其中50表示源节点 $V_2$ 的节点通行容量为50,用于描述正常节点的通行容量;15表示源节点 $V_2$ 的初始待疏散容量为15,表示在疏散开始的时候,该节点有多少容量需要进行疏散;通行容量和初始疏散容量可以相等,也可以不相等; $V_z = \{V_{a+1}, \dots, V_j, \dots, V_b\}$ ,  $V_j$ 表示中间节点集 $V_z$ 中第 $j$ 个中间节点,且 $V_j = \{V_{j,cap}\}$ ,  $V_{j,cap}$ 表示中间节点集 $V_z$ 中第 $j$ 个中间节点 $V_j$ 的节点通行容量,  $a+1 \leq j \leq b$ ,如图1所示,中间节点 $V_4 = \{8\}$ ,其中8表示中间节点 $V_4$ 的节点通行容量为8; $V_d = \{V_{b+1}, \dots, V_k, \dots, V_c\}$ ,  $V_k$ 表示目的节点集 $V_d$ 中第 $k$ 个目的节点,且 $V_k = \{V_{k,cap}, V_{k,occ}\}$ ,  $V_{k,cap}$ 表示目的节点集 $V_d$ 中第 $k$ 个目的节点 $V_k$ 的节点通行容量,  $V_{k,occ}$ 表示目的节点集 $V_d$ 中第 $k$ 个目的节点 $V_k$ 的可容纳疏散容量,  $b+1 \leq k \leq c$ ,如图1所示,目的节点 $V_{13} = \{200, 200\}$ ,其中第一个200表示目的节点 $V_{13}$ 的通行容量;第二个200表示目的节点 $V_{13}$ 可以容纳的疏散容量,即在路网疏散中可以接收从源节点疏散而来的容量;通常状态下设置 $V_{k,occ}$ 与 $V_{k,cap}$ 相等; $E$ 表示边集,并有 $E = \{e_1, e_2, \dots, e_h, \dots, e_n\}$ ,  $e_h$ 表示边集 $E$ 中第 $h$ 条边,假设第 $h$ 条边 $e_h$ 是由节点集 $V$ 中第 $n_h$ 个节点 $V_{n_h}$ 出发到与其相连的第 $n_h'$ 个节点 $V_{n_h'}$ 结束所生成的边,  $n_h$ 和 $n_h'$ 分别代表节点 $V_{n_h}$ 和节点 $V_{n_h'}$ 在节点集 $V$ 中的节点编号,并有 $e_h = \left\{ e_{cap}^{(V_{n_h}, V_{n_h'})}, e_{tra}^{(V_{n_h}, V_{n_h'})} \right\}$ ,  $e_{cap}^{(V_{n_h}, V_{n_h'})}$ 表示第 $h$ 条边 $e_h$ 的通行容量,  $e_{tra}^{(V_{n_h}, V_{n_h'})}$ 表示通过第 $h$ 条边 $e_h$ 的通行时间;

[0075] 如图1所示,由中间节点 $V_8$ 出发到中间节点 $V_{11}$ 组成的边的通行容量为 $e_{cap}^{(V_8, V_{11})} = 7$ ,由中间节点 $V_8$ 出发到中间节点 $V_{11}$ 组成的边的通行时间为 $e_{tra}^{(V_8, V_{11})} = 3$ ;如图1所示,描述的是单行道的路网结构;本专利发明还可以用来描述双车道的路网结构,例如可以利用 $e_{cap}^{(V_8, V_{11})}$ 和 $e_{cap}^{(V_{11}, V_8)}$ 分别代表由中间节点 $V_8$ 出发到中间节点 $V_{11}$ 结束的边以及由中间节点 $V_{11}$ 出发到中间节点 $V_8$ 结束的边的通行容量,两者可以相等也可以不相等;具体情形可以依据现实世界中路网真实数据而定;

[0076] 本实施例中,一种基于最大流率路径优先的路网疏散规划方法是按如下过程进行:

[0077] 步骤1、构造超级源节点 $V_0$ 并与有向路网结构 $G(V,E)$ 进行关联;构造结果如图2所示;

[0078] 步骤1.1、构建超级源节点 $V_0$ , 并有 $V_0 = \{V_{0,cap}\}$ ,  $V_{0,cap} = \sum_{i=1}^a V_{i,occ}$  表示超级源节点 $V_0$ 的疏散容量, 如图2所示, 依据源节点 $V_1, V_2$ 和 $V_3$ 的初始待疏散容量之和, 得到 $V_0 = \{30\}$ , 30表示构造的超级源节点 $V_0$ 的疏散容量为30;

[0079] 步骤1.2、将超级源节点 $V_0$ 分别与源节点集 $V_s$ 中的每个源节点相连, 从而生成 $a$ 条边, 记为 $E_{V_0} = \{e_1', e_2', \dots, e_i', \dots, e_a'\}$ ;  $e_i'$ 表示超级源节点 $V_0$ 与源节点集 $V_s$ 中第 $i$ 个源节点 $V_i$ 相连所生成的边, 并有: 并有:  $e_i' = \{e_{cap}^{(V_0, V_i)}, e_{tra}^{(V_0, V_i)}\}$ ,  $e_{cap}^{(V_0, V_i)}$ 表示超级源节点 $V_0$ 与源节点集 $V_s$ 中第 $i$ 个源节点 $V_i$ 相连所生成的边 $e_i'$ 的通行容量,  $e_{tra}^{(V_0, V_i)}$ 表示通过超级源节点 $V_0$ 与源节点集 $V_s$ 中第 $i$ 个源节点 $V_i$ 相连所生成的边 $e_i'$ 的通行时间;

[0080] 步骤1.3、将源节点集 $V_s$ 中第 $i$ 个源节点 $V_i$ 的初始待疏散容量 $V_{occ}^{(i)}$ 赋值给 $e_{cap}^{(0,i)}$ , 并令 $e_{tra}^{(0,i)} = 0$ , 实现超级源节点 $V_0$ 与源节点集 $V_s$ 中第 $i$ 个源节点 $V_i$ 的关联; 从而实现超级源节点 $V_0$ 与模拟有向路网结构 $G(V, E)$ 的关联, 如图2所示, 超级源节点 $V_0$ 与源节点 $V_2$ 相连生的边 $e_2' = \{15, 0\}$ , 其中15表示边 $e_2'$ 的边通行容量 $e_{cap}^{(V_0, V_2)}$ 为15; 0表示通过边 $e_2'$ 所花费的时间代价 $e_{tra}^{(V_0, V_2)}$ 为0;

[0081] 步骤2、路网疏散规划参数定义:

[0082] 步骤2.1、路径规划参数定义:

[0083] 定义时刻为 $t$ , 并用于记录疏散规划的时间进程, 初始化 $t = 0$ ;

[0084] 定义带时间维度的路径遍历有向路网结构 $G(V, E)'$ , 并用于求解疏散最短路径;

[0085] 定义 $t$ 时刻的路径代价向量为 $D_t$ , 并有 $D_t = \{D_{t,b+1}, D_{t,b+2}, \dots, D_{t,k}, \dots, D_{t,c}\}$ ,  $D_{t,k}$ 表示路径代价向量 $D_t$ 中的第 $k$ 个分量, 并用于存储 $t$ 时刻从超级源节点 $V_0$ 到目的节点集 $V_d$ 中第 $k$ 个目的节点 $V_k$ 所花费的最小时间代价;

[0086] 定义 $t$ 时刻的路径选择队列向量为 $Path_t$ , 并有 $Path_t = \{P_{t,b+1}, P_{t,b+2}, \dots, P_{t,k}, \dots, P_{t,c}\}$ ;  $P_{t,k}$ 表示路径选择队列向量 $Path_t$ 中的第 $k$ 个分量, 并用于存储 $t$ 时刻从超级源节点 $V_0$ 到目的节点集 $V_d$ 中第 $k$ 个目的节点 $V_k$ 所花费的最小时间代价 $D_{t,k}$ 所对应的路径, 并记为最短路径;

[0087] 定义 $t$ 时刻的路径流率向量为 $Rate_t$ , 并有 $Rate_t = \{R_{t,b+1}, R_{t,b+2}, \dots, R_{t,k}, \dots, R_{t,c}\}$ ,  $R_{t,k}$ 表示路径流率向量 $Rate_t$ 中的第 $k$ 个分量, 并用于存储 $t$ 时刻路径选择队列向量 $Path_t$ 中的第 $k$ 个分量 $P_{t,k}$ 上容纳的最大通行容量与通行时间的比率, 即从超级源节点 $V_0$ 到目的节点集 $V_d$ 中第 $k$ 个目的节点 $V_k$ 的最短路径 $P_{t,k}$ 的流率;

[0088] 定义 $t$ 时刻的路径流率向量 $Rate_t$ 中最大分量为最大流率 $R_{t,max}$ ;

[0089] 定义 $t$ 时刻由路径选择队列向量 $Path_t$ 中的最大流率 $R_{t,max}$ 所对应的最短路径, 为最大流率路径, 记为 $P_{t,max}$ ;

[0090] 定义路径输出队列集合为 $OutPut\_Path$ , 并初始化路径输出队列集合 $OutPut\_Path$ 为空;

[0091] 步骤2.2、路网状态参数定义并初始化:



[0092] 定义t时刻通行容量节点集为 $V_{cap}^{(t)}$ ,用来表示节点集V中所有节点在t时刻的剩余通行容量,并有 $V_{cap}^{(t)} = \bigcup_{1 \leq m \leq c} V_{cap}^{(V_m, t)}$ ,  $V_{cap}^{(V_m, t)}$ 表示节点集V中第m个节点 $V_m$ 在t时刻的剩余通行容量;

[0093] 当 $t=0$ 时,初始化 $V_{cap}^{(V_m, 0)} = V_{m, cap}$ ,  $V_{m, cap}$ 表示节点集V中第m个节点 $V_m$ 的通行容量,从而获得 $t=0$ 时刻的通行容量节点集 $V_{cap}^{(0)}$ ;

[0094] 定义剩余容量时间扩展节点集为 $V_{cap}$ ,将 $t=0$ 时刻的通行容量节点集 $V_{cap}^{(0)}$ 存入剩余容量时间扩展节点集 $V_{cap}$ 中;

[0095] 合并边集E和由超级源节点 $V_0$ 分别与源节点集 $V_s$ 中的每个源节点相连从而生成的边集 $E_{V_0}$ 中所有边,构成边全集 $E_{union}$ ;

[0096] 定义t时刻通行容量边集为 $E_{cap}^{(t)}$ ,用于表示边全集 $E_{union}$ 中各条边在t时刻的剩余通行容量,并有 $E_{cap}^{(t)} = \bigcup_{1 \leq r \leq R} e_{cap}^{(r, t)}$ , 且 $R=H+a$ ;  $e_{cap}^{(r, t)} = \{e_{cap}^{(V_{n_r}, V_{n_r'})}^{(t)}\}$ 表示边全集 $E_{union}$ 中第r条边在t时刻的剩余通行容量,假设第r条边的剩余通行容量 $e_{cap}^{(r, t)}$ 是由节点集V和超级源节点 $V_0$ 中第 $n_r$ 个节点 $V_{n_r}$ 出发到与其相连的第 $n_r'$ 个节点 $V_{n_r'}$ 结束所生成的边在t时刻的剩余通行容量,并有 $V_{n_r}, V_{n_r'} \in V \cup V_0$ ,  $n_r$ 和 $n_r'$ 分别代表节点 $V_{n_r}$ 和节点 $V_{n_r'}$ 在节点集V和超级源节点 $V_0$ 中的节点编号;

[0097] 当 $t=0$ 时刻时,将边全集 $E_{union}$ 中从第 $n_r$ 个节点 $V_{n_r}$ 出发到与其相连的第 $n_r'$ 个节点 $V_{n_r'}$ 结束所生成的边的通行容量 $e_{cap}^{(V_{n_r}, V_{n_r'})}$ 赋值给 $e_{cap}^{(V_{n_r}, V_{n_r'})}^{(0)}$ ,从而获得 $t=0$ 时刻的通行容量边集 $E_{cap}^{(0)}$ ;

[0098] 定义剩余容量时间扩展边集为 $E_{cap}$ ,将 $t=0$ 时刻的通行容量边集 $E_{cap}^{(0)}$ 存入剩余容量时间扩展边集 $E_{cap}$ 中;

[0099] 步骤3、疏散规划的路径选择:本方明中路径选择方法的流程图如图3所示:

[0100] 步骤3.1、在时刻t,进行路径规划参数复位操作:

[0101] 执行步骤3.1到步骤3.6能进行一条疏散规划路径的计算;针对在相同时刻下可能存在多条疏散路径,即在同一时刻下可能有多个解,需要多次迭代执行步骤3.1到步骤3.6,因此每次计算前需要对下述参数进行复位,保证当前时刻下的当前迭代次数下的路网规划参数的有效性,并计算当前路网结构状态下有效的最大流率路径;从而保证能在同一时刻下最大限度的获取多条不同的疏散路径;

[0102] 步骤3.1.1、参数复位:

[0103] 设置t时刻的路径代价向量 $D_t = \{D_{t, b+1}, D_{t, b+2}, \dots, D_{t, k}, \dots, D_{t, c}\}$ 中各分量值均为正无穷大;设置t时刻的路径选择队列向量 $Path_t = \{P_{t, b+1}, P_{t, b+2}, \dots, P_{t, k}, \dots, P_{t, c}\}$ 中各向量值均为空;设置t时刻的路径流率向量 $Rate_t = \{R_{t, b+1}, R_{t, b+2}, \dots, R_{t, k}, \dots, R_{t, c}\}$ 中各分量值均为空;设置t时刻的最大流率 $R_{t, max} = 0$ ;设置t时刻的最大流率路径 $P_{t, max}$ 为空;

[0104] 步骤3.1.2、带时间维度的路径遍历有向路网结构 $G(V, E)$ '复位:

[0105] 将有向路网结构 $G(V, E)$ 赋值给 $G(V, E)'$ ;

[0106] 根据剩余容量时间扩展节点集 $V_{cap}$ 中剩余通行容量为0的节点获得所对应的的时间维度,并判定在 $G(V, E)'$ 中剩余通行容量为0的节点在所对应的的时间维度上为无效;

[0107] 根据剩余容量时间扩展边集 $E_{cap}$ 中剩余通行容量为0的边获得所对应的的时间维度,并判定在 $G(V, E)'$ 中剩余通行容量为0的边在所对应的的时间维度上为无效;从而获得带时间维度的路径遍历有向路网结构 $G(V, E)'$ ;

[0108] 例如,带时间维度的路径遍历路网结构 $G(V, E)'$ 中,假设从节点 $V_5$ 出发到节点 $V_8$ 结束的边在时刻 $t=5$ 的剩余容量为0,则将路径遍历路网结构 $G(V, E)'$ 中该条边在 $t=5$ 时刻设为无效边;

[0109] 步骤3.2、根据带时间维度的路径遍历有向路网结构 $G(V, E)'$ 中节点和边是否为无效,遍历带时间维度的路径遍历有向路网结构 $G(V, E)'$ 中从超级源节点 $V_0$ 到目的节点集 $V_d$ 中第 $k$ 个目的节点 $V_k$ 所经过的所有路径;

[0110] 选取到达第 $k$ 个目的节点 $V_k$ 所经过的所有路径中花费时间代价最小的路径作为第 $k$ 个目的节点 $V_k$ 的最短路径并分别存入时刻 $t$ 路径选择队列向量 $Path_t$ 中的第 $k$ 个分量 $P_{t,k}$ 中;从而将目的节点集 $V_d$ 中所有目的节点的最短路径分别存入路径选择队列向量 $Path_t$ 相应的分量中;

[0111] 传统的最短路径算法主要有Dijkstra算法, $A^*$ 算法等,加入时间维度均可获得最短路径;在本发明中,以Dijkstra算法为例,采用一种改进的Dijkstra算法:利用最小堆的方法来优化Dijkstra中对目标节点的选择;假设存在一个已选节点集,在该已选节点集中,利用最小堆方法,从带时间维度的路径遍历有向路网结构 $G(V, E)'$ 中选择到超级源节点 $V_0$ 花费时间代价最小的节点作为过渡节点,再根据剩余容量时间扩展节点集 $V_{cap}$ 和剩余容量时间扩展边集 $E_{cap}$ 的查询结果,判定以该目标节点为起点的所有边和相对应的终点的有效性,如果无效,则将该边设为无效;若有效,则将相应的终点的信息加入已选节点集里,进行下一个节点的寻找操作,直到遍历完所有节点,则可以得到从源节点到目的节点的最短路径;如图2所示,假设在时刻 $t=0$ 首次执行步骤3,从超级源节点 $V_0$ 到目的节点 $V_{13}$ 的路径为例,获得其最短路径 $P_{0,13} = \{V_0, V_1, V_5, V_8, V_{11}, V_{13}\}$ ;

[0112] 步骤3.3、将 $t$ 时刻路径选择队列向量 $Path_t$ 中第 $k$ 个分量 $P_{t,k}$ 所花费的时间代价存入路径代价向量 $D_t$ 中的第 $k$ 个分量 $D_{t,k}$ 中,从而将目的节点集 $V_d$ 中所有目的节点的最短路径所花费时间代价分别存入路径代价向量 $D_t$ 相应的分量中;例如, $t=0$ 时刻从超级源节点到目的节点 $V_{13}$ 的最短路径 $P_{0,13} = \{V_0, V_1, V_5, V_8, V_{11}, V_{13}\}$ 花费的时间代价为路径边的通行时间之和,将其存入路径代价向量 $D_{0,13}$ 中, $D_{0,13} = 0+1+1+3+1=6$ ;

[0113] 步骤3.4、获得 $t$ 时刻路径选择队列向量 $Path_t$ 中第 $k$ 个分量 $P_{t,k}$ 所表示的最短路径上允许通行的最大疏散容量 $F_{P_{t,k}}$ ;从而获得在 $t$ 时刻从超级源节点 $V_0$ 到目的节点集 $V_d$ 中所有目的节点的最短路径上的最大疏散容量集 $\{F_{P_{t,b+1}}, F_{P_{t,b+2}}, \dots, F_{P_{t,k}}, \dots, F_{P_{t,c}}\}$ ;

[0114] 步骤3.4.1、获得 $t$ 时刻路径选择队列向量 $Path_t$ 中第 $k$ 个分量 $P_{t,k}$ 所表示的最短路径,记为 $P_{t,k} = \{V_0^{P_{t,k}}, V_1^{P_{t,k}}, \dots, V_{n_{t,k}-1}^{P_{t,k}}, V_{n_{t,k}}^{P_{t,k}}, \dots, V_{N_{t,k}}^{P_{t,k}}\}$ ,并有 $V_0^{P_{t,k}} = V_0$ ,  $V_{n_{t,k}}^{P_{t,k}} \in V$ ;  $N_{t,k}$ 表示最短路径 $P_{t,k}$ 中除超级源节点 $V_0$ 外所经过的节点总数; $V_{n_{t,k}}^{P_{t,k}}$ 表示最短路径 $P_{t,k}$ 中所经过的第 $n_{t,k}$ 个节

点,  $1 \leq n_{t,k} \leq N_{t,k}$ ; 例如,  $t=0$  时刻从超级源节点  $V_0$  到目的节点  $V_{13}$  最短路径  $P_{0,13} = \{V_0, V_1, V_5, V_8, V_{11}, V_{13}\}$ , 除超级源节点  $V_0$  外经过节点为 5 个, 则  $N_{0,13} = 5$ ;

[0115] 步骤 3.4.2、利用式 (1) 获得最短路径  $P_{t,k}$  中所经过第 0 个节点  $V_0^{P_{t,k}}$  以及经过的第  $n_{t,k}$  个节点  $V_{n_{t,k}}^{P_{t,k}}$  的到达时间  $T_{n_{t,k}}^{P_{t,k}}$ ; 从而获得最短路径  $P_{t,k}$  中所经过的节点的到达时间集  $\{T_0^{P_{t,k}}, T_1^{P_{t,k}}, \dots, T_{n_{t,k}}^{P_{t,k}}, \dots, T_{N_{t,k}}^{P_{t,k}}\}$ ;

$$[0116] \quad \begin{cases} T_0^{P_{t,k}} = t, \\ T_{n_{t,k}}^{P_{t,k}} = T_{n_{t,k}-1}^{P_{t,k}} + e_{\langle V_{n_{t,k}-1}^{P_{t,k}}, V_{n_{t,k}}^{P_{t,k}} \rangle}, \quad \forall n_{t,k}, 1 \leq n_{t,k} \leq N_{t,k} \end{cases} \quad (1)$$

[0117] 式 (1) 中,  $e_{\langle V_{n_{t,k}-1}^{P_{t,k}}, V_{n_{t,k}}^{P_{t,k}} \rangle}$  表示在边全集  $E_{\text{union}}$  中, 即边集  $E$  和超级源节点  $V_0$  分别与源节点集  $V_s$  中的每个源节点相连的边集  $E_{V_0}$  中, 由最短路径  $P_{t,k}$  中从第  $n_{t,k}-1$  个节点  $V_{n_{t,k}-1}^{P_{t,k}}$  出发到与其相连的第  $n_{t,k}$  个节点  $V_{n_{t,k}}^{P_{t,k}}$  结束所生成的边的通行时间; 例如, 时刻  $t=0$  从超级源节点  $V_0$  到目的节点  $V_{13}$  的最短路径  $P_{0,13} = \{V_0, V_1, V_5, V_8, V_{11}, V_{13}\}$ , 最短路径  $P_{0,13}$  中所有节点的到达时间集为  $\{0, 0, 1, 2, 5, 6\}$ ;

[0118] 步骤 3.4.3、更新路网状态参数:

[0119] 步骤 3.4.3.1、根据最短路径  $P_{t,k}$  中所经过第  $n_{t,k}$  个节点  $V_{n_{t,k}}^{P_{t,k}}$  的到达时间  $T_{n_{t,k}}^{P_{t,k}}$ , 查询剩余容量时间扩展节点集  $V_{\text{cap}}$  中是否包含到达时间  $T_{n_{t,k}}^{P_{t,k}}$  的通行容量节点集  $V_{\text{cap}}^{(T_{n_{t,k}}^{P_{t,k}})}$ , 若不包含, 则令  $V_{\text{cap}}^{(T_{n_{t,k}}^{P_{t,k}})}$  为空并存入剩余容量时间扩展节点集  $V_{\text{cap}}$  中; 否则, 维持  $V_{\text{cap}}^{(T_{n_{t,k}}^{P_{t,k}})}$  不变;

[0120] 查询通行容量节点集  $V_{\text{cap}}^{(T_{n_{t,k}}^{P_{t,k}})}$  中是否包含第  $n_{t,k}$  个节点  $V_{n_{t,k}}^{P_{t,k}}$  在到达时间  $T_{n_{t,k}}^{P_{t,k}}$  时的剩余通行节点容量  $V_{\text{cap}}^{(V_{n_{t,k}}^{P_{t,k}}, T_{n_{t,k}}^{P_{t,k}})}$ , 若包含, 则维持  $V_{\text{cap}}^{(V_{n_{t,k}}^{P_{t,k}}, T_{n_{t,k}}^{P_{t,k}})}$  不变; 否则, 将最短路径  $P_{t,k}$  中所经过第  $n_{t,k}$  个节点  $V_{n_{t,k}}^{P_{t,k}}$  在节点集  $V$  中所对应节点的通行容量  $V_{n_{t,k}, \text{cap}}$  赋值给  $V_{\text{cap}}^{(V_{n_{t,k}}^{P_{t,k}}, T_{n_{t,k}}^{P_{t,k}})}$ , 将  $V_{\text{cap}}^{(V_{n_{t,k}}^{P_{t,k}}, T_{n_{t,k}}^{P_{t,k}})}$  存入通行容量节点集  $V_{\text{cap}}^{(T_{n_{t,k}}^{P_{t,k}})}$  中; 从而实现将剩余通行节点容量  $V_{\text{cap}}^{(V_{n_{t,k}}^{P_{t,k}}, T_{n_{t,k}}^{P_{t,k}})}$  存入剩余容量时间扩展节点集  $V_{\text{cap}}$  中;

[0121] 步骤 3.4.3.2、重复执行步骤 3.4.3.1, 从而获得最短路径  $P_{t,k}$  中所经过的节点在各自到达时间维度上的剩余节点容量, 并依次存入剩余容量时间扩展节点集  $V_{\text{cap}}$  中;

[0122] 例如, 假设在时刻  $t=0$  首次执行步骤 3, 当前路网结构  $G(V, E)$  所有节点和边都有效, 已知由步骤 3.2 和步骤 3.4.2 分别获得时刻  $t=0$  从超级源节点  $V_0$  到目的节点  $V_{13}$  的最短路径  $P_{0,13} = \{V_0, V_1, V_5, V_8, V_{11}, V_{13}\}$  以及  $P_{0,13}$  中所有节点的到达时间集  $\{0, 0, 1, 2, 5, 6\}$ , 则源节点  $V_1$  在到达时间  $t=0$  的剩余容量由步骤 2.2 可知  $V_{\text{cap}}^{(V_1, 0)} = V_{1, \text{cap}} = 50$ , 已经被剩余容量时间扩展节点集  $V_{\text{cap}}$  中  $t=0$  时刻的通行容量节点集  $V_{\text{cap}}^{(0)}$  所包含, 则执行最短路径  $P_{0,13} = \{V_0, V_1, V_5,$

$V_8, V_{11}, V_{13}$  中下一个经过的节点  $V_5$  的操作; 然而例如  $V_5$  在到达时间  $t=1$  时, 剩余容量时间扩展节点集  $V_{cap}$  中不包含通行容量节点集  $V_{cap}^{(1)}$ , 则令  $V_{cap}^{(1)}$  为空并存入  $V_{cap}$  中; 若  $V_{cap}^{(1)}$  中没有包含时刻  $t=1$  节点  $V_5$  的剩余通行节点容量  $V_{cap}^{(V_5,1)}$ , 则令  $V_{cap}^{(V_5,1)} = V_{5,cap} = 6$ , 并将  $V_{cap}^{(V_5,1)} = 6$  存入  $V_{cap}^{(1)}$  中, 实现了将  $V_{cap}^{(V_5,1)} = 6$  存入  $V_{cap}$  中; 依次类似, 获得中间节点  $V_8$  在到达时间  $t=2$  的剩余容量  $V_{cap}^{(V_8,2)} = V_{8,cap} = 30$ , 中间节点  $V_{11}$  在到达时间  $t=5$  的剩余容量  $V_{cap}^{(V_{11},5)} = V_{11,cap} = 18$ , 目的节点  $V_{13}$  在到达时间  $t=6$  的剩余容量  $V_{cap}^{(V_{13},6)} = V_{13,cap} = 200$ ; 依次将这些节点在其到达时间维度上的剩余节点容量存入相应时间维度的通行容量节点集中, 从而实现将这些节点的通行节点容量存入  $V_{cap}$  中;

[0123] 步骤3.4.3.3、根据最短路径  $P_{t,k}$  中所经过从第  $n_{t,k-1}$  个节点  $V_{n_{t,k-1}}^{P_{t,k}}$  出发到与其相连的第  $n_{t,k}$  个节点  $V_{n_{t,k}}^{P_{t,k}}$  结束所生成的边的到达时间  $T_{n_{t,k-1}}^{P_{t,k}}$ , 查询剩余容量时间扩展边集  $E_{cap}$  中是否包含到达时间  $T_{n_{t,k-1}}^{P_{t,k}}$  时的通行容量边集  $E_{cap}^{(T_{n_{t,k-1}}^{P_{t,k}})}$ , 若不包含, 令  $E_{cap}^{(T_{n_{t,k-1}}^{P_{t,k}})}$  为空并存入剩余容量时间扩展边集  $E_{cap}$  中; 否则, 维持  $E_{cap}^{(T_{n_{t,k-1}}^{P_{t,k}})}$  不变;

[0124] 查询通行容量边集  $E_{cap}^{(T_{n_{t,k-1}}^{P_{t,k}})}$  中是否包含最短路径  $P_{t,k}$  中从第  $n_{t,k-1}$  个节点  $V_{n_{t,k-1}}^{P_{t,k}}$  出发到与其相连的第  $n_{t,k}$  个节点  $V_{n_{t,k}}^{P_{t,k}}$  结束所生成的边在到达时间  $T_{n_{t,k-1}}^{P_{t,k}}$  时的剩余通行边容量  $e_{cap}^{(V_{n_{t,k-1}}^{P_{t,k}}, V_{n_{t,k}}^{P_{t,k}})_{T_{n_{t,k-1}}^{P_{t,k}}}}$ , 若包含, 则维持  $e_{cap}^{(V_{n_{t,k-1}}^{P_{t,k}}, V_{n_{t,k}}^{P_{t,k}})_{T_{n_{t,k-1}}^{P_{t,k}}}}$  不变; 否则, 将边全集  $E_{union}$  中由最短路径  $P_{t,k}$  中从第  $n_{t,k-1}$  个节点  $V_{n_{t,k-1}}^{P_{t,k}}$  出发到与其相连的第  $n_{t,k}$  个节点  $V_{n_{t,k}}^{P_{t,k}}$  结束所生成的边的通行容量  $e_{cap}^{(V_{n_{t,k-1}}^{P_{t,k}}, V_{n_{t,k}}^{P_{t,k}})}$  赋值给  $e_{cap}^{(V_{n_{t,k-1}}^{P_{t,k}}, V_{n_{t,k}}^{P_{t,k}})_{T_{n_{t,k-1}}^{P_{t,k}}}}$ , 将  $e_{cap}^{(V_{n_{t,k-1}}^{P_{t,k}}, V_{n_{t,k}}^{P_{t,k}})_{T_{n_{t,k-1}}^{P_{t,k}}}}$  存入通行容量边集  $E_{cap}^{(T_{n_{t,k-1}}^{P_{t,k}})}$  中, 从而实现将剩余通行边容量  $e_{cap}^{(V_{n_{t,k-1}}^{P_{t,k}}, V_{n_{t,k}}^{P_{t,k}})_{T_{n_{t,k-1}}^{P_{t,k}}}}$  存入剩余容量时间扩展边集  $E_{cap}$  中;

[0125] 步骤3.4.3.4、重复执行步骤3.4.3.3, 从而获得最短路径  $P_{t,k}$  中所经过的边在各自到达时间维度上的剩余边容量, 并依次存入剩余容量时间扩展边集  $E_{cap}$  中;

[0126] 例如, 上述示例中最短路径  $P_{0,13} = \{V_0, V_1, V_5, V_8, V_{11}, V_{13}\}$  各条边的剩余容量计算如下: 例如由超级源节点  $V_0$  出发到源节点  $V_1$  的结束的边在到达时间  $t=0$  的剩余容量  $e_{cap}^{(V_0, V_1)_0}$  由步骤2.2可知  $e_{cap}^{(V_0, V_1)_0} = 10$ , 已经被剩余容量时间扩展边集  $E_{cap}$  中  $t=0$  时刻的通行容量边集  $E_{cap}^{(0)}$  中所包含, 则执行最短路径  $P_{0,13}$  中下一条经过的边的操作; 由源节点  $V_1$  出发到扩展中间节点  $V_5$  结束所生成的边在到达时间  $t=0$  的剩余容量  $e_{cap}^{(V_1, V_5)_0}$  与  $e_{cap}^{(V_0, V_1)_0}$  类似, 已经被剩余容量时间扩展边集  $E_{cap}$  中  $t=0$  时刻的通行容量边集  $E_{cap}^{(0)}$  所包含, 则继续执行最短路径  $P_{0,13}$  中

下一条经过的边的操作；

[0127] 然而例如由扩展中间节点 $V_5$ 出发到扩展中间节点 $V_8$ 的结束所生成的边在到达时间 $t=1$ 时,在剩余容量时间扩展边集为 $E_{cap}$ 中不包含通行容量边集 $E_{cap}^{(1)}$ ,则令 $E_{cap}^{(1)}$ 为空,并将 $E_{cap}^{(1)}$ 存入 $E_{cap}$ 中;若通行容量边集 $E_{cap}^{(1)}$ 中不包含剩余通行边容量 $e_{cap}^{(\{V_5, V_8\}, 1)}$ ,则令 $e_{cap}^{(\{V_5, V_8\}, 1)} = 4$ ,并存入 $E_{cap}^{(1)}$ 中,从而实现将 $e_{cap}^{(\{V_5, V_8\}, 1)} = 4$ 存入 $E_{cap}$ 中;依次类似,由扩展中间节点 $V_8$ 出发到扩展中间节点 $V_{11}$ 的结束所生成的边在到达时间 $t=2$ 的剩余容量 $e_{cap}^{(\{V_8, V_{11}\}, 2)} = 7$ ,由扩展中间节点 $V_{11}$ 出发到目的节点 $V_{13}$ 的结束所生成的边在到达时间 $t=5$ 的剩余容量 $e_{cap}^{(\{V_{11}, V_{13}\}, 5)} = 6$ ,依次将这些边在其到达时间维度上的剩余边容量存入相应时间维度的通行容量边集中,从而实现将这些边的通行边容量存入 $E_{cap}$ 中;

[0128] 步骤3.4.4、根据式(2)获得路径选择队列向量 $Path_t$ 中的第 $k$ 个分量 $P_{t,k}$ 所表示的最短路径上允许通行的最大疏散容量 $F_{P_{t,k}}$ :

$$[0129] \quad F_{P_{t,k}} = \min \begin{cases} V_{0,cap} \\ V_{cap}^{(V_{n_{t,k}}^{P_{t,k}}, T_{n_{t,k}}^{P_{t,k}})} \\ e_{cap}^{(\{V_{n_{t,k-1}}^{P_{t,k}}, V_{n_{t,k}}^{P_{t,k}}\}, T_{n_{t,k-1}}^{P_{t,k}})} \end{cases} \quad (2)$$

s.t.  $\forall 1 \leq n_{t,k} \leq N_{t,k}$

[0130] 式(1)中,  $V_{cap}^{(V_{n_{t,k}}^{P_{t,k}}, T_{n_{t,k}}^{P_{t,k}})}$ 表示最短路径 $P_{t,k}$ 上的第 $n_{t,k}$ 个节点 $V_{n_{t,k}}^{P_{t,k}}$ 在到达时间 $T_{n_{t,k}}^{P_{t,k}}$ 时的剩余通行容量,由剩余容量时间扩展节点集 $V_{cap}$ 中获得;  $e_{cap}^{(\{V_{n_{t,k-1}}^{P_{t,k}}, V_{n_{t,k}}^{P_{t,k}}\}, T_{n_{t,k-1}}^{P_{t,k}})}$ 表示从最短路径 $P_{t,k}$ 上的第 $n_{t,k-1}$ 个节点 $V_{n_{t,k-1}}^{P_{t,k}}$ 出发到与其相连的第 $n_{t,k}$ 个节点 $V_{n_{t,k}}^{P_{t,k}}$ 结束所生成的边在到达时间 $T_{n_{t,k-1}}^{P_{t,k}}$ 时的剩余通行容量,由剩余容量时间扩展边集 $E_{cap}$ 中获得;

[0131] 例如,步骤3.4.3中所举示例,假设在时刻 $t=0$ 首次执行步骤4,当前有向路网结构 $G(V, E)'$ 中的所有节点和边都有效,获得 $V_{0,cap} = 30$ ,最短路径 $P_{0,13} = \{V_0, V_1, V_5, V_8, V_{11}, V_{13}\}$ 中节点 $\{V_1, V_5, V_8, V_{11}, V_{13}\}$ 在各自所对应的到达时间维度 $\{0, 1, 2, 5, 6\}$ 上的剩余通行容量为 $\{50, 6, 30, 18, 200\}$ ,各边在各自所对应的到达时间维度 $\{0, 0, 1, 2, 5\}$ 上的剩余通行容量为 $\{10, 5, 4, 7, 6\}$ ,则利用式(2)获得这条最短路径上面允许通行的最大疏散容量为 $F_{P_{0,13}} = \min(30, \{50, 6, 30, 18, 200\}, \{10, 5, 4, 7, 6\}) = 4$ ;

[0132] 步骤3.5、利用式(3)获得 $t$ 时刻从超级源节点 $V_0$ 到目的节点集 $V_d$ 中第 $k$ 个目的节点 $V_k$ 的最短路径的流率 $R_{t,k}$ ,例如,最短路径 $P_{0,13}$ 的流率 $R_{0,13} = F_{P_{0,13}}/D_{0,13} = 4/6 = 0.677$ ;从而获得 $t$ 时刻从超级源节点 $V_0$ 到目的节点集 $V_d$ 中所有目的节点的最短路径的流率向量 $Rate_t = \{R_{t,b+1}, R_{t,b+2}, \dots, R_{t,k}, \dots, R_{t,c}\}$ ;并获得 $t$ 时刻的路径流率向量 $Rate_t$ 中最大流率 $R_{t,max}$ :

$$[0133] \quad R_{t,k} = F_{P_{t,k}}/D_{t,k} \quad (3)$$

[0134] 步骤3.6、判断 $R_{t,max} = 0$ 和 $V_{0,cap} = 0$ 是否成立,若都成立,则执行步骤3.10;

[0135] 若只有 $R_{t,\max}=0$ 成立,则表示 $t$ 时刻路径选择队列向量 $Path_t$ 无效,且在 $t$ 时刻不能再次进行疏散,将 $t+1$ 的值赋给 $t$ ,并返回步骤3.1重新计算疏散规划路径;

[0136] 若都不成立,则表示在 $t$ 时刻路径选择队列向量 $Path_t$ 有效,能够选出一条可执行的疏散路径,并从路径选择队列向量 $Path_t$ 中将最大流率 $R_{t,\max}$ 所对应的最短路径选出,存入最大流率路径 $P_{t,\max}$ ,并记为 $P_{t,\max} = \{V_0^{P_{t,\max}}, V_1^{P_{t,\max}}, \dots, V_{n'_{t,\max}-1}^{P_{t,\max}}, V_{n'_{t,\max}}^{P_{t,\max}}, \dots, V_{N'_{t,\max}}^{P_{t,\max}}\}$ ,以及最大流率路径 $P_{t,\max}$ 中所经过节点的到达时间集 $\{T_0^{P_{t,\max}}, T_1^{P_{t,\max}}, \dots, T_{n'_{t,\max}-1}^{P_{t,\max}}, T_{n'_{t,\max}}^{P_{t,\max}}, \dots, T_{N'_{t,\max}}^{P_{t,\max}}\}$ ,并有 $V_0^{P_{t,\max}} = V_0, V_{n'_{t,\max}}^{P_{t,\max}} \in V$ ;  $N_{t,\max}'$ 表示最大流率路径 $P_{t,\max}$ 中除超级源节点 $V_0$ 外所经过的节点总数,  $V_{n'_{t,\max}}^{P_{t,\max}}$ 表示最大流率路径 $P_{t,\max}$ 中所经过的第 $n_{t,\max}'$ 个节点,  $1 \leq n_{t,\max}' \leq N_{t,\max}'$ ;记最大流率路径 $P_{t,\max}$ 所对应的最大疏散容量为 $F_{t,\max}$ ;

[0137] 如图3所示,依据此步骤获得最大流率及相应的最短路径;将最大流率路径 $P_{t,\max}$ 中除超级源节点 $V_0$ 外路径所经过的节点存入路径输出队列集合 $OutPut\_Path$ 中,并记录最大疏散容量和各节点的到达时间,例如,如图2所示的有向路网结构中,假设此步骤首次获得了具有最大流率的最短路径 $P_{t,\max}$ 为在 $t=0$ 时刻从超级源节点 $V_0$ 到目的节点 $V_{13}$ 的最短路径 $P_{0,13} = \{V_0, V_1, V_5, V_8, V_{11}, V_{13}\}$ 及路径中所有节点的相应到达时间集 $\{0, 0, 1, 2, 5, 6\}$ ,允许通行的最大疏散容量为 $F_{P_{0,13}} = 4$ ,如表1中第一行输出所示,除超级源节点不写入,其它节点采用“-节点(到达时间)”的格式写入,路径规划格式为 $V_1(0) - V_5(1) - V_8(2) - V_{11}(5) - V_{13}(6)$ ;该最短路径的源节点为 $V_1$ ,到达的目的节点为 $V_{13}$ ,路径上的最大疏散容量为4,执行该疏散路径所花费的时间代价为6;并执行步骤3.7;

[0138] 步骤3.7、利用式(4)分别对最大流率路径 $P_{t,\max}$ 中所经过的节点和边的剩余通行容量进行修改,将修改值重新存入剩余容量时间扩展节点集 $V_{cap}$ 和剩余容量时间扩展边集 $E_{cap}$ 中,从而实现剩余容量时间扩展节点集 $V_{cap}$ 和剩余容量时间扩展边集 $E_{cap}$ 中各分量值的更新;

$$[0139] \quad \begin{cases} e_{cap}^{(V_{n'_{t,\max}-1}^{P_{t,\max}}, V_{n'_{t,\max}}^{P_{t,\max}}), T_{n'_{t,\max}-1}^{P_{t,\max}}} = e_{cap}^{(V_{n'_{t,\max}-1}^{P_{t,\max}}, V_{n'_{t,\max}}^{P_{t,\max}}), T_{n'_{t,\max}-1}^{P_{t,\max}}} - F_{t,\max} \\ V_{cap}^{(V_{n'_{t,\max}}^{P_{t,\max}}, T_{n'_{t,\max}}^{P_{t,\max}})} = V_{cap}^{(V_{n'_{t,\max}}^{P_{t,\max}}, T_{n'_{t,\max}}^{P_{t,\max}})} - F_{t,\max} \end{cases} \quad (4)$$

s. t.  $n'_{t,\max}, \forall 1 \leq n'_{t,\max} \leq N'_{t,\max}$

[0140] 式(4)中,  $e_{cap}^{(V_{n'_{t,\max}-1}^{P_{t,\max}}, V_{n'_{t,\max}}^{P_{t,\max}}), T_{n'_{t,\max}-1}^{P_{t,\max}}}$ 表示最大流率路径 $P_{t,\max}$ 上的第 $n_{t,\max}'-1$ 个节点 $V_{n'_{t,\max}-1}^{P_{t,\max}}$ 发到与其相连的第 $n_{t,\max}'$ 个节点 $V_{n'_{t,\max}}^{P_{t,\max}}$ 结束所生成的边在到达时间 $T_{n'_{t,\max}-1}^{P_{t,\max}}$ 时的剩余通行容量;  $V_{cap}^{(V_{n'_{t,\max}}^{P_{t,\max}}, T_{n'_{t,\max}}^{P_{t,\max}})}$ 表示最大流率路径 $P_{t,\max}$ 上的第 $n_{t,\max}'$ 个节点 $V_{n'_{t,\max}}^{P_{t,\max}}$ 在到达时间 $T_{n'_{t,\max}}^{P_{t,\max}}$ 时的剩余通行容量;

[0141] 例如,步骤3.4.3中所举示例,首次得到的具有最大流率路径 $P_{0,13} = \{V_0, V_1, V_5, V_8, V_{11}, V_{13}\}$ 中,扩展中间节点 $V_5$ 在到达时间 $t=1$ 时的初始剩余容量 $V_{cap}^{(V_5,1)} = 6$ ,利用式(4)计算

后得到  $V_{cap}^{(V_3,1)} = V_{cap}^{(V_3,1)} - F_{t,max} = 6 - 4 = 2$ ; 扩展中间节点  $V_8$  在到达时间  $t = 2$  时的剩余容量  $V_{cap}^{(V_8,2)} = 30$ , 利用式 (4) 计算后得到  $V_{cap}^{(V_8,2)} = V_{cap}^{(V_8,2)} - F_{t,max} = 30 - 4 = 26$ ; 其它节点在各自所对应的到达时间维度上的剩余容量依次计算获得; 最大流率路径中所经过各边在各自所对应的到达时间上的剩余容量计算操作皆与此类似, 由扩展中间节点  $V_5$  出发到扩展中间节点  $V_8$  结束所生成的边在到达时间  $t = 1$  时的剩余容量为  $e_{cap}^{((V_5, V_8), 1)} = 4$ , 利用式 (4) 计算后得到  $e_{cap}^{((V_5, V_8), 1)} = e_{cap}^{((V_5, V_8), 1)} - F_{t,max} = 4 - 4 = 0$ , 剩余容量为 0, 意味着如步骤 4.1 所示, 该边将在  $t = 1$  时刻无效; 其余边在所对应的到达时间的的时间维度上的剩余容量可依次计算获得;

[0142] 因此, 依次完成剩余容量时间扩展节点集  $V_{cap}$  和剩余容量边集  $E_{cap}$  的中各分量的值的更新;

[0143] 步骤 3.8、利用式 (5) 获得超级源节点  $V_0$  在时刻  $t$  的剩余通行容量  $V_{0, cap}$ , 计算所得为当前还需要疏散的有效容量:

$$[0144] \quad V_{0, cap} - V_{0, cap} = F_{t, max} \quad (5)$$

[0145] 步骤 3.9、判断  $V_{0, cap} = 0$  是否成立, 若成立, 则执行步骤 3.10; 否则, 执行步骤 3.1, 此处转向步骤 3.1 重新执行新一轮的计算, 并没有将  $t+1$  的值赋给  $t$ , 意味着当前时刻  $t$  下还可能存在一条疏散路径, 即将执行的是当前时刻  $t$  下新一轮的最大流率路径的迭代计算;

[0146] 步骤 3.10、疏散规划结束, 获得存储了最大流率路径的路径输出队列集合  $OutPut\_Path$  的最终结果; 例如, 如图 1 所示的路网结构, 其结果如表 1 路径输出队列集合  $OutPut\_Path$  示例所示, 输出了保证总疏散人数全部疏散完毕的多条疏散规划路径, 路径上有允许疏散的人数以及疏散时间。

[0147] 表 1 路径输出队列集合  $OutPut\_Path$  示例

[0148]

源节点	疏散人数	路径规划	疏散时间
$V_1$	4	$V_1(0) - V_5(1) - V_8(2) - V_{11}(5) - V_{13}(6)$	6
$V_1$	1	$V_1(0) - V_5(1) - V_{11}(6) - V_{13}(7)$	7
$V_3$	4	$V_3(0) - V_4(1) - V_5(3) - V_8(4) - V_{11}(7) - V_{13}(8)$	8
$V_3$	1	$V_3(0) - V_6(3) - V_8(4) - V_{11}(7) - V_{13}(8)$	8
$V_2$	2	$V_2(0) - V_4(1) - V_5(3) - V_{11}(8) - V_{13}(9)$	9
$V_2$	5	$V_2(0) - V_4(1) - V_6(4) - V_8(5) - V_{11}(8) - V_{15}(10)$	10
$V_1$	4	$V_1(1) - V_5(2) - V_8(3) - V_{11}(6) - V_{13}(7)$	7
$V_1$	1	$V_1(1) - V_5(2) - V_{11}(7) - V_{13}(8)$	8
$V_2$	4	$V_2(1) - V_4(2) - V_5(4) - V_8(7) - V_9(11) - V_{12}(13) - V_{14}(13)$	13
$V_2$	2	$V_2(1) - V_4(2) - V_5(4) - V_{11}(9) - V_{13}(10)$	10
$V_2$	1	$V_2(1) - V_4(2) - V_6(5) - V_8(6) - V_{11}(9) - V_{13}(10)$	10
$V_2$	1	$V_2(2) - V_4(3) - V_5(5) - V_8(6) - V_{11}(9) - V_{13}(10)$	10

[0149] 以上, 仅为本发明较佳的一种实施方式, 给出了有向单向路网结构的示例, 另外, 本方法完全可以运用于有向的双向路网结构当中, 具体实施方式, 使用者可以根据现实世界的路网交通来进行数学建模及参数设置, 可以取得相同的效果。必须说明的是, 任何熟悉

本技术领域的技术人员在本发明揭露的技术范围内,根据本发明的技术方案及其发明构思加以等同替换或相关参数改变,都应涵盖在本发明的保护范围之内。



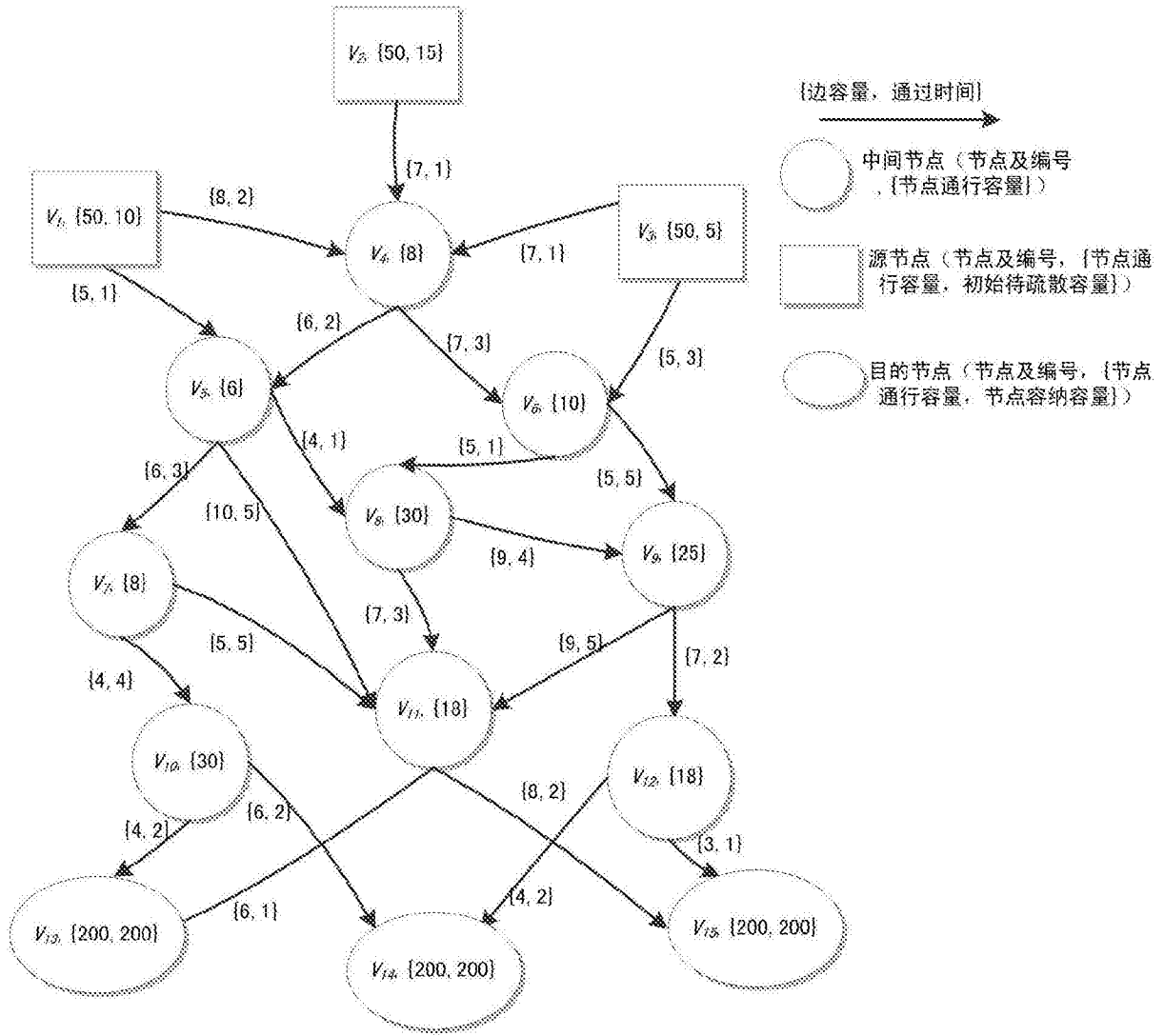


图1

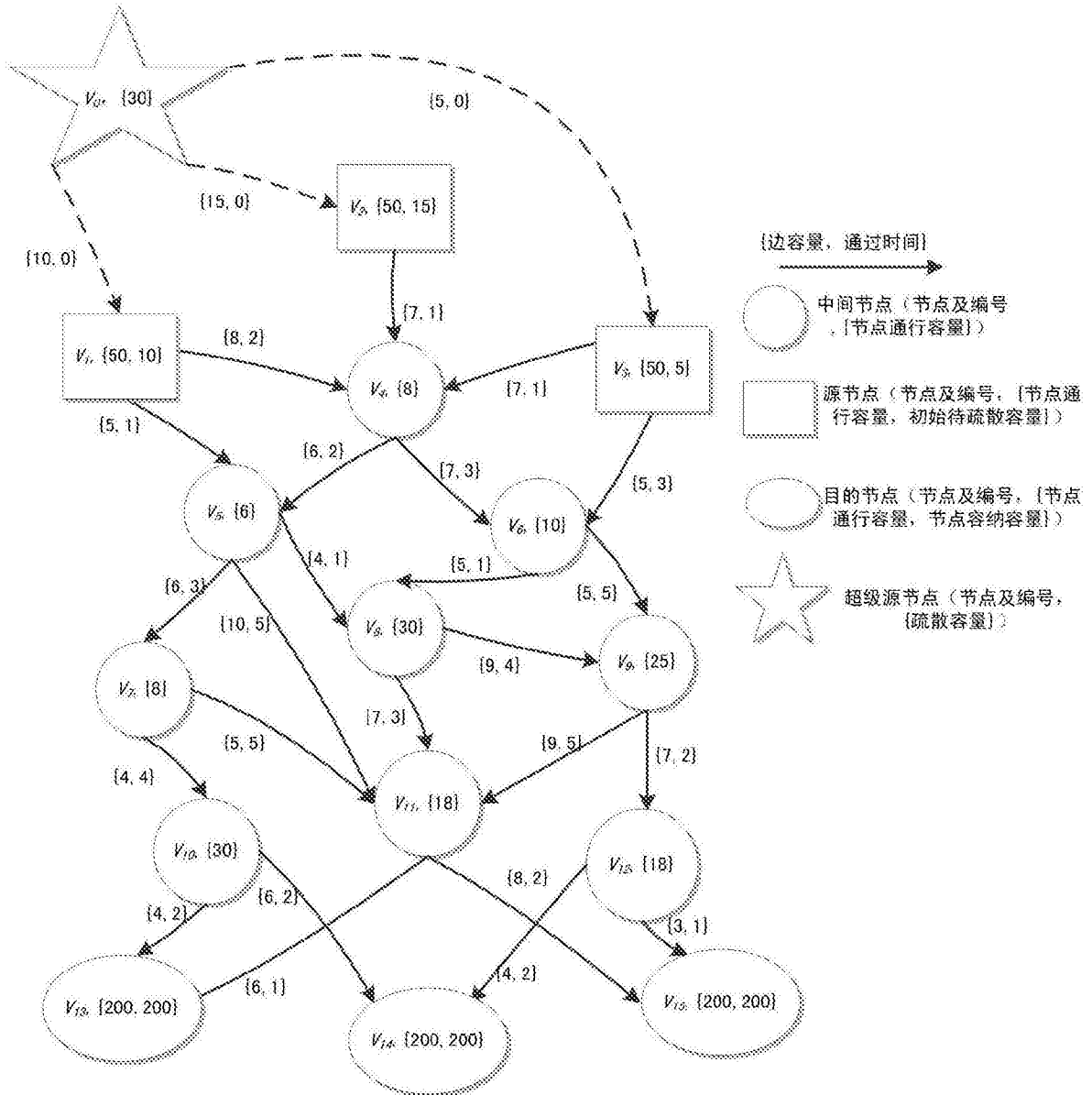


图2

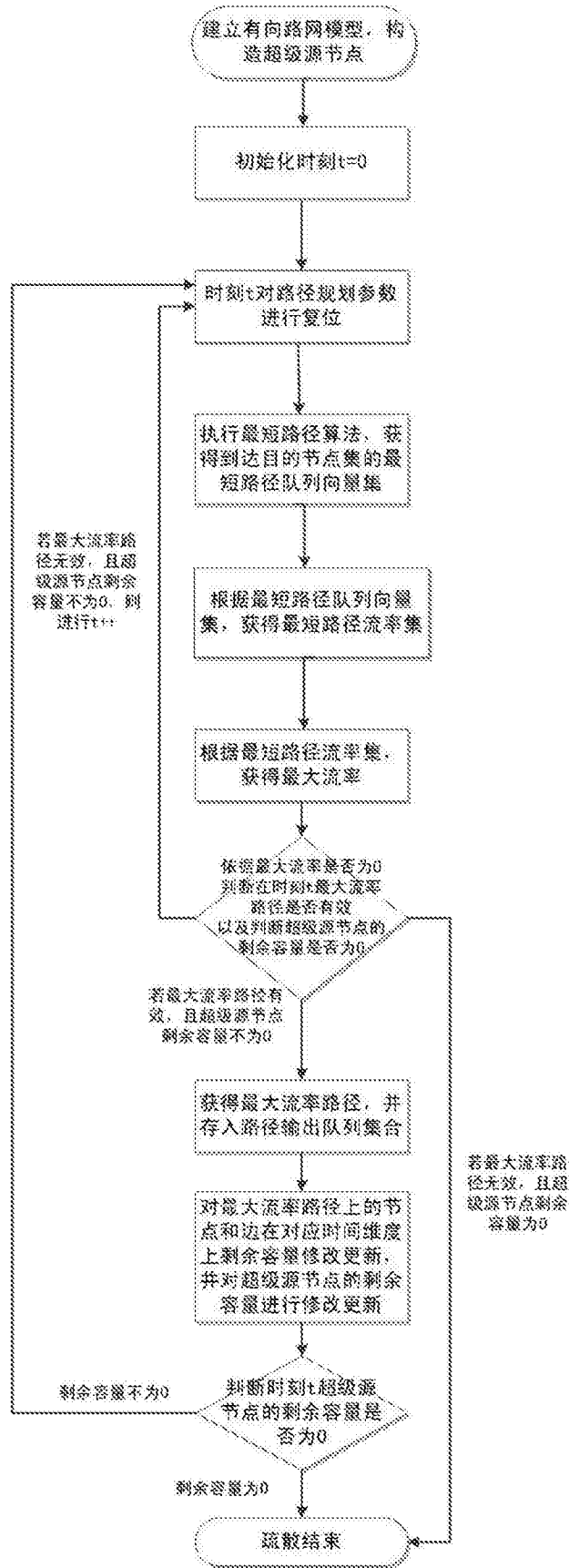


图3