

Evacuation Path Design Based on Floyd Algorithm

Caiyun Bai, Yaohua Yu*, Yang Li*, Yue Wang, Shuang Guo, Xinyu Sun, Changjie Qin

College of Science, Dalian Minzu University, Dalian Liaoning

Email: ^{*}cntimmoo@gmail.com, ^{*}liyang@dlnu.edu.cn

Received: Jul. 27th, 2020; accepted: Aug. 13th, 2020; published: Aug. 20th, 2020

Abstract

The design of evacuation path has an important application in building safety. The design of evacuation path is usually based on the shortest path theory. This paper gives the plan and network diagram of an office building, and uses Floyd algorithm to design the emergency evacuation path of the building. Then, it also considers how to choose the optimal evacuation path in the case that the exit accident of some safe passage cannot be used normally.

Keywords

Optimal Path, Evacuation Path, Floyd Algorithm

基于Floyd算法的疏散路径设计

白彩云, 于耀华*, 李 阳*, 王 越, 郭 爽, 孙欣宇, 覃昶潔

大连民族大学理学院, 辽宁 大连

Email: ^{*}cntimmoo@gmail.com, ^{*}liyang@dlnu.edu.cn

收稿日期: 2020年7月27日; 录用日期: 2020年8月13日; 发布日期: 2020年8月20日

摘 要

疏散路径的设计在建筑安全领域有着重要的应用。疏散路径的设计往往基于最短路径理论。本文给出了一座办公建筑的平面图和网络图, 并利用Floyd算法设计了该建筑紧急疏散路径, 之后还考虑了当个别安全通道出口突发事故而无法使用的情况下如何选择最优的疏散路径。

*通讯作者。

文章引用: 白彩云, 于耀华, 李阳, 王越, 郭爽, 孙欣宇, 覃昶潔. 基于 Floyd 算法的疏散路径设计[J]. 应用数学进展, 2020, 9(8): 1292-1297. DOI: 10.12677/aam.2020.98151

关键词

最优路径, 疏散路径, Floyd算法

Copyright © 2020 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

随着经济的发展和进步, 人们越来越重视生活中的安全性问题。安全性问题也是评判一座建筑好坏的标准之一。因此, 建筑空间疏散路径的设计成为近年来大家关注的热点问题。

通常情况下, 建筑空间的疏散路径设计是以最短路径理论为基础来进行研究的。最短路径理论的经典算法较多, 比如至今还广为使用的 Dijkstra 算法、Floyd 算法、SPFA 算法和 Bellman-Ford 算法等[1] [2] [3] [4], 但是我们在解决实际问题的時候往往不能仅考虑单一的距离长短问题, 其他现实因素有时也需要纳入考虑范围之内。

本文给出了一个建筑空间的平面图形, 并绘制了该建筑的示意图。然后利用经典的 Floyd 算法设计了七个人群聚集地的疏散路线。并且还考虑当某些紧急出口无法正常使用时, 这七个人群聚集地如何选择其他出口。

2. 问题介绍

如图 1 所示, 这是某建筑物的一个平层结构图。

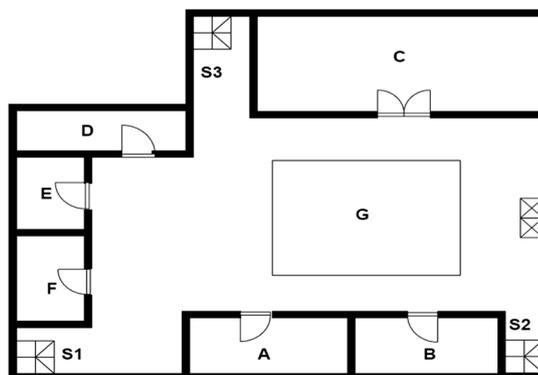


Figure 1. Architectural structural diagram

图 1. 建筑结构图

A, B, C, D, E, F, G 所在位置代表该建筑的办公室、会议室、开放式办公区等七个人群相对密集的地点。我们将这七个地点作为起点。而 S_1, S_2, S_3 是应急楼梯的位置, 作为终点。为了防止突发事件发生时人群慌不择路, 我们需要分别为这七个起点的人群设计最优的疏散路径。终点就是 S_1, S_2, S_3 其中之一。我们把七个起点和三个终点作为网络示意图的节点。

3. 节点间的距离测量

各节点间的距离计算方法如下: 若 M, N 是平面图中任意两节点, 则 M, N 两点的距离 $D(M, N)$ 是由

10 次测量值 $d_i, i=1, \dots, 10$ 按照下面公式算得的:

$$D(M, N) = \sum_{i=1}^{10} \rho_i d_i。$$

记 \bar{d} 为 10 次测量值的均值, 令 $a_i = |d_i - \bar{d}|, i=1, \dots, 10$ 。然后取 d_m 的权值

$$\rho_m = \frac{a_m}{\sum_{i=1}^{10} a_i}, m=1, \dots, 10。$$

用以上方法来确定节点之间的距离能使距离值更准确。为方便计算, 在本文中, 将节点 $A, B, S_2, G, C, S_3, D, E, F, S_1$ 按照顺序记为 x_1, x_2, \dots, x_{10} , 则任意两节点 x_i 与 x_j 之间的距离记为 D_{ij} 。

4. 利用 Floyd 算法设计疏散路径

首先, 介绍该算法步骤:

第 1 步 定义初始的距离矩阵 M_0 和节点序列矩阵 Q_0 ;

第 k 步 令第 k 行、第 k 列为枢轴行和枢轴列, 对于矩阵 M_{k-1} 中的每一个元素 D_{ij} 做以下步骤, 如果满足条件: $D_{ik} + D_{kj} < D_{ij}$ ($i \neq k, j \neq k, i \neq j$), 那么可以进行下面的转化:

用 $D_{ik} + D_{kj}$ 代替 M_{k-1} 中的元素 D_{ij} , 得到矩阵 M_k ;

用 k 代替矩阵 Q_{k-1} 中的元素 q_{ij} , 得到矩阵 Q_k 。令 $k = k + 1$, 如果 $k = n + 1$, 停止; 否则, 重复第 k 步。

经过 n 步后, 可以从矩阵 M_n 中按照下面规则得到节点 i 和节点 j 之间的最短路径:

其中,

在矩阵 M_n 中, D_{ij} 表示节点 i 和节点 j 之间的最短路径的长度;

在矩阵 Q_n 中可以确定中间节点 $k = q_{ij}$ (根据得到的路径 $i \rightarrow k \rightarrow j$)。如果 $q_{ik} = k$ 并且 $q_{kj} = j$, 停止, 因为路径中的所有中间节点都已找到。否则, 在节点 i 与节点 k 之间以及节点 k 与节点 j 之间重复上面的步骤。

5. 数值实验

(1) 各个节点之间的距离网络描述图如图 2 所示。

(2) 使用 Matlab 编程, 按照 $A, B, S_2, G, C, S_3, D, E, F, S_1$ 为顺序构造距离矩阵 $M_0 = (D_{ij})_{10 \times 10}$ 和节点矩阵 Q_0 。上文已经注明 D_{ij} 表示任意两节点之间的距离(单位: m), 且用 inf 表示没有边相连接或是人们在正常情况下不会选择的线路的两点间距离。

$$M_0 = \begin{pmatrix} 0 & 7.3 & 18.5 & 16.4 & \text{inf} & 42.8 & 35.6 & 24.1 & 18.9 & 24 \\ 7.3 & 0 & 12 & 15.3 & \text{inf} & 53 & 49.9 & 46.2 & 41 & 41.5 \\ 18.2 & 12 & 0 & 25.6 & 38.8 & \text{inf} & 67.4 & 56.2 & 50.1 & \text{inf} \\ 16.4 & 15.3 & 25.6 & 0 & 13.9 & 23.7 & 27.1 & 31.3 & 34.1 & 39.7 \\ \text{inf} & \text{inf} & 38.8 & 13.9 & 0 & 20 & 28.9 & 34.3 & 44.9 & 60.4 \\ 42.8 & 53 & \text{inf} & 13.9 & 20 & 0 & 21.5 & 32.6 & 45.6 & \text{inf} \\ 35.6 & 49.9 & 67.4 & 27.1 & 28.9 & 21.5 & 0 & 5.6 & 16 & 31 \\ 24.1 & 46.2 & 56.2 & 31.3 & 34.3 & 32.6 & 5.6 & 0 & 9 & 28 \\ 18.9 & 41 & 50.1 & 34.1 & 44.9 & 45.6 & 16 & 9 & 0 & 15 \\ 24 & 41.5 & \text{inf} & 39.7 & 60.4 & \text{inf} & 31 & 28 & 15 & 0 \end{pmatrix}, \text{ 而 } Q_0 = (0)_{10 \times 10}。$$

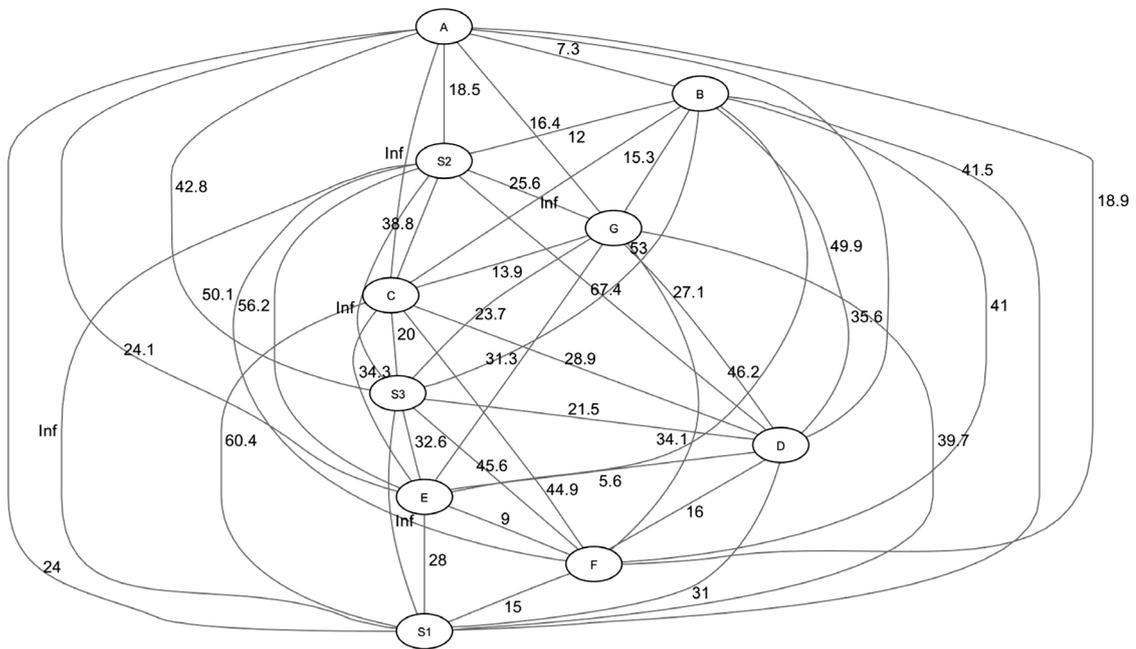


Figure 2. Network diagram
图 2. 网络图

利用 Floyd 算法来计算出最优路径，得到最终的矩阵 M_{10} ，

$$M_{10} = \begin{pmatrix} 0 & 7.3 & 18.5 & 16.4 & 30.3 & 40.1 & 29.7 & 24.1 & 18.9 & 24 \\ 7.3 & 0 & 12 & 15.3 & 29.2 & 39 & 37 & 31.4 & 26.2 & 31.3 \\ 18.2 & 12 & 0 & 25.6 & 38.8 & 49.3 & 47.9 & 42.3 & 37.1 & 42.2 \\ 16.4 & 15.3 & 25.6 & 0 & 13.9 & 23.7 & 27.1 & 31.3 & 34.1 & 39.7 \\ 30.3 & 29.2 & 38.8 & 13.9 & 0 & 20 & 28.9 & 34.3 & 43.3 & 53.6 \\ 30.3 & 29.2 & 39.5 & 13.9 & 20 & 0 & 21.5 & 27.1 & 36.1 & 51.1 \\ 29.7 & 37 & 48.2 & 27.1 & 28.9 & 21.5 & 0 & 5.6 & 14.6 & 29.6 \\ 24.1 & 31.4 & 42.6 & 31.3 & 34.3 & 27.1 & 5.6 & 0 & 9 & 24 \\ 18.9 & 26.2 & 37.4 & 34.1 & 43.3 & 36.1 & 14.6 & 9 & 0 & 15 \\ 24 & 31.3 & 42.5 & 39.7 & 53.6 & 51.1 & 29.6 & 24 & 15 & 0 \end{pmatrix}.$$

且 M_{10} 表明了各个起点与应急通道之间最优路径的距离，例如该矩阵第一行表示 A 起点到 $A, B, S_2, G, C, S_3, D, E, F, S_1$ 点的最短路径长度，而比较 A 点到三个终点的路径长度，发现最佳逃生路径为出口 S_2 ，距离为 18.5 米。由此总结出所有起点对应的最佳疏散出口如表 1 所示：

Table 1. Escape path table
表 1. 疏散路径表

起点	A	B	C	D	E	F	G
最佳出口	S_2	S_2	S_3	S_3	S_3	S_1	S_1

但是在某种紧急情况下，可能出现应急出口因为安全原因而无法使用，这时就需要重新考虑各个地点与应急之间的最优路径问题了。于是本文又考虑了当三个出口分别出现问题，无法使用时，这七个起点如何选择路线。

将无法使用的应急出口与各个地点间的距离权重设置为 inf。假设 S_1 无法使用，我们首先得到初始距离矩阵

$$M_0 = \begin{pmatrix} 0 & 7.3 & 18.5 & 16.4 & \text{inf} & 42.8 & 35.6 & 24.1 & 18.9 & \text{inf} \\ 7.3 & 0 & 12 & 15.3 & \text{inf} & 53 & 49.9 & 46.2 & 41 & \text{inf} \\ 18.2 & 12 & 0 & 25.6 & 38.8 & \text{inf} & 67.4 & 56.2 & 50.1 & \text{inf} \\ 16.4 & 15.3 & 25.6 & 0 & 13.9 & 23.7 & 27.1 & 31.3 & 34.1 & \text{inf} \\ \text{inf} & \text{inf} & 38.8 & 13.9 & 0 & 20 & 28.9 & 34.3 & 44.9 & \text{inf} \\ 42.8 & 53 & \text{inf} & 13.9 & 20 & 0 & 21.5 & 32.6 & 45.6 & \text{inf} \\ 35.6 & 49.9 & 67.4 & 27.1 & 28.9 & 21.5 & 0 & 5.6 & 16 & \text{inf} \\ 24.1 & 46.2 & 56.2 & 31.3 & 34.3 & 32.6 & 5.6 & 0 & 9 & \text{inf} \\ 18.9 & 41 & 50.1 & 34.1 & 44.9 & 45.6 & 16 & 9 & 0 & \text{inf} \\ \text{inf} & 0 \end{pmatrix}$$

利用 Floyd 算法计算后得到最终的矩阵：

$$M_{10} = \begin{pmatrix} 0 & 7.3 & 18.5 & 16.4 & 30.3 & 40.1 & 29.7 & 24.1 & 18.9 & \text{inf} \\ 7.3 & 0 & 12 & 15.3 & 29.2 & 39 & 37 & 31.4 & 26.2 & \text{inf} \\ 18.2 & 12 & 0 & 25.6 & 38.8 & 49.3 & 47.9 & 42.3 & 37.1 & \text{inf} \\ 16.4 & 15.3 & 25.6 & 0 & 13.9 & 23.7 & 27.1 & 31.3 & 34.1 & \text{inf} \\ 30.3 & 29.2 & 38.8 & 13.9 & 0 & 20 & 28.9 & 34.3 & 43.3 & \text{inf} \\ 30.3 & 29.2 & 39.5 & 13.9 & 20 & 0 & 21.5 & 27.1 & 36.1 & \text{inf} \\ 29.7 & 37 & 48.2 & 27.1 & 28.9 & 21.5 & 0 & 5.6 & 14.6 & \text{inf} \\ 24.1 & 31.4 & 42.6 & 31.3 & 34.3 & 27.1 & 5.6 & 0 & 9 & \text{inf} \\ 18.9 & 26.2 & 37.4 & 34.1 & 43.3 & 36.1 & 14.6 & 9 & 0 & \text{inf} \\ \text{inf} & 0 \end{pmatrix}$$

从而得到在 S_1 无法使用的情况下所有起点对应的最佳出口如表 2 所示：

Table 2. Evacuation path without S_1 exit
表 2. S_1 出口无法使用时的疏散路径

起点	A	B	C	D	E	F	G
最佳出口	S_2	S_2	S_3	S_3	S_3	S_3	S_3

同理可得当 S_2 无法使用的情况下所有起点对应的最佳出口如表 3 所示：

Table 3. Evacuation path without S_2 exit
表 3. S_2 出口无法使用时的疏散路径

起点	A	B	C	D	E	F	G
最佳出口	S_1	S_1	S_3	S_3	S_3	S_1	S_1

S_3 无法使用的情况下所有起点对应的最佳出口如表 4 所示：

Table 4. Evacuation path without S_3 exit
表 4. S_3 出口无法使用时的疏散路径

起点	A	B	C	D	E	F	G
最佳出口	S_2	S_2	S_2	S_2	S_1	S_1	S_1

6. 结论

本文利用经典 Floyd 算法求解了一个建筑空间的疏散路径设计问题。并给出了当某个紧急出口无法使用情况下如何重新选择路径的方法。

基金项目

大连民族大学大学生创新创业项目(201912026039, 201912026449), 大连民族大学理学院信息与计算科学专业建设项目。

参考文献

- [1] 车建涛, 高方玉, 解玉文, 李端玲, 牛坤, 马士恩. 基于 Dijkstra 算法的水下机器人路径规划[J]. 机械设计与研究, 2020, 36(1): 44-48.
- [2] 唐志华. 基于 floyd 和遗传算法的应急物流 LRP 优化研究[J]. 现代商贸工业, 2019, 40(19): 25-28.
- [3] 张巍, 姜大立, 周振, 徐建楠. 基于 Floyd 算法的常规导弹连续波次作战运输规划[J]. 国防科技, 2019, 40(3): 18-27.
- [4] 邱晓鹏, 王丽君. 基于 Floyd 算法的最短路径优化研究[J]. 太原师范学院学报(自然科学版), 2019, 18(2): 53-56+67.