

Research Article

Design and Analysis of Hospital Throughput Maximization Algorithm under COVID-19 Pandemic

Haochen Zou , Geer Jiang, Bowen Cheng, and Dejian Wang

Department of Computer Science and Software Engineering, Concordia University, Montreal, QC, Canada H3G 1M8

Correspondence should be addressed to Haochen Zou; haochen.zou@mail.concordia.ca

Received 27 May 2022; Accepted 18 July 2022; Published 11 August 2022

Academic Editor: Plácido R. Pinheiro

Copyright © 2022 Haochen Zou et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Under the global pandemic of COVID-19, public health facilities, such as hospitals, are required to readjust, design, and plan a safe movement flow of people to meet the social distance rules and quarantine COVID-19 and the non-COVID-19 patients to prevent cross-infection. However, readjustments to separate patients have significantly reduced the maximum throughput of public health facilities, worsening already scarce public health resources. Therefore, this paper proposes throughput maximization algorithms based on the one-way street problem which meets the requirements of social distance rules. First, the floor plan of a hospital is transformed into a graph, each node is traversed by breadth-first search. Then, this paper considers patients' node pair sets as different set unions, the direction of edges, and the color of links based on DFS-XOR algorithm are designed to distinguish the paths of COVID-19 and non-COVID-19 patients. Finally, this paper utilizes minimum shared link algorithms to determine the minimized sharing links between paths linking different set unions and components. The throughput is maximized by reducing the number of shared links and alternating links. The results indicate that compared with the brute force algorithms, the algorithms proposed in this paper significantly improve the maximum throughput.

1. Introduction

The COVID-19 pandemic broke out in December 2019 and quickly swept the world, with the virus affecting the global economy and everyone's daily lives. The COVID-19 pandemic has introduced unprecedented challenges in the world. Due to the emergence and prevalence of COVID-19 and variant strains of COVID-19 such as Omicron, hospitals and other public health facilities are required to readjust, design, and plan a safe movement flow of people to meet social distancing requirements [1]. The goal is to isolate COVID-19 patients from non-COVID-19 populations to avoid crossinfection and thereby reduce the number of COVID-19 cases [2]. However, although the redesigned flow of movement meets the requirements of social distance, the ability of public health facilities to accommodate patients is significantly reduced due to the need for extra spaces. This worsens the already strained public medical resources. At the same time, doctors and nurses may also be infected with COVID-19 during work and leave their posts, which exacerbates the shortage of hospital staff and further reduces

hospitals' patient intake levels [3]. Therefore, this paper intends to solve the problem of how to redesign and plan the movement routes of people in hospitals and other public health facilities under the premise of meeting the requirements of social distancing in order to maximize throughput.

The arrival of infectious variants of COVID-19, such as Omicron, has led to an exponential and dramatic increase in confirmed COVID-19 cases in a relatively short period, and the number of patients needing to visit public health facilities grows as well [4]. Much of the disaster planning in hospitals around the country addresses overcrowded emergency departments and decompressing these locations. However, in the case of COVID-19 pandemic, intensive care units, emergency departments, and medical wards ran the risk of being overwhelmed by a large influx of patients needing high-level medical care [5]. Sarier et al. contradistinguished the number of patients attending the hospital urology clinic during the first year of the COVID-19 pandemic compared to 2018 and 2019 [6]. Previous works have been conducted to modify and deploy existing hospital consultation plans to quickly care for large numbers of

medically complex patients. These studies are designed to maximize the throughput of patients that hospitals can accommodate. Bowden et al. described a scale-up approach to managing residents' clinical practices during the COVID-19 pandemic. Their study provides a framework for maximizing patient admission while also leveraging a broad clinician workforce, minimizing exposure, and maximizing the pool of clinicians who may not be involved in inpatient care [5]. However, their study is aimed at describing a scale-up approach to managing residents' clinical practices during the COVID-19 pandemic, targeting physicians rather than hospitals, since the situation of physicians varies greatly from hospital to hospital. Therefore, although the plan has been successful in their institutions, it needs to be significantly modified for other public health facilities. Meanwhile, their plan does not consider the social distance rules. Shahverdi et al. proposed models for assessing strategies for improving hospital capacity for handling patients during the pandemic [7]. It enables hospitals to repurpose space, modify operations, implement crisis standards of care, collaborate with other health care facilities, or request external support, thereby increasing hospital capacity. However, the strategy evaluation model they designed is mainly focusing on the routine emergency and emergency care under the pandemic, which is difficult to apply to the overall situation of hospitals and other public health facilities. Meanwhile, only a fraction of all COVID-19 cases requires emergency care, meaning that the majority of COVID-19 patients will not benefit from the above strategic assessment model [8]. Olanipekun focuses on preventing delays in transferring patients to long-term lower acuity level nursing facilities, reducing length of hospital stay, improving patient flow, and ultimately freeing up hospital beds for incoming COVID-19 patients [9]. The above-described approach maximizes the throughput of patients. However, the process involves transferring patients to nursing facilities to shorten the length of stay, rather than improving hospitals' own situations.

Therefore, in addition to the approaches discussed above, this paper is aimed at designing hospital throughput maximization algorithms which meet the requirements of social distance and can be widely implemented by most hospitals and other public health institutions. Based on the aforesaid objectives, we first transformed the floor plan of a hospital into a strong orientation connected graph for implementing graph algorithms [10]. Then, the patient node pair sets are treated as the union of different sets, the direction of edges, and the color of links are assigned based on the DFS-XOR algorithm to distinguish the walking paths of COVID-19 and non-COVID-19 patients. Finally, the minimum shared edge algorithm is implemented to determine the minimization of shared links between paths connecting different sets of unions and components. The maximized throughput is achieved by reducing the number of shared links and alternating links. The contrast experiment indicates that compared with brute force algorithm to split the walking paths of COVID-19 and non-COVID-19 patients, the algorithm proposed in this paper improves the maximum throughput of patients in public health institutions such as hospitals.

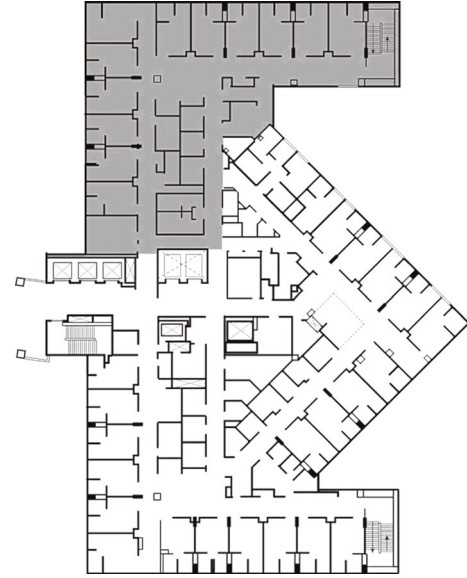


FIGURE 1: The floor plan of McGill University Health Centre.

2. Materials and Methods

2.1. Convert Hospital Floor Plans to Graphics. The throughput maximization algorithm proposed in this paper is based on the graph algorithm. Therefore, the first step of the method is to transform the floor plans of hospitals and other public health facilities into graphs. A floor plan of the McGill University Health Centre is displayed in Figure 1. The graph is composed of the set of nodes and the set of edges. Nodes can be further divided into ordinary nodes and special nodes [11]. In a hospital floor plan, the corridors are the edges of the graph.

The entrances or exits of rooms can be considered as the ordinary nodes of the graph. The main entrance of the floor, the main exit of the floor, and the intersections of the corridors are decided as the special nodes of the graph.

Based on the above definition of nodes and edges in the graph, the hospital floor plan can be transformed into an undirected graph $G = (V, E)$, where V is the set of nodes and E is the set of edges. Automatic understanding of floor plan images and converting floor plans to graphs could be accomplished by the deep recognition framework proposed by Lu et al. [12]. A sample of transformation and the result are displayed in Figure 2.

Under the background of the COVID-19 pandemic, some consulting rooms of hospitals are required to treat the COVID-19 patients; the other consulting rooms are receiving non-COVID-19 patients [13]. In public health, social distancing, also called physical distancing, is a set of nonpharmaceutical interventions or measures intended to prevent the spread of a contagious disease by maintaining a physical distance between people and reducing the number of times people come into close contact with each other [14]. Canada adopted a two-meter (approximately 6ft) social distancing policy [15]. Although the requirement of social distance in public places has been gradually abolished, the



FIGURE 2: A sample of transformation from a floor plan to a graph: (a) the floor plan; (b) the transformed graph.

restriction of social distance is still strictly enforced in public health facilities such as hospitals [16].

Hospital corridors are clearly inadequate for social distance rules, especially when there is a large flow of people. Therefore, COVID-19 patients and other patients cannot share the same corridor at the same time. When one corridor is shared by both COVID-19 patients and other

patients, we consider the edge of this corridor as the shared edge.

Therefore, the hospital floor plan further consists of the following parts: consulting rooms for COVID-19 and non-COVID-19 patients with entrances and exits, the main entrance and exit of the floor, corridors for patients to move, and intersections of corridors. The sample floor plan of a

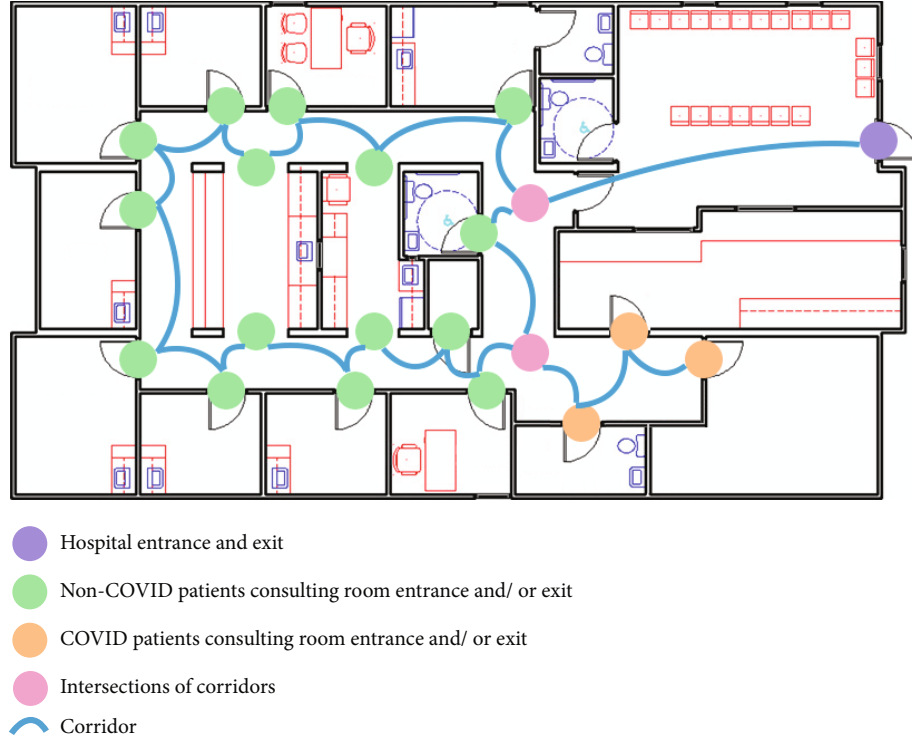


FIGURE 3: The sample floor plan of a hospital with COVID-19 and non-COVID-19 node sets.

hospital in a graph with COVID-19 and non-COVID-19 node sets is displayed in Figure 3.

The hospital floor plan is eventually converted into an undirected graph. In the graph, we require that each consulting room must have (1) at least one path from the main entrance of the floor to the entrance of the consulting room; (2) at least one path from the exit of the consulting room to the main exit of the floor; and (3) at least one path from the exit of any consulting room to the entrance of any other consulting room within the same node set, to meet the accessibility. Therefore, there must be some edges that need to be considered as shared edges as defined above. In the graph algorithm, the edge is undirected, and the link is directed [17]. The flow of people on one edge can be maximized when the edge is oriented as a one-way link, while to satisfy the accessibility, some of the edges must be determined as two-way links; these links are defined as alternating links. Alternating links are the links that are used in both directions, utilized alternately in each direction, with the usage of a single direction at any given time and a waiting area at each end of the links associated with alternating links. Throughput is rate of production or the rate at which something is processed [18]. In the context of the paper, throughput is the number of people reaching their destination per time unit [19]. As the shared links defined and discussed above, the existence of alternating links and shared links reduces the throughput of patients in a hospital because when one group is passing, the other group must stay at one end and wait, which significantly weakens the traffic efficiency of the flow. Shared links and alternating links contain waiting areas at their origins, which correspond

to converge flows, and therefore lead to a source of flow slowdown [20].

Hence, the essence of the throughput maximization algorithm is to minimize the number of shared links and alternating links in the graph of the hospital floor plan.

2.2. Label Colors of the Edges. Given the undirected graph of a floor plan $G = (V, E)$ computed from the above section, two sets of node pairs are generated: SD1: the non-COVID-19 node pairs set, and SD2: the COVID-19 node pairs set. Sample SD1 and SD2 is shown in Figure 3, as the non-COVID-19 and COVID-19 patients consulting room entrances and/or exits separately. Nodes from SD1: $s_1, s_2, \dots, s_n \in V$. Nodes from SD2: $s'_1, s'_2, \dots, s'_n \in V$. Particularly, special nodes can be considered as belonging to both node pairs set, as the main entrance of the floor, the main exit of the floor, and the intersections of the corridors can be passed by both non-COVID-19 and COVID-19 patients. Corridors $l_1, l_2, \dots, l_n \in E$. The goal is to label the same color link paths for individual node sets, such that COVID-19 patients walk along the red color link paths, and non-COVID-19 patients walk along the green color link paths, therefore, to detect all edges $l \in E$ and $l' \in E$ in the graph G where $l(u, v)$, $u \in \text{SD1}$, $v \in \text{SD1}$, and $l'(u', v')$, $u' \in \text{SD2}$, $v' \in \text{SD2}$.

The breadth-first search (BFS) algorithm is implemented on the undirected graph $G = (V, E)$ to make sure that all nodes in each set are reachable from the root nodes which are the main entrance [21]. The algorithm is described below in Algorithm 1.

```

1:  procedure BreadthFirstSearch ( $G(V, E)$ , SD, root)
2:    for each vertex  $u$  in  $G(V)$ 
3:       $u.color \leftarrow$  WHITE
4:      //initializing all vertices in the given graph
5:       $u.d \leftarrow \infty$ 
6:       $u.\pi \leftarrow$  NIL
7:    end for
8:     $u \leftarrow$  root
9:     $P \leftarrow$  EmptyArrayList
10:    $P \leftarrow P \cup$  MODIFIEDBFS( $G, SD, u$ )
11:   //ModifiedBFS algorithm with specified root vertex
12:   for each vertex  $u \in G(V) - \{root\}$ 
13:     if  $u.color$  is WHITE
14:        $P \leftarrow P \cup$  MODIFIEDBFS( $G, SD, u$ )
15:     //ModifiedBFS algorithm with unreached vertices
16:     end if
17:   end for
18:   return  $P$ 

```

ALGORITHM 1: Breadth-first search algorithm.

Apply BFS search with the condition that only the node with SD1 or SD2 sets can be added to the queue [22]. The breadth-first tree formed after running the traditional algorithm may not visit all the vertices in some graphs for instance directed cyclic and acyclic graphs. Therefore, the traversing may be incomplete. We traverse the graphs with the usage of a modified BFS algorithm discussed below in Algorithm 2 [23].

The result of Algorithm 1 and Algorithm 2 will be the data structure of linked list as edges to form a path. It stores the node objects for COVID-19 patients and non-COVID-19 patients. We implement a Color Path Algorithm to color the original graph with two undirected paths consists of edges, as discussed in the Algorithm 3.

The total running time of the proposed algorithm is $O(V + E)$ [24], since the most time taken part is custom BFS by using the linked list data structure to store the whole graph. After above steps, two datasets are formed. There are two undirected paths in the datasets. One is for COVID-19 patients, and the other one is for non-COVID-19 patients.

2.3. Define Orientation to Minimize Alternating Links. In this section, we define the orientation of edges in the undirected paths. The aim is to minimize the alternating links in the process of orientation, and meanwhile, satisfy the requirements of accessibility. The one-way street problem is a graph orientation problem, which means to define the orientation of edges (undirected links) subject to conditions [25]. The definition of the one-way street problem is as follows: consider an undirected graph, is it possible to choose a direction for each edge, turning it into a directed graph that has a path from every vertex to every other vertex [26]? Strongly connected components satisfy the requirement of the one-way street problem [27]. Therefore, we need to detect the strongly connected components inside each graph formed by the undirected paths.

```

1:  procedure ModifiedBFS ( $G(V, E)$ , SD,  $u$ )
2:     $u.color \leftarrow$  GREY
3:     $u.d \leftarrow 0$ 
4:    //tree from the specified root vertex
5:     $u.\pi \leftarrow$  NIL
6:     $Q \leftarrow$  EmptyQueue
7:     $P \leftarrow$  EmptyArrayList
8:    ENQUEUE( $Q, u$ )
9:    while not EMPTY( $Q$ )
10:      $u \leftarrow$  DEQUEUE( $Q$ )
11:     if  $u \in$  SD
12:        $P \leftarrow P \cup \{u\}$ 
13:     end if
14:     for each vertex  $v$  in  $ADJ[u]$ 
15:       if  $v.color$  is WHITE
16:          $u.color \leftarrow$  Grey
17:          $u.d \leftarrow u.d + 1$ 
18:          $u.\pi \leftarrow u$ 
19:         ENQUEUE( $Q, v$ )
20:       end if
21:     end for
22:      $u.color \leftarrow$  BLACK
23:   end while
24:   return  $P$ 

```

ALGORITHM 2: Modified BFS algorithm.

For the two undirected graph generated by the two undirected paths, we first consider all the edges as alternating links to make the graph directed. Then, we detect the strongly connected components inside the graph. If inside the node sets, there exist cycle component connected nodes, the direction of the links formed the cycle can be defined as one side, and the cycle is considered as strongly connected component [28]. Otherwise, inside the component, some nodes may not be reachable under the one-direction way street problem's condition; hence, alternating links will be


```

1:  procedure GetAllEdges( $P$ )
2:     $P1 \leftarrow$  EmptyArrayList
3:    //path  $P$  is an array list with node objects
4:    for  $i \leftarrow 1$  to length of  $P$ 
5:      if  $i + 1 \leq$  length of  $P$ 
6:         $P1 \leftarrow P1 \cup (P[i], P[i + 1])$ 
7:    procedure RemoveUsedPath ( $G(V, E), P$ )
8:      Edges  $\leftarrow$  GETALLEDGES( $P$ )
9:      for each  $e \in$  Edges
10:       if  $e \in G(E)$ 
11:          $G \leftarrow G - \{e\}$ 
12:       end if
13:     end for
14:     return  $G$ 
15:  procedure TraverseEdges ( $G(V, E), SD1, SD2$ )
16:    ///Assume that  $SD1$  is non-COVID nodes dataset
17:    ///Assume that  $SD2$  is COVID nodes dataset
18:     $G1 \leftarrow$  DirectedGraphConvert( $G$ )
19:     $P \leftarrow$  BreadthFirstSearch( $G1, SD1, \text{root}$ )
20:    if  $P$  covers all nodes in  $SD1$ :
21:       $P1 \leftarrow$  GetAllEdges( $P$ )
22:       $G2 \leftarrow$  RemoveUsedPath( $G1, P1$ )
23:    end if
24:     $P2 \leftarrow$  BreadthFirstSearch( $G2, SD2, \text{root}$ )
25:    if  $P2$  covers all nodes in  $SD2$ :
26:       $P3 \leftarrow$  GetAllEdges( $P2$ )
27:    end if
28:    result $G \leftarrow$  ColorPath( $G1, P1, P2$ )
29:    return result $G$ 
30:  procedure ColorPath ( $G(V, E), P1, P2$ )
31:    /// $P1$  belongs to non-COVID edges generated from BFS algorithm
32:    /// $P2$  belongs to COVID edges generated from BFS algorithm
33:    for each  $e \in G(E)$ 
34:      if  $e \in P1$ 
35:        Color( $e, \text{GREEN}$ )
36:      else if  $e \in P2$ 
37:        Color( $e, \text{RED}$ )
38:      else
39:        Color( $e, \text{GREY}$ )
40:    return  $G$ 

```

ALGORITHM 3: Color path algorithm.

implemented. The direction of a strongly connected component cycle path will be defined from the entry to the exit or vice versa. The DFS-XOR algorithm is implemented for detecting whether there are cycle components in the graph or not as Algorithm 4 displayed below.

The total running time of the proposed algorithm is $O(V + E)$ [29], since the most time taken part is custom depth-first search (DFS) by using the queue data structure to store the whole graph. The graph with define orientation paths after processing the cycle detection algorithm is displayed in Figure 4.

2.4. Define Orientation to Minimize Shared Links. We intend to confirm the orientation of the left alternating links in the graph, detect, and minimize the shared links to achieve the goal of maximum throughput. The current graph of a floor plan is shown in Figure 5. As displayed, there are links with

already defined orientation, such as the red links and the green links in the graph. There also exist alternating links which required to be further oriented, such as the grey links in the graph.

A strong orientation is an orientation that results in a strongly connected graph. Robbins' theorem states that a graph has a strong orientation if and only if it is two-edge-connected, for example, if the graph has no bridge assuming the graph is connected [30]. An orientation of a graph G is an assignment of a direction to each edge of G , which obtains, as a result, a digraph. Let G be a connected graph and B be one of its edges. We define B as a bridge of G if the graph $G1$ obtained by removing B from G is disconnected [31]. We can conclude from the above description that the shared link is the bridge.

Identification of bridges can be done in linear time using Trajan's algorithm [32]. Multiple techniques can be

```

1:  procedure DFSCycleCountXOR ( $G(V, E)$ )
2:    //initialization
3:     $n \leftarrow 0$ 
4:    // $n$  stores number of discovered cycles in operation
5:    CycleQueue  $\leftarrow$  NIL
6:    //stores the discovered cycles
7:    TotalCycleQueue  $\leftarrow$  NIL
8:    //stores the discovered cycles
9:    //end of initialization
10:   DFSVisit( $u$ )
11:   //calling DFS for computing initial set of cycles
12:   TotalCycleQueue  $\leftarrow$  TotalCycleQueue + CycleQueue
13:   for  $l = 2$  to  $n$ 
14:     // $l$  stores the level of XOR to be done between the unique cycles
15:     for each combination comp of size =  $l$  in CycleQueue
16:       cycle  $\leftarrow$  XOR all cycles in comp
17:       if cycle is a valid cycle
18:         then TotalCycleQueue  $\leftarrow$  cycle + TotalCycleQueue
19:   DFSVisit( $u$ )
20:   set  $u$  to gray
21:   time  $\leftarrow$  time + 1
22:    $d[u] \leftarrow$  time
23:   for each  $v \in$  Adjacent[ $u$ ]
24:     if  $v$  is white
25:       then DFSVisit( $v$ )
26:     if  $v$  is white
27:       then CycleQueue  $\leftarrow$  CycleQueue + new cycle
28:   //discovered using back tracking the DFS path from vertex  $V$  to itself
29:    $n = n + 1$ 
30:   //count number of discover cycles
31:   set  $u$  to black

```

ALGORITHM 4: DFS cycle count XOR algorithm.

implemented to minimize the number of shared links. Computing a Hamiltonian path or a cycle (Hamiltonian path with the option of going several times through the same nodes) is a mechanism for reducing the number of shared links [33]. The algorithm to compute a Hamiltonian path or a cycle is the DFS and backtracking algorithm [34]. Overall, the minimum shared edges (MSE) problem is defined as follows.

Given a directed graph $G = (V, E)$, two special nodes $s, t \in V$, and integer $k > 0$. Find a set P of k paths from s to t in G so as to minimize $c(P) = \sum_{e \in E} \lambda(e)$, where $\lambda(e) = 0$ if e is used in at most one path of P , and $\lambda(e) = 1$ otherwise. An edge e with $\lambda(e) = 1$ is called a shared edge. As discussed by Omran et al., the MSE problem is NP-hard [35].

We implement a modified depth-first search (DFS) method to detect and minimize the number of bridges. The DFS algorithm generates four types of links: tree, forward, backward, and transversal [36]. Consider the DFS algorithm, while we are looking for vertices adjacent to vertex v , will be a bridge if and only if none of the vertex's u or any of its descendants in the DFS traversal tree has a back edge to vertex v or any of its ancestors. We can check the existence of back edges in $O(V + E)$ time as the domain time to do the depth-first search. Let $TIN(v)$ denotes the entry time for node v . We introduce an array LOW which will

let us check the fact for each vertex v . $LOW(v)$ is the minimum of $TIN(v)$, entry times $TIN(p)$ for each node p that is connected to node v via a back-edge (v, p) , and the values of $LOW(w)$ for each vertex w which is a direct descendant of v in the DFS tree:

$$LOW(v) = \begin{cases} TIN(v), \\ TIN(p), \text{ for all } p \text{ such that } (v, p) \text{ is a back edge,} \\ TIN(v), \text{ for all } w \text{ such that } (v, w) \text{ is a tree edge.} \end{cases} \quad (1)$$

After the above processes, there is a back edge from vertex v or one of its descendants to one of its ancestors if and only if vertex v has a child w for which $LOW(w) \leq TIN(v)$. If $LOW(w) = TIN(v)$, the back edge comes directly to v ; otherwise, it comes to one of the ancestors of v . Therefore, the current edge (v, w) in the DFS tree is a bridge if and only if $LOW(w) > TIN(v)$ [37].

The minimized number of shared links is detected according to the method. The shared link is used by both COVID-19 and non-COVID-19 patients; therefore, such should be orientated in both directions. Figure 6 shows the graph of a hospital's floor plan with the designed movement

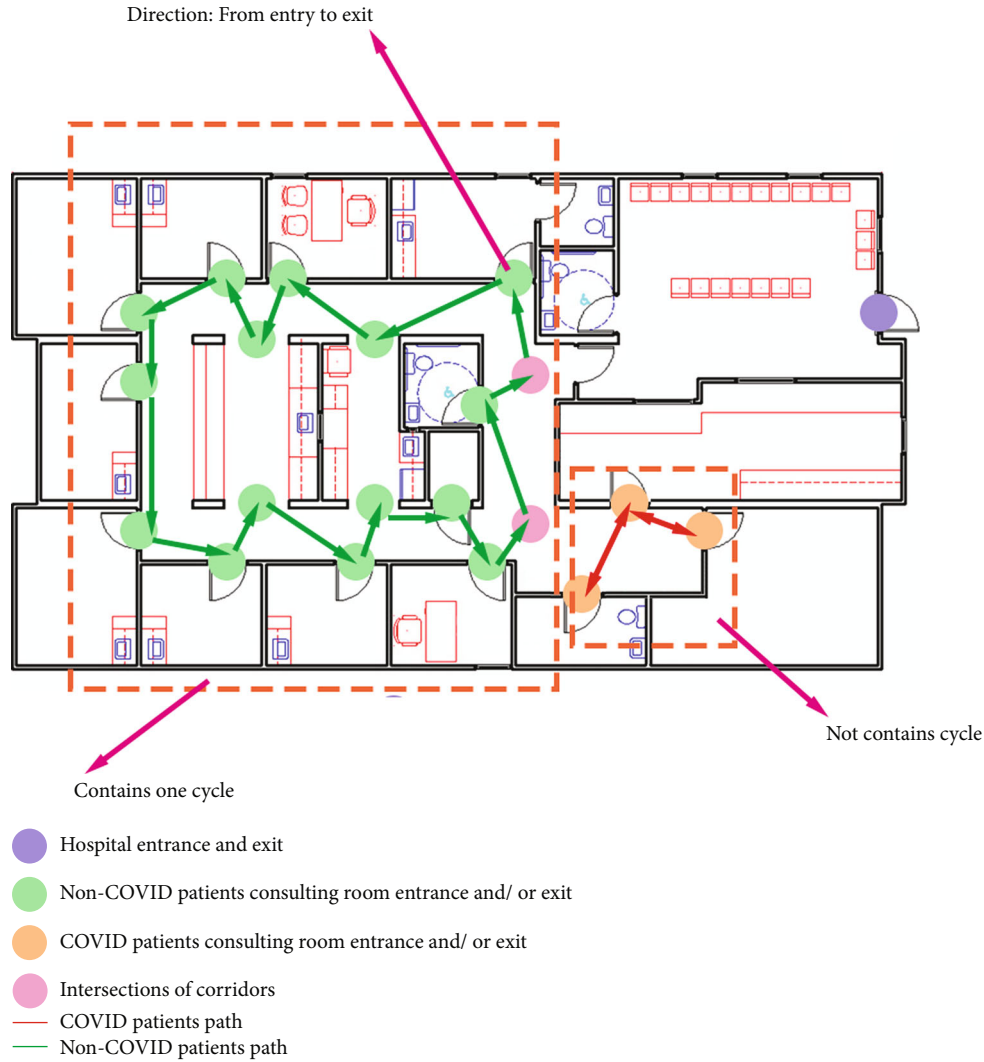


FIGURE 4: The graph with direction after algorithm modification.

flow of patients after all the algorithms' modifications. In the movement flow designed in the figure, the maximum throughput of the floor plan is obtained by reducing the number of alternating links and shared links.

3. Results and Discussion

3.1. Results. This section presents the analysis result of the designed hospital throughput maximization algorithm. There are two crowd movement path flows for one general hospital; one is according to the brute force method, and the other one is designed based on maximum throughput algorithm proposed in this paper. We establish the crowd flow simulation model of two different paths to calculate and count the time of the crowd consumed by moving from the exits of the consulting rooms to the main exit of the floor. The time difference between the two paths is compared to achieve the purpose of contradicting throughput. Since throughput is the number of people reaching their destination per time unit. Based on the same

amount of people in the crowd, the longer the consumption on the path, the smaller the corresponding throughput.

Pathfinder is selected for simulation in this paper. Pathfinder is a simulator based on human movement simulation. It provides users with a graphical user interface for simulation design and operation, as well as 2D and 3D visualization tools for analysing results [38]. Pathfinder utilizes a geometric model that supports 3D. This model includes rooms, doors, stairs, exits, and other structures. The three-dimensional structure diagram of a general hospital is shown in Figure 7.

The human is the main body of the simulation function, and the simulation model of the software allows to test the speed of the human, the width of the human model, the priority level, and special behaviours of the human. Pathfinder supports both steering and SFPE mobile simulation modes [39]. By using the hypothesis group in the SPFE handbook, the simulated personnel will not try to avoid each other when moving and will be crowded with each other [40]. The traveling speed of the crowd is affected by the space density of the room, and the pedestrian flow at the exit is

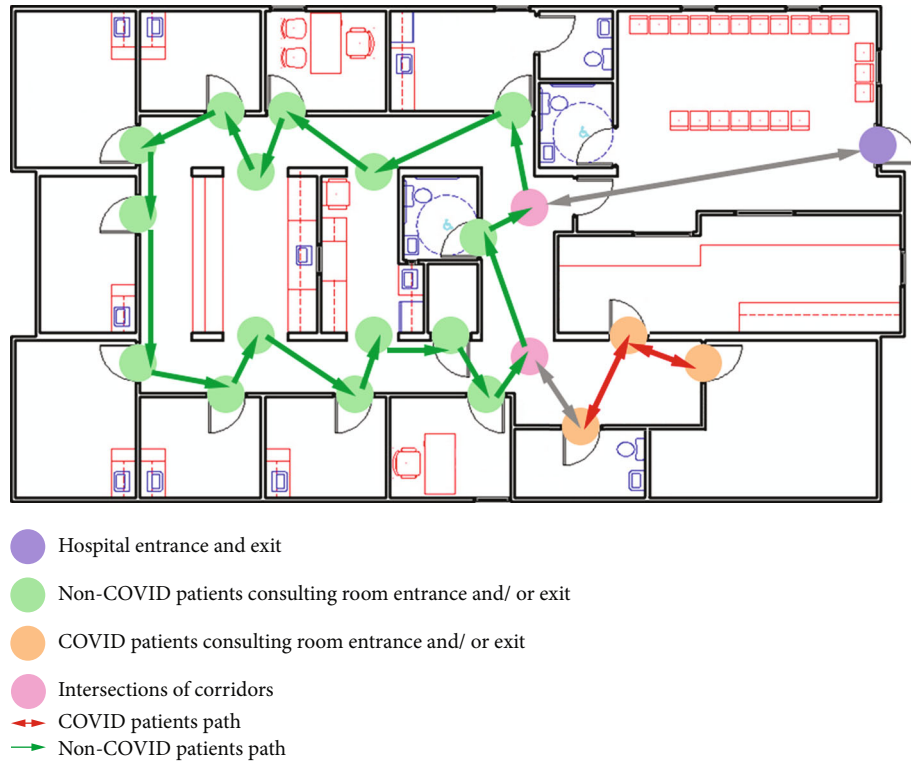


FIGURE 5: The current graph of a floor plan with oriented links and links to be oriented.

determined by the width of the door. In the steering mode, a reasonable distance will be maintained between people [41]. If the distance between moving people and the path of the nearest point reach or exceed a certain threshold, the system will automatically select other feasible paths and change the walking path of simulated personnel. Since the steering mode is closer to the real moving situation, the paper implements the steering mode in the simulation.

In the analysis of this paper, the population in the building is divided into eight types: adult male, sick male, adult female, sick female, normal old people, sick old people, normal children, and sick children. Simulated speed and simulated shoulder width of all kinds of personnel are adjusted according to the set parameters. The average shoulder width of personnel is set as 45 cm. According to the information provided by the hospital and field investigation, during the peak period of medical treatment, the flow of people on each floor of a hospital at a certain point is about 200 people. The building area of each floor is 1,500 square meters, and the crowd density is less than one person per square meter.

The movement speed and agility of eight different categories of personnel are different [42]. In the simulation analysis of personnel flow, the proportion and speed of personnel in the simulation software are shown in Table 1.

After generating corresponding simulators from various personnel data, we input the corresponding number of simulators into the system according to the expected personnel distribution and obtain the graph of generating simulators, as shown in Figure 8.

The two kinds of hospital personnel movement paths are generated based on the brute force method and the algo-

rithm proposed in this paper. The simulation model with simulators in consulting rooms and corridors of the hospital moving to the main exit of the floor along the movement paths is displayed in Figure 9.

In this paper, the first floor, second floor, and third floor plans of the hospital are selected for the establishment and research of simulation model. The path diagram of personnel movement is shown in Figure 10. Randomly distributed people at each floor will choose the nearest path to the exit.

The thermal map of personnel density is shown in Figure 11.

From the simulated personnel movement path and personnel density heat map, it can be seen that the randomly distributed people on the first, second, and third floors of the hospital will choose the path to the main exit of the whole floor nearby for movement. According to the statistical time, there are 450 people on the whole three floors. The total time to reach the main exit of each floor using the movement path generated by the brute force method is 335.9 seconds, while the total time to reach the main exit of each floor using the movement path generated by the algorithm proposed in this paper is 227.6 seconds. The algorithm introduced in this paper can improve the hospital throughput.

3.2. Discussion. The present study developed algorithms for maximizing the patients' throughput in public health institutions such as hospitals under the requirements of social distancing. Different various methods previously designed by other scholars to improve patient acceptance are the scale-up approach to managing residents' clinical practices

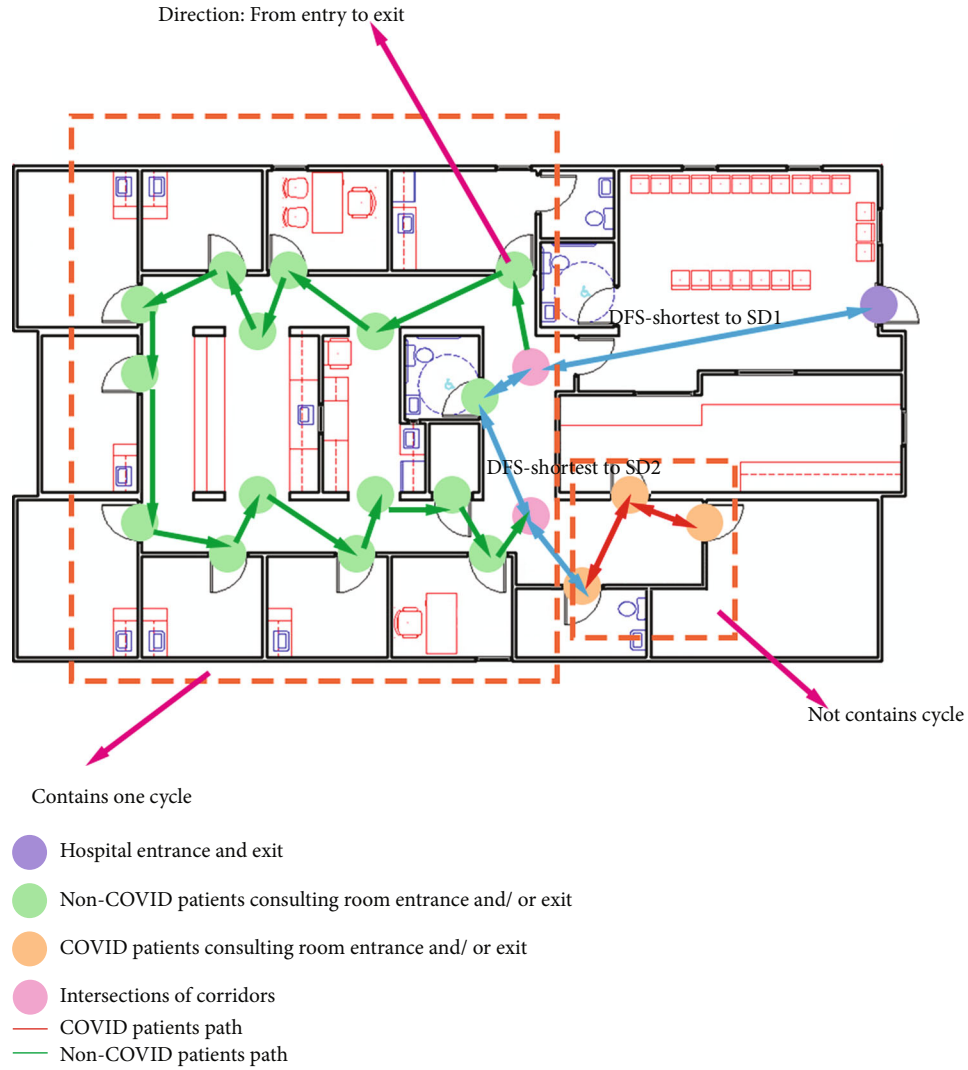


FIGURE 6: The movement flow of a floor plan graph after algorithm modification.

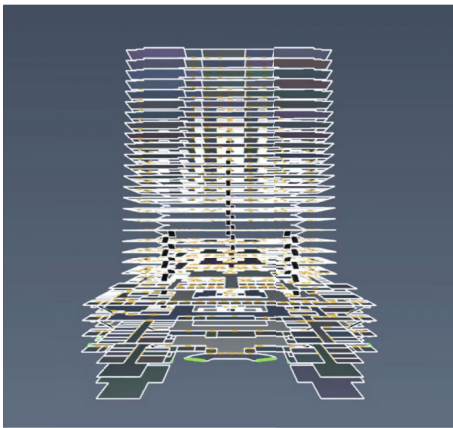


FIGURE 7: The three-dimensional structure diagram of a general hospital.

TABLE 1: Proportion of eight types of people and the walking speed.

Type of people	Proportion (%)	Walking speed (m/s)
Adult male	13	1.3
Sick male	25	1.0
Adult female	12	1.1
Sick female	25	0.85
Normal old people	3.25	0.8
Sick ole people	9.75	0.65
Normal children	3	1.0
Sick children	9	0.8

during the COVID-19 pandemic [5], the model for evaluating strategies to improve hospital patient handling capacity during the COVID-19 pandemic [6], and the approach of transferring patients to nursing institutions during the COVID-19 pandemic to shorten hospital stay and maximize the number of patients accepted by hospitals [8]. This paper

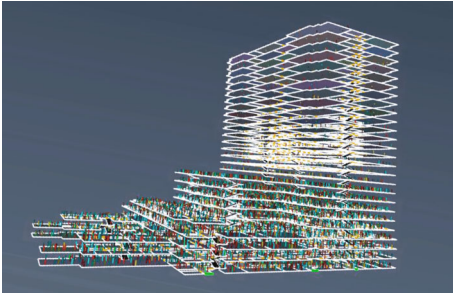


FIGURE 8: The three-dimensional structure diagram of a general hospital with simulators.

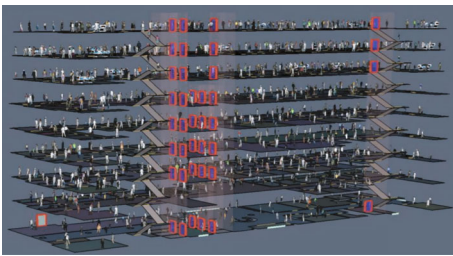


FIGURE 9: The simulation model with simulators.

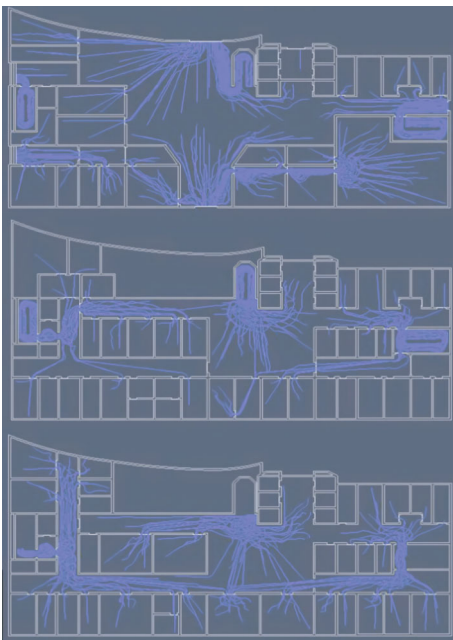


FIGURE 10: The path diagram of personnel movement.

focuses on the walking path of patients in the hospital, by applying the concept of maximum flow in the field of computer science and algorithm and reducing the number of shared links and alternating links, to achieve the goal of maximization the hospital throughput. The results of comparative experiments indicate that compared with the paths generated by the brute force method, the paths designed by the algorithm proposed in this paper can improve the throughput of hospitals by nearly 1.5 times. The study

extends the current literature. Researchers can expand this study by including other public health institutions and exploring other factors affecting the maximum throughput.

The maximum flow problem is a combinatorial optimization problem, which discusses how to make full use of the capacity of the equipment to maximize the flow and achieve the best effect [43]. The labelling algorithm for maximum flow was first proposed by Ford and Fulkerson in 1956. The “network flow theory” established by Ford and Fulkerson in the 1950s is an important component of network applications [44]. Network flow is a kind of specific flow solving method, which is closely related to linear programming. The theory and application of network flow are developing continuously, and new topics such as flow with gain, multiterminal flow, multicommodity flow, and the decomposition and synthesis of network flow appear [45]. Network flow has been widely used in communication, transportation, power, engineering planning, task assignment, equipment updating, and computer-aided design. In this paper, the application of the maximum flow problem is extended to medical fields such as public health facilities to meet social distancing requirements under the COVID-19 pandemic.

The COVID-19 pandemic and the emergence of COVID-19 variants such as Omicron have put a strain on already stretched public health resources, with existing facilities overwhelmed by the soaring number of people needing to visit hospitals. Based on the concept of the maximum flow algorithm, walking path algorithms satisfying social distance are designed in this paper. On the basis of the existing public health resources, the hospital can maximize the throughput of people by minimizing unnecessary waiting time and preventing dangerous situations happen such as crossinfection. At the same time, the results of this paper can also be applied to other places to prevent crossinfection of infectious diseases, such as mobile cabin hospitals. Efforts should be made to reduce the waste of public medical resources which are already scarce.

Our study focuses on developing pseudocode-based algorithms for maximizing the throughput in hospitals. Since different public health institutions implement different programming languages for their Hospital Information Systems (HIS), the original intention of designing algorithms utilizing pseudocode is to hope that public health institutions such as hospitals and clinics can implement the algorithm by themselves by applying the programming language they utilize. However, this may prove difficult. Meanwhile, when applying the algorithms to some relatively complex floor plan scenes, the results generated in each step of the algorithms may need to be slightly improved according to the actual situation. These improvements require algorithms, graph theory, and other related knowledge for relevant medical staff, which might cause inconvenience. The above-mentioned points lead to the limitations of the research. Thus, future studies should design programs and systems which can be run on portable devices [46], in order to achieve the automatic generation of walking paths that can help hospitals and other public health facilities obtain maximum flow. What is more, it is the goal of future

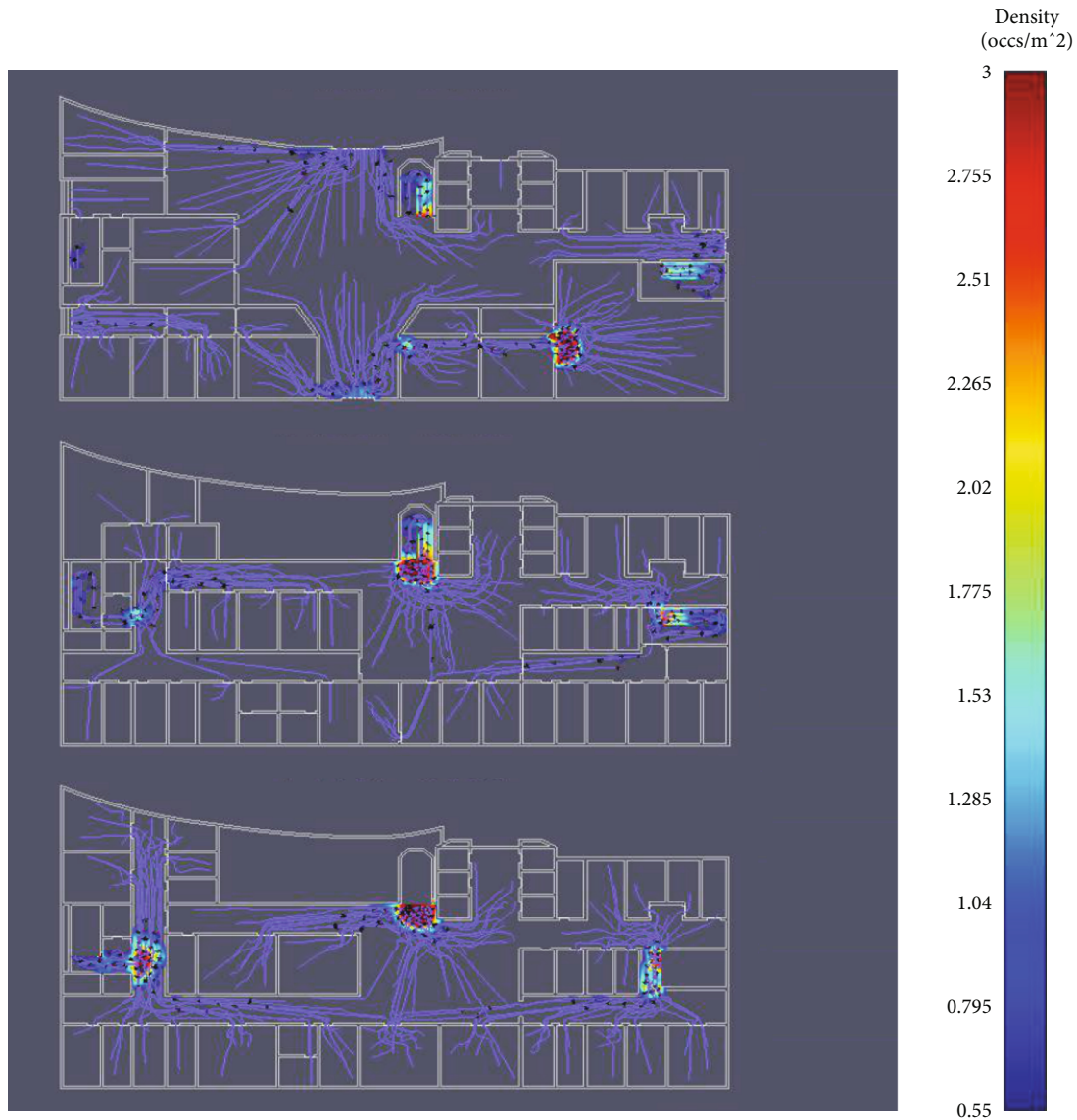


FIGURE 11: The path diagram of personnel movement.

research and works to improve the matching degree and accuracy of the results of the designed algorithm with the real public health institutions environment, so as to reduce the subsequent modification operations for medical staff. Meanwhile, future work will focus on reducing the number of iterations in the algorithm, such as loops and recursion, to improve the efficiency of the algorithm as much as possible and reduce the time and space complexity of the algorithm.

4. Conclusions

Due to the COVID-19 pandemic, hospitals and other public health facilities are being requested to redesign the way people walk in corridors to meet social distancing requirements. In this paper, algorithms based on the maximum flow problem are proposed in order to present movement flows and

satisfy social distancing requirements. The core idea of the algorithm is to minimize the number of shared links and alternating links. The experimental results indicate that compared with the traditional methods, the designed algorithms can improve the throughput of public health institutions and achieve the aim of maximizing the usage of public medical resources.

Data Availability

The data underlying the results presented in the study are available within the manuscript.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] M. Misra, H. Joshi, and R. Sarwal, "Exit strategies from lockdowns due to COVID-19: a scoping review," *BMC Public Health*, vol. 22, pp. 1–16, 2022.
- [2] J. Wong, Q. Y. Goh, Z. Tan et al., "Preparing for a COVID-19 pandemic: a review of operating room outbreak response measures in a large tertiary hospital in Singapore," *Canadian Journal of Anesthesia/Journal Canadien D'anesthésie*, vol. 67, no. 6, pp. 732–745, 2020.
- [3] Y. Akkuş, Y. Karacan, R. Güney, and B. Kurt, "Experiences of nurses working with COVID-19 patients: a qualitative study," *Journal of Clinical Nursing*, vol. 31, no. 9-10, pp. 1243–1257, 2022.
- [4] F. Grabowski, M. Kočańczyk, and T. Lipniacki, "The spread of SARS-CoV-2 variant Omicron with a doubling time of 2.0–3.3 days can be explained by immune evasion," *Viruses*, vol. 14, no. 2, p. 294, 2022.
- [5] K. Bowden, E. L. Burnham, A. Keniston et al., "Harnessing the power of hospitalists in operational disaster planning: COVID-19," *Journal of General Internal Medicine*, vol. 35, no. 9, pp. 2732–2737, 2020.
- [6] M. Sarier, M. Demir, M. Emek et al., "Comparison of spermograms of infertile men before and during the COVID-19 pandemic," *Revista da Associação Médica Brasileira*, vol. 68, no. 2, pp. 191–195, 2022.
- [7] B. Shahverdi, E. Miller-Hooks, M. Tariverdi, H. Ghayoomi, D. Prentiss, and T. D. Kirsch, "Models for assessing strategies for improving hospital capacity for handling patients during a pandemic," *Disaster Medicine and Public Health Preparedness*, pp. 1–10, 2022.
- [8] S. L. Chang, N. Harding, and C. Zachreson, "Modelling transmission and control of the COVID-19 pandemic in Australia," *Nature Communications*, vol. 11, pp. 1–13, 2020.
- [9] T. Olanipekun, "The impact of COVID-19 testing on length of hospital stay and patient flow in hospitals," *Journal of Community Hospital Internal Medicine Perspectives*, vol. 11, no. 2, pp. 180–183, 2021.
- [10] R. Gao, M. Zhao, and T. Ye, "Jigsaw: indoor floor plan reconstruction via mobile crowdsensing," in *In Proceedings of the 20th annual international conference on Mobile computing and networking*, pp. 249–260, Maui, Hawaii, USA, 2014.
- [11] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi, "Defining and identifying communities in networks," *In Proceedings of the National Academy of Sciences*, vol. 101, no. 9, pp. 2658–2663, 2004.
- [12] Z. Lu, T. Wang, J. Guo et al., "Data-driven floor plan understanding in rural residential buildings via deep recognition," *Information Sciences*, vol. 567, pp. 58–74, 2021.
- [13] E. Yong, "Why health-care workers are quitting in droves," *The Atlantic*, 2021.
- [14] K. Prem, Y. Liu, T. W. Russell et al., "The effect of control strategies to reduce social mixing on outcomes of the COVID-19 epidemic in Wuhan, China: a modelling study," *The Lancet Public Health*, vol. 5, no. 5, pp. e261–e270, 2020.
- [15] S. Tremblay-Huet, T. McMorrow, E. Wiebe, M. Kelly, M. Hennawy, and B. Sum, "The impact of the COVID-19 pandemic on medical assistance in dying in Canada and the relationship of public health laws to private understandings of the legal order," *Journal of Law and the Biosciences*, vol. 7, no. 1, p. Isaa 087, 2020.
- [16] S. Grundel, S. Heyder, T. Hotz, T. K. S. Ritschel, P. Sauerteig, and K. Worthmann, "How much testing and social distancing is required to control COVID-19? Some insight based on an age-differentiated compartmental model," *SIAM Journal on Control and Optimization*, vol. 60, no. 2, pp. S145–S169, 2022.
- [17] M. Cheng and L. Yin, "Transmission scheduling in sensor networks via directed edge coloring," in *In Proceeding of the 2007 IEEE International Conference on Communications*, pp. 3710–3715, Glasgow, UK, June 2007.
- [18] R. Rizvi, E. P. Nguyen, M. D. Kowal et al., "High-throughput continuous production of shear-exfoliated 2D layered materials using compressible flows," *Advanced Materials*, vol. 30, no. 30, p. 1800200, 2018.
- [19] H. Zhang and X. Lu, "Vehicle communication network in intelligent transportation system based on internet of things," *Computer Communications*, vol. 160, pp. 799–806, 2020.
- [20] M. Ameli, J. P. Lebacque, and L. Leclercq, "Simulation-based dynamic traffic assignment: meta-heuristic solution methods with parallel computing," *Computer-Aided Civil and Infrastructure Engineering*, vol. 35, no. 10, pp. 1047–1062, 2020.
- [21] R. K. Ghosh and G. P. Bhattacharjee, "Parallel breadth-first search algorithms for trees and graphs," *International Journal of Computer Mathematics*, vol. 15, no. 1-4, pp. 255–268, 1984.
- [22] A. Bundy, *Catalogue of Artificial Intelligence Tools. Catalogue of Artificial Intelligence Tools*, Springer, Berlin, Heidelberg, 1986.
- [23] I. Baidari and A. Hanagawadimath, "Traversing directed cyclic and acyclic graphs using modified BFS algorithm," in *In 2014 Science and Information Conference*, pp. 175–181, London, UK, 2014, August.
- [24] B. Awerbuch and R. G. Gallager, "Distributed BFS algorithms in proceeding of the 26th annual symposium on foundations of computer science scfs," in *26th Annual Symposium on Foundations of Computer Science (sfcs 1985)*, pp. 250–256, Portland, OR, USA, October 1985.
- [25] V. Verbavatz and M. Barthelemy, "From one-way streets to percolation on random mixed graphs," *Physical Review E*, vol. 103, no. 4, article 042313, 2021.
- [26] R. Lewis, "Algorithms for finding shortest paths in networks with vertex transfer penalties," *Algorithms*, vol. 13, no. 11, p. 269, 2020.
- [27] S. Illgen and M. Höck, "Literature review of the vehicle relocation problem in one-way car sharing networks," *Transportation Research Part B: Methodological*, vol. 120, pp. 193–204, 2019.
- [28] A. Volkova, M. Niedermeier, and R. Basmadjian, "Security challenges in control network protocols: a survey," *IEEE Communications Surveys & Tutorials*, vol. 21, pp. 619–639, 2018.
- [29] X. Wan and H. Wang, "Efficient semi-external depth-first search," *Information Sciences*, vol. 599, pp. 170–191, 2022.
- [30] S. Špacapan, "The diameter of strong orientations of Cartesian products of graphs," *Discrete Applied Mathematics*, vol. 247, pp. 116–121, 2018.
- [31] S. I. Khalaf, V. Mathad, and S. S. Mahde, "Edge hub number in graphs," *Online Journal of Analytic Combinatorics*, vol. 14, pp. 1–8, 2019.
- [32] G. Yan, Q. Li, D. Guo, and X. Meng, "Discovering suspicious APT behaviors by analyzing DNS activities," *Sensors*, vol. 20, no. 3, p. 731, 2020.
- [33] R. Pereira, J. Oliveira, and M. Sousa, "Bioinformatics and computational tools for next-generation sequencing analysis in

- clinical genetics,” *Journal of Clinical Medicine*, vol. 9, no. 1, p. 132, 2020.
- [34] G. B. Mertzios and D. G. Corneil, “A simple polynomial algorithm for the longest path problem on cocomparability graphs,” *SIAM Journal on Discrete Mathematics*, vol. 26, no. 3, pp. 940–963, 2012.
- [35] M. T. Omran, J. R. Sack, and H. Zarrabi-Zadeh, “Finding paths with minimum shared edges,” *Journal of Combinatorial Optimization*, vol. 26, no. 4, pp. 709–722, 2013.
- [36] V. Ojha and G. Nicosia, “Backpropagation neural tree,” *Neural Networks*, vol. 149, pp. 66–83, 2022.
- [37] L. Roditty and U. Zwick, “A fully dynamic reachability algorithm for directed graphs with an almost linear update time,” *SIAM Journal on Computing*, vol. 45, no. 3, pp. 712–733, 2016.
- [38] I. Lochhead and N. Hedley, “Mixed reality emergency management: bringing virtual evacuation simulations into real-world built environments,” *International Journal of Digital Earth*, vol. 12, no. 2, pp. 190–208, 2019.
- [39] Y. H. Kim, J. S. Choi, T. F. Yuan, and Y. S. Yoon, “Building-information-modeling based approach to simulate strategic location of shelter in place and its strengthening method,” *Materials*, vol. 14, no. 13, p. 3456, 2021.
- [40] A. D’Orazio, L. Grossi, and D. Ursetta, “Egress from a hospital ward during fire emergency,” *International Journal of Safety and Security Engineering*, vol. 10, pp. 1–10, 2020.
- [41] M. Li, S. Zhu, J. Wang, and Z. Zhou, “Research on fire safety evacuation in a university library in Nanjing,” *Procedia Engineering*, vol. 211, pp. 372–378, 2018.
- [42] V. Alonso-Gutierrez and E. Ronchi, “The simulation of assisted evacuation in hospitals,” in *Proceeding of Fire and Evacuation Modelling Technical Conference*, Malaga, Spain, 2016Malaga, Spain.
- [43] J. Wu, A. Abbas-Turki, and A. El Moudni, “Cooperative driving: an ant colony system for autonomous intersection management,” *Applied Intelligence*, vol. 37, no. 2, pp. 207–222, 2012.
- [44] D. R. Fulkerson, “Flow networks and combinatorial operations research,” *The American Mathematical Monthly*, vol. 73, no. 2, pp. 115–138, 1966.
- [45] M. Klimm, M. E. Pfetsch, R. Raber, and M. Skutella, “On the robustness of potential-based flow networks,” *Mathematical Programming*, pp. 1–38, 2022.
- [46] H. Zou, K. Cao, and C. Jiang, “Spatio-temporal visual analysis for urban traffic characters based on video surveillance camera data,” *ISPRS International Journal of Geo-Information*, vol. 10, no. 3, p. 177, 2021.