

Research Article

Resource Scheduling for Postdisaster Management in IoT Environment

J. Sathish Kumar  and **Mukesh A. Zaveri** 

Computer Engineering Department, SVNIT, Surat, India

Correspondence should be addressed to J. Sathish Kumar; sathish613@gmail.com

Received 27 November 2018; Revised 4 February 2019; Accepted 18 February 2019; Published 20 March 2019

Guest Editor: Maurizio Casoni

Copyright © 2019 J. Sathish Kumar and Mukesh A. Zaveri. 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.

For postdisaster management, rescue and recovery operations are very critical. It is desired that the rescue and recovery operation should be handled through efficient resource management to minimize the postdisaster effects in terms of human loss and other types of damage. Resource management requires addressing various challenging issues like scheduling and monitoring of the resources which need real time information of various activities or events occurring anytime, anywhere, and anyplace. To satisfy such requirements, Internet of Things, an advanced upcoming technology, can be utilized for resource monitoring and scheduling. In this context, we propose resource scheduling algorithm for the postdisaster management. As mentioned above various tasks of rescue and recovery operation should be carried out with different priority and there should not be deadlock while availing the resources. In our approach, we estimate the waiting time using queuing theory for the availability of the resources for different activities that are to be performed at various locations. The simulation results of the proposed method are analyzed using different standard parameters like resource utilization and the waiting time for different activities. The proposed method is further visualized through real time annotation of resources and activities represented with the help of Google maps using android based application on the smartphone. The proposed algorithm is further compared in terms of computational complexity and fairness analysis for the effective utilization of the available resources.

1. Introduction

Natural disasters such as earthquakes, floods, hurricanes, volcanic eruptions, and tsunamis strike at various places on the earth every year with a great loss of human beings and the economy. United Nations International Strategy for Disaster Reduction (UNISDR) survey [1] reported that the number of disasters across the earth has been 346 during the year 2015. As a consequence of these disasters, 22,773 people lost their lives, 98.6 million people were affected, and an economic loss of USD 66.5 billion had occurred. In India alone, 38 million people were affected with different types of disasters and USD 29 million of economic damage occurred [2] during 2011-2015. From this information, it is clear that handling the disaster situation is very critical. So, there is a need for an efficient disaster management.

Among all the various phases in the disaster [3, 4], the response and recovery phases are crucial for postdisaster management. For quick response and recovery, there is a

need for different resources and these resources must be distributed in such a way that various activities can be performed efficiently. This leads to a need for proper resource utilization as the number of resources is limited. In this context, scheduling of resources is required for completion of different activities or tasks for postdisaster management. Scheduling of resources in the response phase is crucial because it has to perform various activities like the evacuation of the people and shifting them to a safe location on an urgent basis. To handle disaster-related rescue operations, the real time need of various activities should be identified properly. For instance, in the context of disaster, various activities like gathering the information across the disaster places for establishing communication network are essential and critical. If a communication network is IP enabled and if it is possible to connect various resources using this IP enabled network, then it is possible to have information about the need of resources and whereabouts of various resources in real time. This information can be utilized for the distribution

and scheduling of resources for different activities. Such IP enabled network, having resources with sensor devices and RFID tag, is possible to realize using the Internet of Things (IoT). IoT technology is very influential that works based on Internet [5–7] and can help in a great extent to meet the requirements of the postdisaster management. In this paper, we propose algorithm for such IoT based networks. As mentioned above with the help of IoT, it is possible to access the information anytime, anyplace, and anywhere which helps in decision making.

Depending on the type of activity and its impact, various activities should be executed in a timely and proper manner. In such scenarios, it is necessary that each activity should be assigned the priority. Based on the priority of various activities, they should be given more importance while scheduling different resources. For example, the hospitalization of the injured people is having more priority compared to the reallocation of properties or handling the economic loss related activities. Further, the resources are scheduled so that all activities should be addressed with minimum waiting time for completing the respective activities. For estimating the waiting time of the various activities, queuing theory is explored. Oftentimes, disaster activities might take place in an overlapped manner with respect to time. This situation leads to a set of requests simultaneously for many resources by different activities at various locations which results in deadlock or race conditions. If rescue operations are not carried out at right time, the situation may deteriorate. From the above discussion, it is clear that efficient resource scheduling for various activities is necessary and challenging. In this context, we propose the priority based resource scheduling algorithm in this paper.

The main contribution and novelty of our paper is that we propose a resource scheduling algorithm in IoT environment for postdisaster management, by considering the priorities of the activities and to avoid deadlock conditions, with optimum number of resource centers by estimating minimum waiting time using queuing theory. For evaluating our proposed method, a case study is conducted in Surat city of India, by emulating the affected area, for considering different activities to be completed with the given number of resources. The algorithm output is visualized on android based application using Google map. The resource monitoring at various activities can be carried out effectively. The proposed queuing model is evaluated with respect to resource utilization and waiting time with the different number of resource centers by varying the arrival rate and service rate of activities. The proposed approaches are compared with the standard set of parameters like fairness in the allocation of resources for utility and computational complexity.

The rest of the paper is organized as follows. Related work is presented in Section 2. Section 3 introduces the problem formulation. The system model based on queuing theory for evaluating the waiting time for resource scheduling is detailed in Section 4. The proposed resource scheduling algorithm is described in Section 5 and in Section 6, the simulation results are presented. Section 7 concludes the paper with references at the end.

2. Related Work

In this section, the review of overall postdisaster management resource scheduling and allocation of various approaches is explored. A study conducted by Fraunhofer Gesellschaft on emergency and disaster management systems [8, 9] concludes that, besides the information and communication management, the other areas which are critical in disaster management task are viz., visualization, decision support, optimization and simulation, geographic information systems (GIS), simulation, and training. Two important activities in disaster situations are logistics support and evacuation. Various approaches related to evacuation have been reported in the literature [10, 11]. The approaches related to the logistics have been explored in [10, 12–15]. A generic approach to handle the disaster management has been proposed in [16, 17] using greedy and game theoretic based models. Regarding the resource allocation and scheduling during disaster situations, various approaches have been proposed in [18–21]

Optimal resource allocation in emergency management was carried out with the help of agent-based system using cost allocation by creating events with a severity of the disasters at various levels and verified with Pareto optimality test in [18]. This approach is not suitable for handling resource allocation for concurrent emergency events, in which the occurrence time of different events could overlap with each other. Resource allocation model with two stages with vulnerability analysis has been proposed in [20]. The approach proposed in [20] works at rescue team level. The higher authorities may not be able to coordinate the tasks efficiently due to lack of the information. For postdisaster management, centralized and distributed coordination is required. But for disaster management, as the whole task is very complex, the top level decision must be incorporated for designing the response strategy which was lacking in [20]. Dynamic scheduling based optimized resource allocation has been proposed in [21] using the genetic algorithm. The approach in [21] may not be suitable for varying the situations and for real time resource mapping as various critical factors are not considered and further, it lacks in terms of integration of various modules required for disaster management. In [22], scheduling using priorities transportation agency in hurricane recovery has been proposed.

For disasters like earthquake, the resource allocation approach has been optimized for emergency response using dynamic combinatorial optimization model [23]. The approach in [23], specifically proposed to handle earthquake situation, might not be suitable for other kinds of disasters. Distribution in humanitarian relief using mixed integer linear programming approach has been proposed in [24] using different parameters like transportation cost and a number of days needed to rescue. The approach reported in [24] may increase computational complexity if an optimized solution is required for a given situation. Allocation and scheduling of rescue units in natural disaster management has been proposed in [25] using Monte-Carlo based heuristic approach. As stated in [25], the performance could be improved by coordinating and collaborating over the autonomous agents.

For flood-related disasters, an approach for logistics preparation in an uncertain environment has been proposed in [26]. Similarly, for urgent relief in disaster situation based on demand, for logistics distribution has been proposed in [27].

In the literature, for handling emergency response scenarios the queuing theory has been used to address various issues. In [28], a detailed survey has been carried out for disaster management operations. Queuing theory has been explored for health care management services in [29–34]. Queuing theory based model [29] has been proposed for the patients requiring an immediate emergency cardiac response. Likewise, in [30], using queuing theory method, approach has been proposed for enhancing hospital emergency service. These approaches lack the integration of various processes and the automation of the complete system. In [31], a technique has been suggested to analyze the completion time for responding in case of accidents and emergency situations, which can be utilized by the respective departments of the UK government. In [31], it is stated that as the arrival rate of patients increases, then the system becomes unstable in estimating the targets of such accidents. Similarly, the allocation of various resources in ICU (intensive care unit) of the hospitals during busy times of emergency has been explored in [32]. As per the study carried out in [32], the allocation of resources in disaster situations yielded uncertain results due to randomness and unpredictability. Kristin Fitzgerald et al. [33] proposed a queue-based Monte-Carlo analysis in disaster situations for decision making to execute the rescue operations by the hospital management as fast as possible. In [33], the waiting time for the patients to get the nursing resource as demanded was evaluated. In the approach proposed in [33], worst case scenarios have not been considered. For disaster circumstances, the solution to handle worst case scenarios is very much crucial and needed at the highest priority. Likewise, in [34], the queuing theory has been exploited to make the analysis for managing the hospital systems during emergency times efficiently. The waiting time for the patients for availing the treatment was not discussed in [34].

The approaches that have been discussed so far give a glimpse of the resource scheduling and allocation in disaster management situations, but to understand the current scenarios, we explored various approaches that have been recently proposed [35–40]. The resource allocation approach for postdisaster management in the context of IoT has been proposed by considering the priorities of both various activities and resources using stable matching approach [35]. Likewise, resource allocation for maximum utilization of the resources using maximum bipartite graph theory has been proposed in the context of IoT for postdisaster management [36]. In both approaches of [35, 36], the estimating of the waiting time for allocation and number of resource centers to be utilized has not been detailed. The approach to optimize the decisions related to logistics in a collaborative way for the flood disaster situation has been proposed [37]. But, it is reported in that the proposed model is not able to manage the differences between the various organizations involved for rescue and their utilization to an essential required place [37]. Likewise, an approach for postdisaster humanitarian

logistics in the context of proving the medical assistance teams has been proposed [38]. However, few limitations are reported that the classification of various types of relief medical supplies is not carried out and while dispatching the logistics, heterogeneous vehicle routing problem is not addressed. Further, resource scheduling approach is proposed in the IoT environment for postdisaster management [39]. But, the drawback in this approach [39] is that the estimating of the waiting time for various tasks and resources is not addressed. An approach for allocation of the resources during simultaneous disasters has been proposed [40]. They performed the risk analysis with resource constraints for meeting the given targets and estimated the financial aspects with different budgets and further the political effects have been considered.

From the above literature, it is clear that there is a scope for an efficient resource scheduling algorithm for postdisaster management. In this context, we propose an algorithm which considers the different critical parameters for allocation and scheduling of the resources in an effective manner. The proposed scheduling algorithm along with the problem formulation is described in the following sections.

3. Problem Formulation

For postdisaster management, the rescue and recovery operations are very crucial and to carry out these operations resources are required. Having limited number of resources, for various task accomplishment, resource scheduling must be done in an efficient manner. Scheduling of the resources can be performed using the number of parameters. In this paper, we consider two parameters, namely, the priority of the activities and the expected completion time of an activity. The completion time of any activity depends upon the resource availability. In such case, the resource availability is very crucial and must be handled efficiently. In this context, it is very important to find out time duration after which the resource may be available for a given task or activity. This time duration can be considered as a waiting time. So, it is necessary to estimate a waiting time for resources and it indirectly provides a measure for the completion time of an activity. In our proposed algorithm, we estimate the waiting time for the availability of the resources that is required for a specific activity. Generally, various resources are dispatched from different locations in the affected region. These locations are considered as resource centers. At the beginning of any rescue operation, various resources are distributed from different resource centers. It is very important to decide the number of resource centers that should be established to cover the maximum affected area. Similarly, the disaster affected area should be divided into subareas or zones for effective rescue operations. Each zone may have at least one resource center. In this context, the required optimal number of resource centers needs to be estimated for the postdisaster management activities.

In brief, from the above discussion, the waiting time for the availability of a resource and the number of resource centers required to carry out rescue operation are significant

and this operation should be performed in a very efficient and effective manner. In this view, we exploit a very standard approach, queuing theory, for estimating a waiting time and the number of resources and using these estimated parameters, the resource scheduling algorithm is introduced. It is also required that the various activities should be completed in a particular order, which leads to a situation where the given activities should be completed with a particular priority which may be preassigned. The waiting time evaluation and estimating the optimal number of resource centers and the priority assigned to different activities are the key parameters for effective and efficient postdisaster management. Further, the scheduling of the resources must be done in a fair manner avoiding the race conditions or starving among the activities. For scheduling, a large number of methods have been reported in the literature which is detailed in the related work section. Among various scheduling algorithms surveyed, we explore Banker's algorithm [41, 42] for our proposed scheduling algorithm. Our proposed algorithm is enhanced in such a way that it should meet the required conditions like the priority of a given task or activity. In the literature, Banker's algorithm has been used for process scheduling in the operating system. Earlier, we have proposed a scheduling approach [43] for postdisaster management using Banker's algorithm. In this paper, our earlier proposed algorithm is enhanced by incorporating the parameters, namely, the priority of the task and waiting time for the availability of the resources for resource scheduling.

From the above discussion, for our proposed system, we address mainly three issues which are very important and crucial for rescue and response operations as a part of postdisaster management. These three issues are the number of resource centers to be utilized and resource scheduling of the respective activities by considering the priorities and scheduling them without deadlock conditions. For this purpose, the disaster affected region may be divided into regions or zones for effective activity management. All the available resources are to be visualized at the top level as a group. In this context, the problem formulation is represented in the form of a graph model with various resources and their interconnections in the form of activities.

Let us say that there are l activities and m resources in the system. These activities say a_1, a_2, \dots, a_l can be denoted by a_i , where $(1 \leq i \leq l)$. Likewise, the resource of j type is termed as r_j , where $(1 \leq j \leq m)$. In the system, it is assumed that the number of similar types of resources is available, i.e., if a particular resource r_j is available with k number of quantities, then we define it as k instances of r_j . The number of instances of k varies from one resource to other resources. This information is available at the time of planning. For example, three ambulances are present in a system considered as three instances of resource type ambulance.

In a graph G , each node is labeled, either, as an activity node or a resource node. The node is represented as an activity node, a_i , and resource node, r_j . These particular activity nodes and resource nodes are interconnected through the edges. In real practice, the rescue and response operation have to perform the number of tasks. Each task consists of different activities which results in dependencies among the

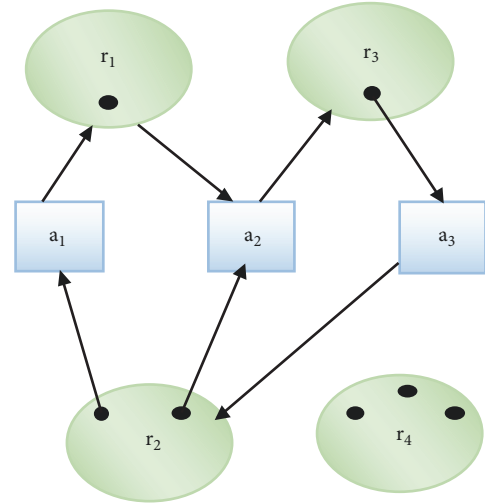


FIGURE 1: Illustration of improper scheduling.

activities for completing a task. In our proposed system, the dependency among the activities is handled at the scheduling level. In this view, the typical graph using activities and resources looks like as shown in Figure 1. To make graph representation simpler, the different instances of resources are also represented as graph nodes and these instances are represented by black dots within the resource node itself. It is assumed in this example that there is no activity dependency. So, the activity nodes are not interconnected. Figure 1 reflects the same, where r_1 and r_3 have one instance each, whereas r_2 and r_4 have two and three instances, respectively.

As pointed out earlier, various tasks consist of different activities and each activity requires a set of resources. It may result in deadlock or race condition among the different activities. As mentioned, the dependency among the activities is handled by the scheduling algorithm but it is necessary to map these dependencies in the graph of activities and resources nodes. The scheduling algorithm is devised in such a way that it should help in resolving the deadlock or race conditions. In such cases, it may happen that different activities require the same set of resources for completing the task. As each activity needs a set of resources that need to be scheduled, the allocation of resources to the activities and scheduling them must be handled simultaneously to avoid the deadlock or race situations. It is important to note that, in our proposed systems, we take care of this critical requirement through the graph which helps the scheduling algorithm for fair allocation. In this view, the resource dependencies among the activities are represented through directional edges in the graph. The arrow from an activity node to the resource node indicates that a particular activity needs that resource. The arrow from resource node instance to activity node indicates that a particular instance of the resource is allocated to that activity. Indirectly, this representation of activities and resources allows one to find a cycle or deadlock situations or race conditions at a given instance of time. All these directional edges represent the demand in the allocation at a given instance of time and that demand varies with the time.

The major contribution of this work is to represent the demand and the allocation of the resources and dependencies among the activities, through the directional graph. This makes our graph dynamic which actually represents a dynamic network of resources. This graph representation again helps the management authority at a given instance of time for monitoring the resources and activities and hence, the task at any level can be performed with ease. As shown in Figure 1, activity a_1 needs resource r_1 which is currently allocated to activity a_2 . Similarly, two instances of r_2 are allocated to activities a_1 and a_2 , respectively. But the same instance resource r_2 is required by the activity a_3 . As there is no additional instance of r_2 available, it cannot be allotted to activity a_3 . This requires an effective and efficient scheduling. The demand and allocation are represented as a directed graph $G = (V, E)$, where V is a set of vertices that indicates the activities and resources and $E \subseteq \{[a_i, r_j] \mid a_i, r_j \in V, a_i \neq r_j\}$ defines the potential allocation edges. Any resource scheduling algorithm should perform the scheduling of resources among the activities in a fair manner. There should not be any deadlock situation or race condition which results in the starvation of resources which holds the complete system in a standstill condition. So, it is important to schedule the different activities and allocate the resources based on some additional information or the number of parameters. These parameters should be chosen in such a way that it should lead to a very effective and efficient resource allocation and scheduling for postdisaster management activities.

These parameters are utilization of the optimum number of resource centers, evaluating the waiting time for availability of the resources for completing a particular task, i.e., waiting time for a particular resource, and resource scheduling discussed earlier with the priority of different activities. In this view, the system must have a methodology to incorporate these three parameters in resource allocation and scheduling. The postdisaster management requires real time dynamic decision making. For supporting the dynamic decision making using these three parameters, it is necessary to evaluate them in real time, i.e., based on the demand of the allocations. Considering this fact, the best solution that one has is based on queuing theory. In this context, our system is modeled using queuing theory for evaluating these parameters. In the following section, we describe the queuing based modeling using the above parameters.

4. Queuing Theory Based System Model

In the literature, the queuing theory has been very well exploited for the problem of demand and supply or producer and consumer based applications. For demand and supply based applications, the system has the number of service providers and the number of service users. In queuing theory, various parameters like arrival rate of service user and service rate, length of the requests in the queue, and waiting time of the requests in the queue are very critical and modeled based on statistical distribution. In the context of postdisaster management, the task and activities are the services to the

user or consumer, whereas resources act as a service provider or producer. In this view, the queuing theory based modeling helps in designing the robust system which not only provides an optimal solution but allows one to complete the required task in an effective and efficient manner. The performance of the queuing based system depends on the number of available resources, i.e., service centers or servers in the system and the length of the requests in the queue. Consequently, these two parameters decide the waiting time for the availability of a resource for a particular activity. As discussed earlier, these resources are distributed and located at various places in the disaster affected area.

The queuing system is represented using Kendall's notation [28]. The Kendall's notation consists of four variables that define the system parameters. The standard Kendall's notation is given as $M/M/C/K$; then these parameters are defined as arrival rate/service rate/number of servers/capacity of each queue. For instance, if the arrival rate is represented as λ /hour and the service rate as μ /hour, with 1 server of the capacity of 10 items in the queue being hold, then the Kendall's notation can be given as $\lambda/\mu/1/10$. In the context of the disaster management, the various activities are requesting the resources to accomplish the given job. Hence, the resources are acting as servers and activities are waiting in the queue to get served by the resources. These resources are available in various resource centers. Therefore, the queuing theory in our system is defined as arrival rate of requests by various activities/service rate of the resources/number of resource centers/number of activities waiting in the queue.

Generally, the number of requests is not known in advance. In such a case, the queuing system can be modeled using an infinite number of requests. Similarly, for serving these requests, the resources available in the system are considered to be finite. The arrival rate and the service rate are modeled based on different probabilistic distributions. For instance as shown in the above notation, $M/M/C/K$ represents the modeling of arrival rate as Poisson distribution M as the first parameter, followed by the second parameter for service rate which is modeled as exponential distribution M . C is the number of resources centers which are available in the system and K is the number of requests arriving to the system by various activities. The reason for using Poisson distribution for arrival rate is because it involves memoryless waiting time until the arrival of the next request in nonoverlapping time intervals that are probabilistically independent.

For the postdisaster management, we explore the queuing theory and model the system accordingly. Postdisaster management consists of various tasks to be accomplished. These tasks are decomposed into different activities and each activity requires a set of resources for its completion. For our system, an arrival rate of the requests is nothing but the arrival of activities for different resources. The request is originated by a particular activity for a given task. Similarly, a service rate of different requests is nothing but a particular resource utilized by a particular activity of a given task. As discussed in the problem formulation, the service rate, i.e., the time for which a resource is utilized by an activity affects or decides when the same resource will be available to another

activity, which is considered as a waiting time for the next activity to acquire that same resource. In a queuing theory, this waiting time is also modeled similarly to the service rate; i.e., if the service rate is modeled using exponential distribution, then waiting time also follows an exponential distribution. The main aim of our work is to distribute the resources to different activities with minimum waiting time. In this context, we model our system using queuing theory to estimate the optimal number of resource centers required to minimize the overall waiting time for all the activities in the system. These resource centers can be utilized for efficient resource scheduling. The queuing system based modeling is detailed as below.

In this paper, for modeling the system using queuing theory, we define four parameters. These parameters are related to (i) the number of requests that arrive for different resources at resource centers, (ii) the service rate or execution time for completing the requests, (iii) the length of the queue for holding these requests for different resources at resource centers, and (iv) the capacity of the systems which is a measure of how many requests can be handled by the system at a given time, i.e., the number of the requests in a queue and the number of requests in a service or execution and overall time from entry to exit of a request through the system. Here, the system consists of the number of resource centers and each resource center has a number of resources. In this

context, these four parameters are, namely, categorized as (i) the length of the activities in the queue to hold the number of requests that are coming to the system, (ii) the waiting time for the activities in the queue as the resources and the resource centers are busy with serving the other activities, (iii) the length of the total activities which is the capacity of the system holding the number of requests in the queue as well as in the service, and (iv) the waiting time of the total number of the activities which reflects the duration from entry to exit through the system. Determining the optimal number of resource centers and the waiting time for the requests using the queuing model based on the above four parameters is detailed as below. Now, the total number of activities in the system is defined in

$$\begin{aligned} \text{Total number of activities in the system} \\ = \frac{\text{Activities arrival rate/Resources service rate}}{1 - \text{Activities arrival rate/Resources service rate}} \end{aligned} \quad (1)$$

Since the length of the system, total number of activities in the system, is known, the number of activities that are waiting in the queue, waiting time of the all the activities, and waiting time in the queue can be determined using Little's equation [28] as shown in

$$\begin{aligned} \text{Total number of activities in the system} &= \text{Total number of activities in the queue} \\ &+ \text{Expected number of activities are being served by the resources} \end{aligned} \quad (2)$$

$$\text{Waiting time in the queue} = \frac{\text{Total number of activities in the queue}}{\text{Activities arrival rate}} \quad (3)$$

$$\text{Waiting time of all activities} = \frac{\text{Total number of activities}}{\text{Activities arrival rate}} \quad (4)$$

The example of the single resource center is explained with an example in Appendix A.

The above model is derived using a single resource center. The waiting time for the availability of resources for different activities which are served by a single resource helps further to design the system with multiple resource centers. In real practice, there are many areas in the affected region where the different tasks are to be performed and only one resource

center is not sufficient. Using the above model which is derived for a single resource center can be extended for multiple resource centers so that the waiting time for the availability of different resources for different activities of various tasks can be minimized and the rescue and recovery operation can be executed efficiently. In this context, the queuing system is modeled with number of resource centers that can be determined by the following equation

$$\begin{aligned} \text{Total Number of Activities in the queue} \\ = \frac{\text{Activities arrival rate/Resources service rate}^{\text{Number of Resource Centers}+1} P_0}{(\text{Number of Resource Centers} - 1)! (\text{Number of Resource Centers} - \text{Activities arrival rate/Resources service rate})^2} \end{aligned} \quad (5)$$

where, P_0 is the probability that there is no queue. Equation (5) represents the relationship between the length of the requests in the queue and the number of resource centers.

The optimal number of resource centers can be evaluated by verifying by (5) =1, i.e., no waiting time. It means that as soon as the requests come, they will be processed which depicts an

```

(1) procedure Scheduling( $Available_{Curr}$ ,
     $Allocation_{Curr}$ ,  $Demand_{Curr}$ ,  $Requested$ ,  $Priority_p$ )
(2)   Schedule the activities by sorting from high to low priority
(3)   while for all activities w.r.t  $Priority_p$  do
(4)     if  $Requested \geq Demand_{Curr}$  then
(5)       Go for next priority activity
(6)     else if  $Requested \geq Available_{Curr}$  then
(7)       Go for next priority activity
(8)     else
(9)        $Available_{Next} = Available_{Curr} - Requested$ 
(10)       $Allocation_{Next} = Allocation_{Curr} + Requested$ 
(11)       $Demand_{Next} = Demand_{Curr} - Requested$ 
(12)    end if
(13)     $Available_{Curr} = Available_{Next}$ 
(14)  end while
(15)  Return schedule and execute the activities.
(16) end procedure

```

ALGORITHM 1: Resource Scheduling Algorithm (Without Waiting Time Estimation).

ideal situation. The example of the single resource center is explained with an example in Appendix B.

5. Proposed Resource Scheduling Algorithm

In the postdisaster management, rescue and response operations are very crucial. The rescue operations should be carried out effectively in such a way that various services should be restored and need of the affected people should be addressed. For this purpose, acquiring the data and availability of this data anytime, anyplace, and anywhere can be achieved through IoT.

Further, different IoT devices, sensing various events like fire, temperature, and gas leakage, detectors of human life, etc., are deployed at the affected area after the disaster. These devices may be useful for tracking the further stroke of such disasters because IoT devices help in managing and tracking the different resources in the affected area. In such scenarios, it is necessary to group all these devices so that the information acquired through devices can be channelized or utilized for rescue work. In this context, it is important to localize these devices which are deployed through drone like equipment [44]. It is assumed that localization of all these devices is carried out using [45]. Similarly, for effective data acquisition and information extraction, all these devices must be clustered. For clustering, our earlier proposed approaches [46–49] are used.

In a real application, the number of resource centers is limited and constrained by the number of other issues. So, it is difficult to achieve the state of the system, where there is no waiting time. It means that the number of requests is more than the number of servers/resources. In this scenario, it is required to maintain the queue and schedule the activities in such a way that there are no starvation and race conditions. In the literature, various approaches have been reported for resource scheduling as discussed in Section 2. Using our system model described above, we present the scheduling algorithm as shown in

Algorithms 1 and 2. Among various algorithms reported in the literature, Banker's algorithm [41, 42] has been proposed for resource scheduling for avoiding the deadlock situations. The limitation of this Banker's algorithm is that it requires in-advance knowledge of the number of resources required which is not possible in real applications. But at the same time, the advantage of Banker's algorithm is that it avoids the deadlock conditions and schedules the resources in a safe sequence. For a particular disaster management, to initiate the rescue and recovery operations, the work in the affected area is generally started with the available number of resources. In this context, Banker's algorithm is more suitable for the resource scheduling in a disaster affected area. Earlier, Banker's algorithm has been explored for resource scheduling in [43]. In a real environment, the various tasks which are to be performed in a disaster affected area have different priorities so that there will be minimum human and financial loss. So it is important to incorporate the priority of the task in allocating the resources or scheduling the resources. Second, the number of resource centers will be limited and the number of requests will be more. In such a situation, for completing the different activities of a task, it has to wait for the availability of the resources from the resource centers which introduces the waiting time for the requests to be executed. This discussion shows that the priority of the activity and waiting time for the availability of the resource must be considered in scheduling the different resources. This motivates us to propose modified Banker's algorithm which incorporates the priority of the tasks and the waiting time for the availability of a particular resource for completing a particular activity of a task. Queuing theory based modeling described in the previous section provides the queue length for the resource and waiting time of the system. As discussed in the problem formulation, let us assume that there are l activities and m resources in the system. The activities are denoted by a_1, a_2, \dots, a_l are given as a_i , where $(1 \leq i \leq l)$ and, likewise, the r_j , where $(1 \leq j \leq m)$ with k instances that vary from one resource to another. The total number

```

(1) procedure Waiting Time Estimation(Arrival rate, Service rate and Number of Resource Centers)
(2)   while for  $i$  in all activities  $a_i$  do
(3)     Calculate the values as per equation (1) to (5). ▷ Estimate waiting time for activities
(4)     Determine the optimal number of resource centers using (5)
(5)     Verify the waiting time is  $\approx 0$  using multiple resource centers.
(6)   end while
(7)   Call Algorithm 1;
(8) end procedure

```

ALGORITHM 2: Resource Scheduling Algorithm (With Waiting Time Estimation using Queuing Theory).

of resources of type j required by an activity i over the time to complete the activity is represented as $Demand_{ij}$. Similarly, the number of resources of type j allocated to an activity a_i is indicated by $Allocation_{ij}$. This representation leads to the two-dimensional matrices demand and allocation represents the current state of the requests for a particular resource and number of resources in service. As the requests arrive for the different resources from the different activities, $Demand_{ij}$ varies. As the resources are allocated to a particular activity, $Allocation_{ij}$ is updated. As discussed earlier for a resource type r_j , the number of currently available instances is stored in $Available_j$. The particular type of the resource may be required by different activities of different tasks. So it is not possible to allocate all the available resources for a particular activity at a given moment of time. The $Allocated_{ij}$ is the number of resources of type j currently allocated to activity a_i against its $Demand_{ij}$. So, additional resources of type j required by an activity a_i for its completion are stored in $Requested_{ij}$. With these definitions, now we are able to verify the resources that are available and can be allocated immediately. The first criterion checks the requested number of resources that should be less than the available number of resources; i.e., the additional required resources are less than the total available resources which can be represented as (6). The proposed algorithm fulfills the essential conditions of the activities and verifies the availability of the requested resources using the given equation

$$\sum_{i=1}^l \sum_{j=1}^m Requested_{ij} \leq \sum_{j=1}^m Available_j \quad (6)$$

for l number of activities, i.e., a_i where $i=1$ to l , and j number of different types of resources r_j where j varies from 1 to m . This criterion presents that a particular activity needs a different type of the resources for its completion; i.e., the equation indicates that the total number of additional resources required by the different activities is less than the total number of available different resources. Similarly, the second criterion states that the demand is more than the available resources; then the request is placed in the queue. This can be as shown in

$$\sum_{i=1}^m r_i \leq \sum_{i=1}^l \sum_{j=1}^m Demand_{ij} \quad (7)$$

for l number of activities, i.e., a_i where $i=1$ to l , and j number of different types of resources r_j where j varies from 1 to m .

With these formulations, the algorithm is represented with a simpler resource scheduling algorithm based on Banker's method with priority assigned to each activity. The flowchart of the proposed algorithm is shown in Figure 2 and corresponding pseudocode is depicted in *Resource Scheduling Algorithm 1*. In the proposed algorithm, the subscript *Curr* and *Next* are used with the different notations to represent the current value of that variable and the value available for the next iteration, respectively. For example, $Available_{Next} = Available_{Curr} - Requested$ represents the number of resources of a particular type available for the next iterations which can be evaluated by subtracting the additionally required number of resources by a particular activity from the currently available number of resources of a particular type.

In this context, two variants of proposed algorithm are presented for resource scheduling. The first variant considers only the priority assigned to the activity without considering evaluation of waiting time for availability of a resource. Second, we represent scheduling algorithm where the priority assigned to different activities and waiting time for availability of a particular resource is also considered which makes our resource scheduling algorithm dynamic and elegant. Algorithm 1 presents resource scheduling algorithm considering the priority of the task or activity on hand and Algorithm 2 estimates the waiting time for the resource availability for completing different activities using queuing theory and calls Algorithm 1. To evaluate the waiting time in the queue, parameters like arrival rate, service rate, and number of resource centers are important and essential. In our proposed approach, by keeping number of resource centers constant, we observe the waiting time by increasing the service rate for different arrival rates. The same information is presented as a flowchart in Figure 3 and corresponding pseudocode is shown in Algorithm 2. Upon the observation with different arrival rates, the optimum number of resource centers can be utilized. After the resource centers are finalized, the scheduling algorithm is called for the resource allocation for various activities.

The time complexity of the proposed *Resource Scheduling Algorithm 1* is $O(a^2r)$ with utmost a possible activities and r resources, whereas the brute force approach time complexity is $O(a!/(a-r)!)$. For a number of activities and r number of resources, the enumeration in FCFS approach [50] is aP_r , which is equal to $(a!/(a-r)!)$. Therefore, the proposed

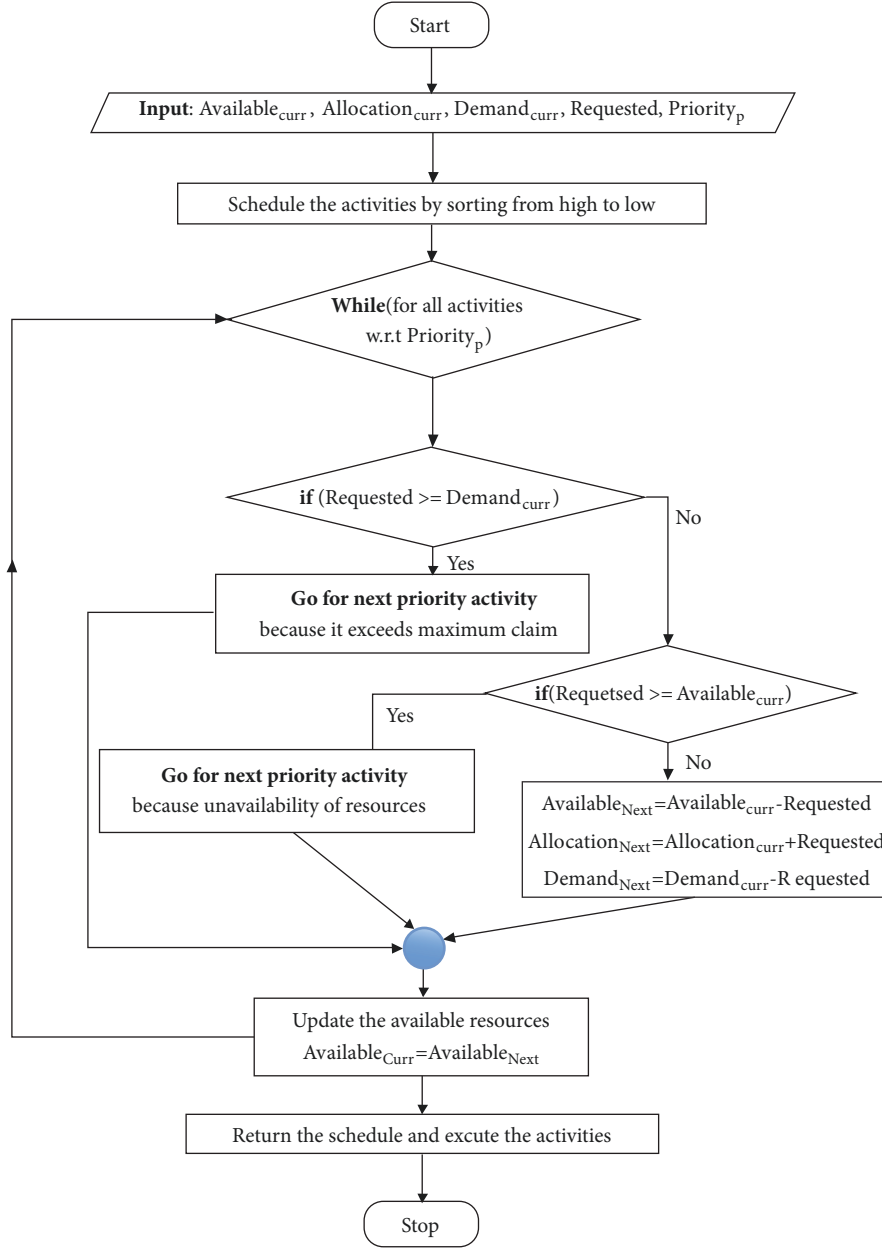


FIGURE 2: Flowchart of the Proposed Resource Scheduling Algorithm 1.

resource scheduling algorithm using Banker's strategy gives optimal results compared with the brute force approach.

6. Simulation Results

In this section, the simulation result is presented in three subsections. The first subsection presents the evaluation of queuing based proposed system whereas second subsection describes the result of evaluation of proposed priority based scheduling algorithm. The simulation results are carried out for evaluating the proposed queuing theory based system. The simulations are performed using different scenarios for evaluating the waiting time and determining the optimum

number of resource centers given queuing system parameters like arrival rate and service rate. In each scenario, the aim is to minimize waiting time and complete different activities of various tasks required for rescue and recovery operation for postdisaster management. Efficient scheduling of various resources through different resource centers should be allocated in an optimal way. A case study is presented having different emergency locations in the area where different activities are to be performed using Google map. For this case study, the number of resource centers with different types of resources is considered. These resources are required by various activities of different tasks that are to be performed for rescue and recovery operations. The proposed scheduling

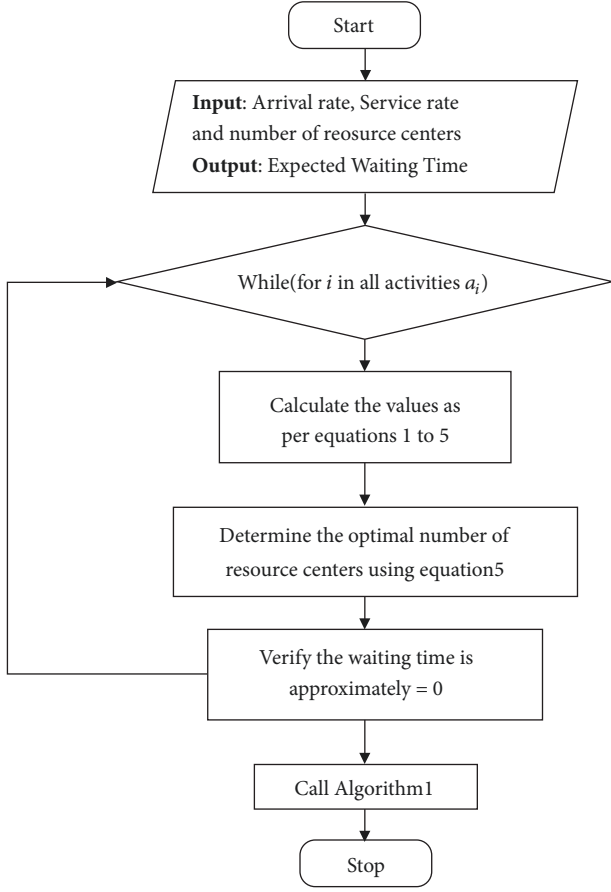


FIGURE 3: Flowchart of the Proposed Resource Scheduling Algorithm 2.

algorithm outcomes are depicted in the visual form using the developed application on the android platform which can be deployed for postdisaster management. For the actual implementation purpose, we designed and created our own android application using API:19 Android 4.4 (KitKat) [51]. In the third subsection, the comparative analysis is presented. The proposed algorithm is evaluated and compared with existing approaches reported in the literature. The performance analysis is carried out using different standard parameters, algorithm computational complexity time, and fairness in resource allocation and scheduling.

6.1. Evaluation of the Queuing Based System. For demonstrating the robustness of our queuing theory based system, the proposed model is evaluated in terms of resource utilization and waiting time with the different number of resource centers and activities. For ease of understanding, we used few notations to represent the various parameters, such as arrival rate of the activities = λ , service rate of the resources = μ , number of resource centers = C , and resource utilization = $\rho = \lambda/\mu$. With these notations, the simulations are performed in different scenarios. Initially, the resource utilization is evaluated by keeping the number of resource centers constant. For the given number of resource centers,

the simulations are performed by varying the arrival rate of requests originating from different emergency locations for various activities. The resource utilization is plotted with respect to increasing rate of service for different requests that arrive at resource centers. It is shown in Figure 4.

It is clearly observed from Figure 4 that the resource utilization decreases as the service rate increases for different arrival rates. As the number of resource centers increases, the resource utilization decreases which can be visualized from Figure 4. As shown in Figure 4, as the resource center increases from 1, 2, 3, and 5, the maximum resource utilization decreases from 1.0, 0.5, 0.35, and 0.2. Similarly, another parameter that is evaluated is the waiting time. For the different arrival rates, the waiting time for availability of the resource for different activities is plotted with respect to increasing rate of service which is shown in Figure 5.

As the service rate increases, the waiting time decreases for different arrival rates. Further, the waiting time is decreasing as the number of resource centers is increasing. As shown in Figure 5, as the resource centers increase from 1, 2, and 3 the maximum waiting time decreases. The corresponding values for resource centers 1, 2, and 3 are 1.0 hrs, 0.025 hrs, and 0.003 hrs, respectively. Also, it is observed that if the number of resource centers is increased more than 3, the waiting time of the various activities is almost zero as the service rate increases. Based on the above experimental results, it is observed that as the number of resource centers increases, the waiting time for availability of the resource is decreasing. But at the same time, the cost of the capital amount for setting up the resource centers increases as the C increases. Hence, there should be a tradeoff in utilizing the optimum number of resource centers such that the waiting time for the activities can be handled effectively.

6.2. Evaluation of Proposed Resource Scheduling Method. For evaluating the proposed modified Banker algorithm for resource scheduling, the real scenario is emulated as a case study. The Google map is used to show different emergency locations in the area chosen for emulation. These are the areas of local city, Surat, of the state Gujarat in India. The complete visualization of resource scheduling is executed through the android application developed for smartphone devices. The idea is on-site application can be used in case of disaster and various resource centers can be monitored in real time. All devices which are part of resources may be IP enabled and may serve as IoT devices. These IoT devices now can be monitored through our developed application anywhere and anytime. This is the advantage of having IoT enabled environment for postdisaster management. All the tasks and various activities of these tasks along with their utilization of various resources can be monitored using a simple mobile device. The network can be accessed using a simple plugged router as an access point in the affected area. The proposed modified Banker algorithm is implemented on android platform for online monitoring of the complete network. The resource scheduling scenario is tested and analyzed. The various screenshots of execution of resource scheduling method are presented using Google map [52].

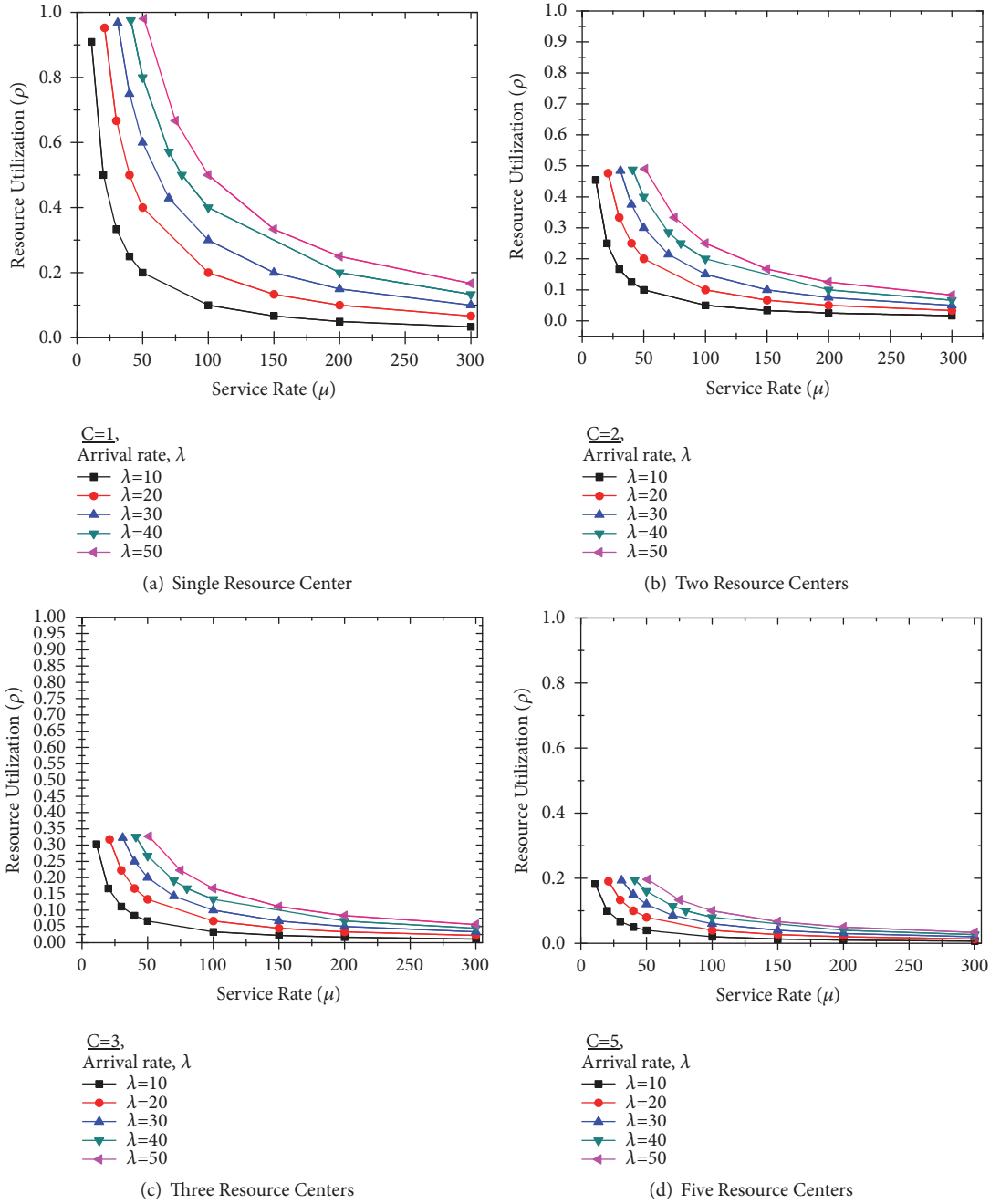


FIGURE 4: Resource Utilization of Various Resource Centers.

The resource scheduling is evaluated by considering different areas of Surat city as emergency locations. Let us assume that the resources are requested by different tasks at emergency locations that are local areas, namely, VR Mall, Big-bazaar, SVNIT, Adajan, and Athwa Gate. The resources are requested from the different resource centers located at different areas of the city, namely, Adajan, Majura gate, and Katargam. The locations of emergency places and resource centers are visualized using Google map. These locations are shown in Figure 6. Figure 6(a) represents the particular

center places are listed. Figure 6(b) displays the locations of resource centers using Google map. Figure 6(c) shows the emergency locations in the Surat city.

For compact visualization on a small screen of the mobile device, the symbols are assigned as R1, R2, and R3 for resource centers. Similarly, the emergency places are assigned names A_0, A_1, A_2, A_3 , and A_4 . These A_0, A_1, A_2, A_3 , and A_4 may represent the different tasks or activities at respective places or locations. This representation is shown in Figure 7. Various activities of different tasks to be performed at emergency locations require a different number of resources from the

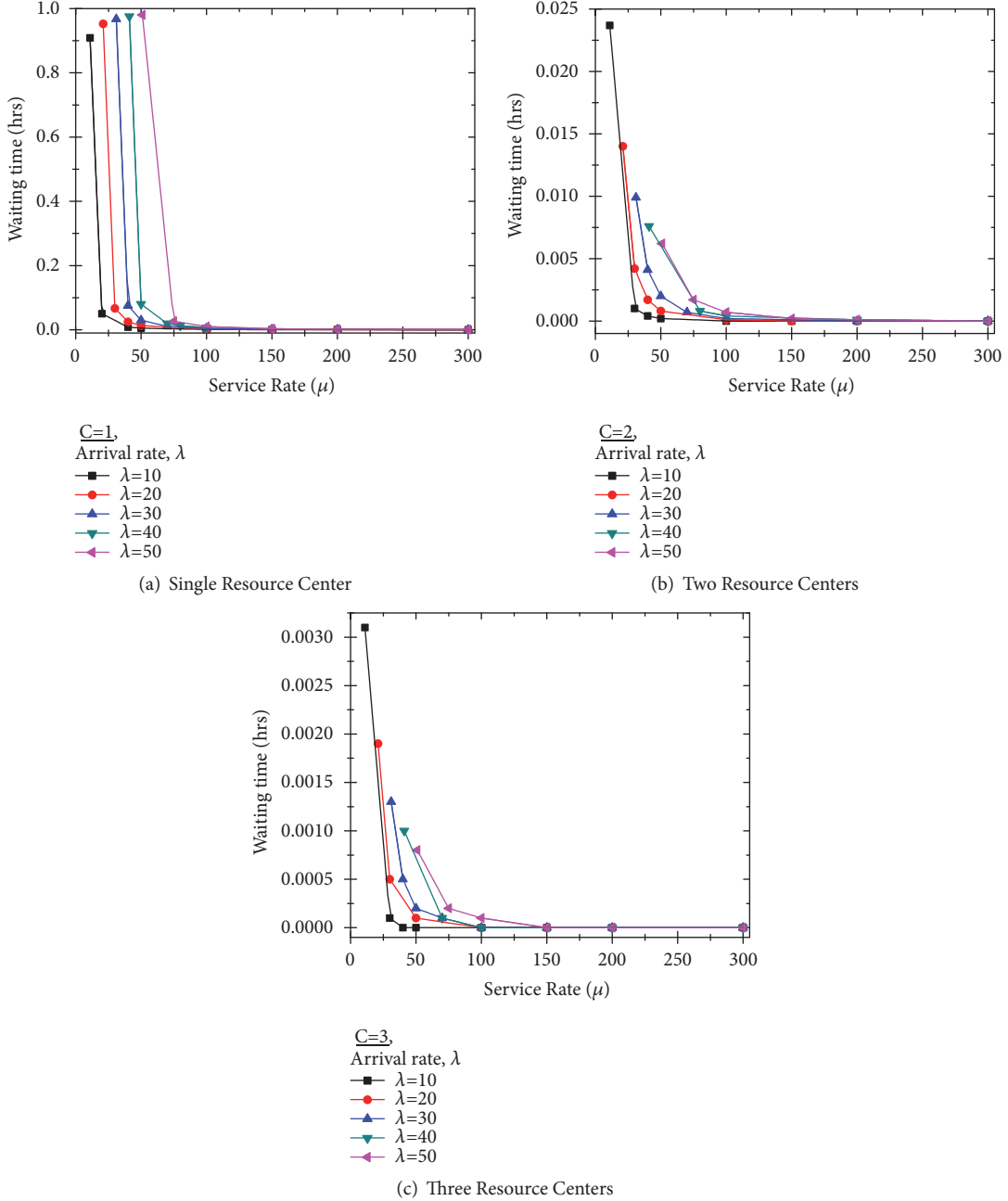


FIGURE 5: Waiting Time of Various Activities.

respective resource center which is represented as two-dimensional entries in the matrix form. As discussed in the proposed algorithm section, these two parameters are incorporated for better resource allocation and scheduling, i.e., the priority of the activity and the waiting time for the availability of the resource for an activity. To minimize the waiting time for the availability of the resources for different activities, we utilized queuing based modeling which is incorporated in the proposed resource scheduling algorithm and the simulation is carried out.

Let us assume that there are 3 types of resources, say, r_1 , r_2 , and r_3 with 10, 5, and 7 instances of each resource,

respectively. These resources r_1 , r_2 , and r_3 are available at resource centers R1, R2, and R3, respectively. Let the resources be allocated to 5 activities, say, A_0 , A_1 , A_2 , A_3 , and A_4 . As shown in Figure 7(a), at place A_0 , activities have resources $(r_1, r_2, r_3) = (0, 1, 0)$ initially allocated from resource centers R1, R2, and R3. Similarly, at location A_1 , activities have been allocated resources $(r_1, r_2, r_3) = (2, 0, 0)$ initially. At other places, A_2 has $(3, 0, 2)$, A_3 has $(2, 1, 1)$, and A_4 has $(0, 0, 2)$ resources initially allocated. The additional requirement of resource by each activity is shown in Figure 7(b); i.e., at place A_0 , activities have demand of resources $(r_1, r_2, r_3) = (7, 5, 3)$ from resource centers R1, R2, and R3, respectively. Similarly, at

2:42 PM 4G VoLTE 72%

Resource Scheduling

List of Resource Centers are:

R1 - Guru Ram Pavan Bhumi, Adajan Gam, Adajan, Surat, Gujarat 395009

R2 - NX-1, Ghod Dod Rd, Sanskar Nagar, Athwa, Surat, Gujarat 395017

R3 - Katargam, Surat, Gujarat 395004

List of Activities and Emergency Locations are:

A0 - SVNIT

A1 - Athwa Gate

A2 - Adajan

A3 - VR Mall

A4 - Big baazar

NEXT PAGE >>

(a) Details of resource centers and emergency locations



(b) Resource Centers on Google map



(c) Emergency locations on Google map

FIGURE 6: Google Map Representation.

11:56 AM 4G VoLTE 53%

☒ All EXAMPLE

Allocated Resources

| Activity - Location | Allocation | R1 | R2 | R3 |
|---|------------|----|----|----|
| <input checked="" type="checkbox"/> A0 - SVNIT | | 0 | 1 | 0 |
| <input checked="" type="checkbox"/> A1 - Athwa Gate | | 2 | 0 | 0 |
| <input checked="" type="checkbox"/> A2 - Adajan | | 3 | 0 | 2 |
| <input checked="" type="checkbox"/> A3 - VR Mall | | 2 | 1 | 1 |
| <input checked="" type="checkbox"/> A4 - Big baazar | | 0 | 0 | 2 |

(a) Resource allocation values

11:56 AM 4G VoLTE 53%

☒ All EXAMPLE

Maximum Resources Required for Activity Completion

| Activity - Location | Max | R1 | R2 | R3 |
|---|-----|----|----|----|
| <input checked="" type="checkbox"/> A0 - SVNIT | | 7 | 5 | 3 |
| <input checked="" type="checkbox"/> A1 - Athwa Gate | | 3 | 2 | 2 |
| <input checked="" type="checkbox"/> A2 - Adajan | | 9 | 0 | 2 |
| <input checked="" type="checkbox"/> A3 - VR Mall | | 2 | 2 | 2 |
| <input checked="" type="checkbox"/> A4 - Big baazar | | 4 | 3 | 3 |

(b) Maximum resource required

11:56 AM 4G VoLTE 53%

☒ All EXAMPLE

Available Resources at Resource Centers

| Available | R1 | R2 | R3 |
|-----------|----|----|----|
| | 3 | 3 | 2 |

SUBMIT

(c) Available resource values

FIGURE 7: Input details of location and values for all tasks.

other emergency locations, A_1 needs (3 2 2), A_2 needs (9 0 2), A_3 needs (2 2 2), and A_4 needs (4 3 3) additional number of resources from A, B, and C resource centers. After initial allocation, the available numbers of resources at resource centers A, B, and C are 3, 3, and 2, respectively, which is shown in Figure 7(c). The resource centers and emergency locations which are listed in Figure 6(a) are represented using Google map and are displayed in Figure 8(a).

Now, the execution of our proposed resource scheduling algorithm results in resource scheduling of resources for different activities, in a safe sequence, avoiding deadlock

and race conditions which are shown in Figure 8(b). Using Banker's algorithm for resource scheduling, the execution of various activities scheduled at different emergency locations is represented by the sequence of numbers 1, 3, 4, 0, and 2 in Figure 8(b). This sequence number represents the emergency location places and without priorities. That is, the activities are scheduled at places A_1 , A_3 , A_4 , A_0 and A_2 in sequence. So activity at place A_1 is executed first followed by the activities at place A_3 . Then, activities at place A_4 are executed followed by activities at A_0 and A_2 . In terms of actual locations of the Surat city, the resource is allocated to Athwa Gate first,

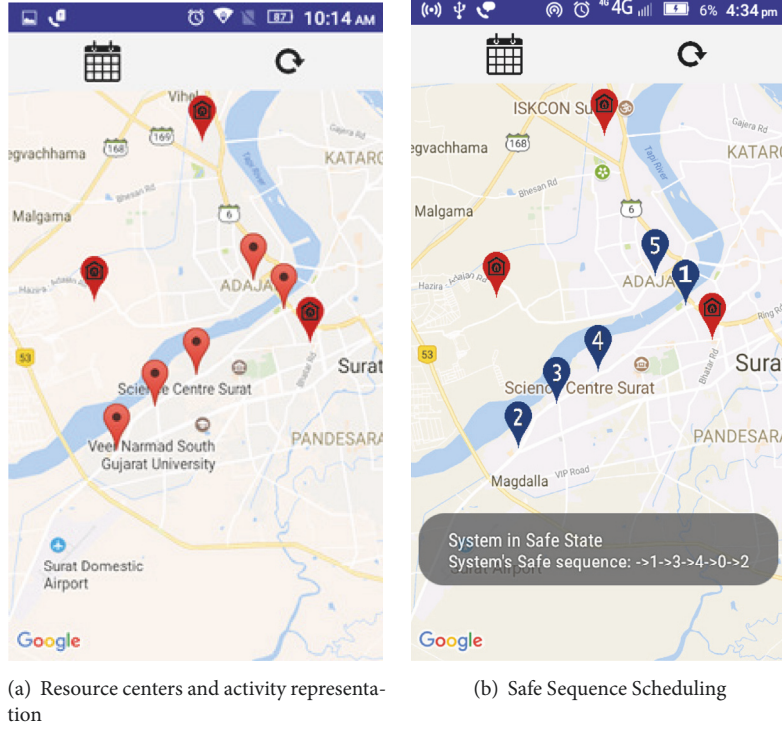


FIGURE 8: Scheduling of activities without priorities.

followed by VR Mall, Big-bazaar, SVNIT, and Adajan in sequence. After the allocations of resources at these places in sequence as discussed above the sequence of execution at different places is shown with a blue marker in Figure 8(b).

Afterward, the simulation result is presented where another parameter, i.e., the priority of the activity, is incorporated for resource scheduling. The visual representation assignment of the priority to different activities is shown in Figure 9(a). The activities at place A_0 are assigned lowest priority whereas the activities at place A_3 are assigned the highest priority compared to other places. The output of resource scheduling method, with two parameters, the waiting time for availability of the resource for different activities and the priority of the activity, is displayed in Figure 9(b).

The resource scheduling for different activities at different places using the proposed method results in a different sequence of places for executing different activities. The sequence is now, 3, 1, 2, 4, and 0, as shown in Figure 9(b). This sequence indicates that the scheduling of the activities at place A_3 should be executed first due to the higher priority, followed by activities at place A_1 . The activities at place A_2 are having higher priority than the activities at place A_3 . Because the requested number of resources is 9 from resource center A by activities at place A_2 it is not possible to allocate those many resources as the available number of resources is only 5. Due to nonavailability of the sufficient number of resources, the next priority activities at place A_1 are executed. Afterward, the activities at place A_2 , followed by the activities at place A_4 and the activities at place A_0 , are executed in sequence. Using this sequence, now,

the resources are allocated to VR Mall, followed by Athwa Gate, Adajan, Big-bazaar, and SVNIT places, respectively. The sequence of places for execution of different activities using our resource scheduling method is shown with a blue marker in Figure 9(b). This demonstrates that, using our priority based resource scheduling algorithm, it is possible to execute the different activities at different places without deadlock or any race conditions for availing different resources and it results in a safe sequence of activity execution across different places.

6.3. Comparative Analysis. For demonstrating the efficacy of our proposed algorithm, the performance analysis is carried out by comparing our approaches with different algorithms reported in the literature [10, 35, 36, 50]. The approaches proposed for resource allocation with first come first served (FCFS) strategy [50], greedy strategy [10], stable matching approach [35], and maximum bipartite graph approach [36] are considered for our comparison purpose. The performance is compared using the standard parameters [53], that is, fairness in resource allocation and execution time for completing the different activities.

The proposed approaches are devised in such a way that maximum utilization of the resources is achieved. For each activity, the resources were allocated with maximum fairness. Jain et al. [53] proposed a technique to measure the fairness by quantifying it. The fairness is given by the following equation

$$f(X) = \frac{[\sum_{i=1}^l a_i]^2}{n} \times \sum_{i=1}^l a_i^2 \quad (8)$$

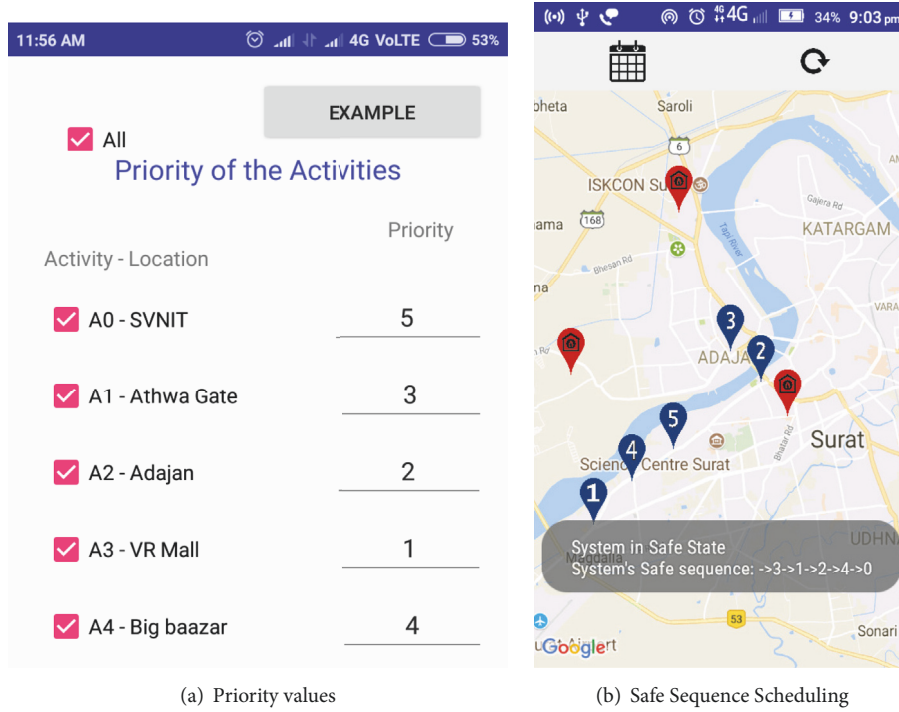


FIGURE 9: Scheduling of Activities with Priorities.

where $0 \leq f(X) \leq 1$ is fairness measure of resource allocation. $X = (a_1, a_2, \dots, a_l)$ represents the number of resources a_i allocated to activity i . There are l activities assumed. A large value of $f(X)$ represents fairer resource allocation from the system perspective. The corresponding result is shown in Figure 10. It is observed that the fairness of Algorithm 1 is better than the other approaches compared. Due to the additional parameter of priority, Algorithm 2 gives less fairness in the allocation of resources than Algorithm 1. From this figure it is clear that the fairness achieved using resource scheduling methods using priority of activities and waiting time for availability of the resource for different activities is much higher compared to that of other methods reported in the literature.

We also analyze the proposed algorithm in terms of computational complexity for critical time analysis and response. As discussed earlier, for the proposed *Resource Scheduling* Algorithm 1, the time complexity is $O(a^2r)$ with utmost a possible activities and r resources. But brute force approach takes $O(a!/(a-r)!)$. Also, the greedy approach proposed in [10] takes $O(r^2 a \log a)$ because of the sorting of the activities. The time complexity of stable matching approach is $O(ar)$ whereas maximum bipartite graph approach takes $O(pq)$ where $p = |a| + |r|$ and $q = |E|$, i.e., number of edges. The comparative results of computational complexity time analysis are shown in Figure 11, which helps us to determine the proposed algorithm results better than brute force approach [50] which is FCFS and greedy approach [10]. It is observed that stable matching algorithm takes less time followed by simple Banker's algorithm which takes less time among all the approaches compared. Because of the

complexity consideration for queuing theory, Algorithm 2 takes more time than Algorithm 1.

7. Conclusion

Resource scheduling and activity management are crucial during disaster circumstances. The functionality of IoT in the IP-network formation for communications in real time monitoring helps in effective tracking and utilization of the available resources. The empirical results show that using queuing theory helps to determine the optimal number of resource centers. Based on the usage of optimum resource centers, the proposed algorithm schedules the resources by considering the priority of the activities depending on the situational requirements. The proposed algorithm is further verified by taking a case study of Surat city and demonstrated using real time android application with the help of Google maps. The proposed approaches are evaluated in terms of fairness and computational complexity time analysis, which shows better results than the approaches reported in the literature. The proposed algorithm can be further extended as future work for dealing with the response time of each resource at respective resource center which, in turn, affects the performance of the overall system.

Appendix

A. Single Server Resource Center

To illustrate the single resource centers with an example, assume the arrival rate of the activity is 8 per hour and service

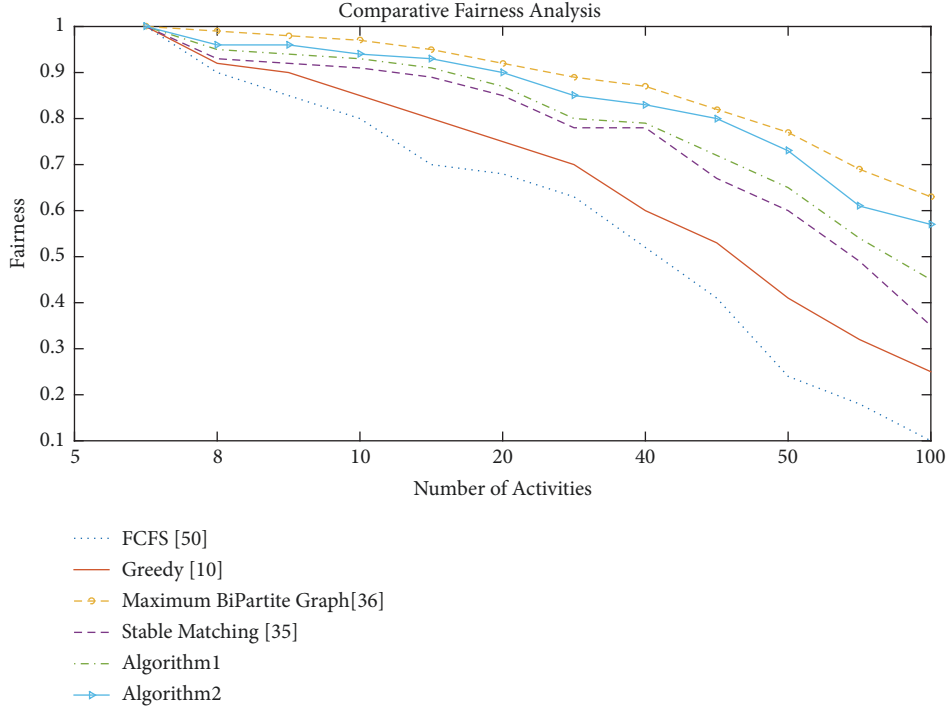


FIGURE 10: Fairness Comparative Analysis.

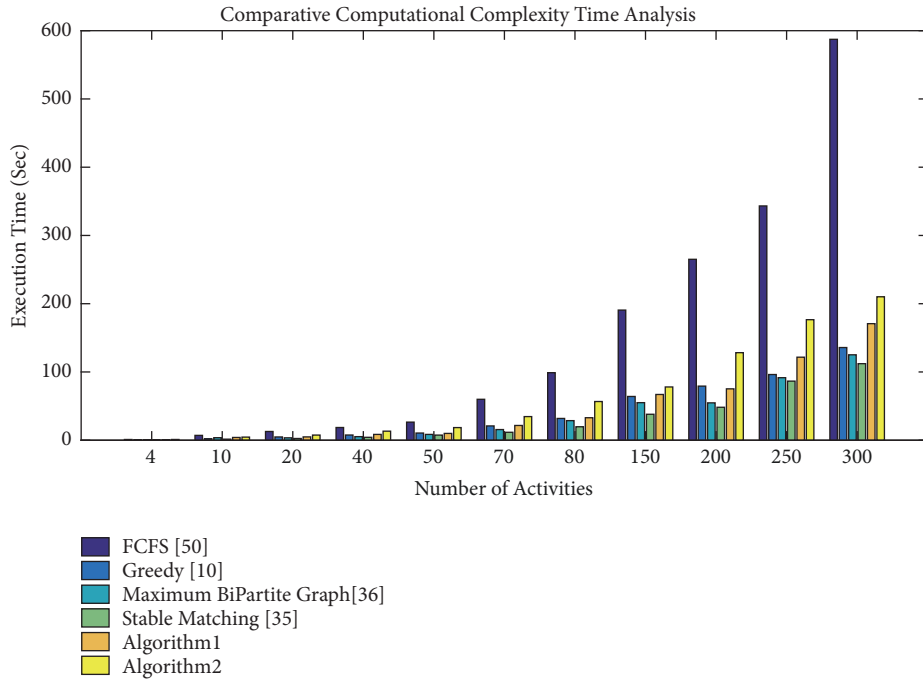


FIGURE 11: Algorithm Computational Complexity Time Analysis.

rate of resources is given as 9 per hour; then we have the following.

- (i) Now, $P_0 = 1 - \rho = 0.111$. 11.1% of the times there is no activity in the system.
- (ii) The probability that the resources are not available can be determined by $1 - P_0 = \rho = 0.88$. Therefore, 88.8% of the times the resources are busy.
- (iii) The probability that there is no queue is determined by $P_0(\text{no_queue}) = P_0 + P_1 = 0.11 + \rho P_0 = 0.111(1 +$

0.88) = 0.2098. So, 21% of the times there will be no queue.

- (iv) The probability of 10 activities in the system, P_{10} (in the system), $P_{10} = \rho^{10}P_0 = 0.88^{10} * 0.11 = 0.0341$. So, 3.4% times, out of 10 activities in the system, 1 is getting served and 9 are waiting for the service.
- (v) The probability of at least 2 or more activities in the system is given by $P(n \geq 2) = P_2 + P_3 + \dots + P_{\infty} = 1 - P_0 + P_1 = 1 - 0.2098 = 0.7902$. So, 79% of the times there will be 2 or more activities in the system.
- (vi) The waiting time of the system is 1 hour.

B. Multiple Server Resource Center

To illustrate the multiple resource centers with an example, assume the arrival rate of the activity is 10 per hour and service rate of resources is given as 6 per hour and the number of resource centers is 2; then, activities arrival rate/ (Numner of resource centers * service rate of resources) = $10/12 = 0.833 < 1$.

- (i) Now, $P_0 = 0.0909$. So, 9.1% of the times there is no activity in the system.
- (ii) The probability that both resources are free such that an activity is assigned is given by $P(\text{Activity_gets_a_resource}) = P_0 + P_1 = 0.2424$. So, 24.24% of the times the activity gets the resources.
- (iii) The probability that there is no queue is determined by $P_0(\text{no_queue}) = P_0 + P_1 + P_2 = 0.11 + \rho P_0 + \rho^2 P_0/2 = 0.389$. So, 39% of the times there will be no queue.
- (iv) Waiting time = 0.6 hr. The waiting time is 36 minutes.

Data Availability

None of the included data is available online.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

This work is supported by the Ministry of Electronics and Information Technology (MeiTY), funded by Government of India under the Grant No. 13(4)/2016-CC&BT.

References

- [1] M. Wahlstrom and D. Guha-Sapir, *The Human Cost of Weather-Related Disasters 1995-2015*, UNISDR, Geneva, Switzerland, 2015, https://www.unisdr.org/files/46796_cop21weatherdisaster-report2015.pdf.
- [2] "Data collection Survey for Disaster Prevention in India," Japan, 2015, <http://open.jicareport.jica.go.jp/pdf/12245155.pdf>.
- [3] M. Muaafa, A. L. Concho, and J. E. Ramirez-Marquez, "Emergency Resource Allocation for Disaster Response: An Evolutionary Approach," 2014.
- [4] L. Pearce, "Disaster management and community planning, and public participation: how to achieve sustainable hazard mitigation," *Natural Hazards*, vol. 28, pp. 211-228, 2003.
- [5] G. M. Lee, N. Crespi, J. K. Choi, and M. Boussard, "Internet of things," in *Evolution of Telecommunication Services*, pp. 257-282, Springer, 2013.
- [6] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): a vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645-1660, 2013.
- [7] L. Yang, S. H. Yang, and L. Plotnick, "How the internet of things technology enhances emergency response operations," *Technological Forecasting & Social Change*, vol. 80, no. 9, pp. 1854-1867, 2013.
- [8] A. Meissner, T. Luckenbach, T. Risse, T. Kirste, and H. Kirchner, "Design challenges for an integrated disaster management communication and information system," in *Proceedings of the First IEEE Workshop on Disaster Recovery Networks (DIREN '02)*, vol. 24, pp. 1-7, 2002.
- [9] G. Studie, "Market analyse katastrophen-und not fall management systeme," Kaiserslautern, 2002.
- [10] L. Özdamar and W. Yi, "Greedy neighborhood search for disaster relief and evacuation logistics," *IEEE Intelligent Systems*, vol. 23, no. 1, pp. 14-23, 2008.
- [11] T. J. Cova and R. L. Church, "Modelling community evacuation vulnerability using GIS," *International Journal of Geographical Information Science*, vol. 11, no. 8, pp. 763-784, 1997.
- [12] A. M. Caunhye, X. Nie, and S. Pokharel, "Optimization models in emergency logistics: a literature review," *Socio-Economic Planning Sciences*, vol. 46, no. 1, pp. 4-13, 2012.
- [13] H. O. Mete and Z. B. Zabinsky, "Stochastic optimization of medical supply location and distribution in disaster management," *International Journal of Production Economics*, vol. 126, no. 1, pp. 76-84, 2010.
- [14] G. Barbarosolu, L. Özdamar, and A. Çevik, "An interactive approach for hierarchical analysis of helicopter logistics in disaster relief operations," *European Journal of Operational Research*, vol. 140, no. 1, pp. 118-133, 2002.
- [15] E. T. Erdemir, R. Batta, P. A. Rogerson, A. Blatt, and M. Flanagan, "Joint ground and air emergency medical services coverage models: a greedy heuristic solution approach," *European Journal of Operational Research*, vol. 207, no. 2, pp. 736-749, 2010.
- [16] E. Rolland, R. A. Patterson, K. Ward, and B. Dodin, "Decision support for disaster management," *Operations Management Research*, vol. 3, no. 1-2, pp. 68-79, 2010.
- [17] A. C. Chapman, R. A. Micillo, R. Kota, and N. R. Jennings, "Decentralised dynamic task allocation: a practical game: theoretic approach," in *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems*, vol. 2, pp. 915-922, 2009.
- [18] J. Zhang, M. Zhang, F. Ren, and J. Liu, "An innovation approach for optimal resource allocation in emergency management," *IEEE Transactions on Computers*, 2016.
- [19] S. Babashetty, A. G. Kurian, and R. Vonna, "Multi-tenant disaster recovery management system and method for intelligently and optimally allocating computing resources between multiple subscribers," (Jul. 26 2016), uS Patent 9,400,718.

- [20] C. Zhang, X. Liu, Y. Jiang, B. Fan, and X. Song, "A two-stage resource allocation model for lifeline systems quick response with vulnerability analysis," *European Journal of Operational Research*, vol. 250, no. 3, pp. 855–864, 2016.
- [21] S. Chang, M. Ichikawa, H. Deguchi, and Y. Kanatani, "A general framework of resource allocation optimization and dynamic scheduling," *SICE Journal of Control, Measurement, and System Integration*, vol. 10, no. 2, pp. 77–84, 2017.
- [22] J. H. Lambert and C. E. Patterson, "Prioritization of schedule dependencies in hurricane recovery of transportation agency," *Journal of Infrastructure Systems*, vol. 8, no. 3, pp. 103–111, 2002.
- [23] F. Fiedrich, F. Gehbauer, and U. Rickers, "Optimized resource allocation for emergency response after earthquake disasters," *Safety Science*, vol. 35, no. 1, pp. 41–57, 2000.
- [24] B. Balcik, B. M. Beamon, and K. Smilowitz, "Last mile distribution in humanitarian relief," *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations*, vol. 12, no. 2, pp. 51–63, 2008.
- [25] F. Wex, G. Schryen, S. Feuerriegel, and D. Neumann, "Emergency response in natural disaster management: allocation and scheduling of rescue units," *European Journal of Operational Research*, vol. 235, no. 3, pp. 697–708, 2014.
- [26] M.-S. Chang, Y.-L. Tseng, and J.-W. Chen, "A scenario planning approach for the flood emergency logistics preparation problem under uncertainty," *Transportation Research Part E: Logistics and Transportation Review*, vol. 43, no. 6, pp. 737–754, 2007.
- [27] J. B. Sheu, "An emergency logistics distribution approach for quick response to urgent relief demand in disasters," *Transportation Research Part E: Logistics and Transportation Review*, vol. 43, no. 6, pp. 687–709, 2007.
- [28] N. Altay and W. G. I. Green III, "OR/MS research in disaster operations management," *European Journal of Operational Research*, vol. 175, no. 1, pp. 475–493, 2006.
- [29] A. M. De Bruin, A. C. Van Rossum, M. C. Visser, and G. M. Koole, "Modeling the emergency cardiac in-patient flow: an application of queueing theory," *Health Care Management Science*, vol. 10, no. 2, pp. 125–137, 2007.
- [30] L. Mayhew and D. Smith, "A multi-class queueing network analysis methodology for improving hospital emergency department performance," *Health Care Management Science*, vol. 11, no. 1, pp. 11–21, 2008.
- [31] J. K. Cochran and K. T. Roche, "Using queueing theory to analyse the governments 4-h completion time target in accident and emergency departments," *Computers & Operations Research*, vol. 36, no. 5, pp. 1497–1512, 2009.
- [32] M. L. McManus, M. C. Long, A. Cooper, and E. Litvak, "Queueing theory accurately models the need for critical care resources," *Anesthesiology*, vol. 100, no. 5, pp. 1271–1276, 2004.
- [33] K. Fitzgerald, L. Pelletier, and M. A. Reznick, "A queue-based monte carlo analysis to support decision making for implementation of an emergency department fast track," *Journal of Healthcare Engineering*, vol. 2017, Article ID 6536523, 8 pages, 2017.
- [34] G. R. Rodríguez Jáuregui, A. K. González Pérez, S. Hernández González, and M. D. Hernández Ripalda, "Analysis of the emergency service applying the queueing theory," *Contaduría y Administración*, vol. 62, no. 3, pp. 733–745, 2017.
- [35] J. S. Kumar, M. A. Zaveri, and M. Choksi, "Activity based resource allocation in IoT for disaster management," in *Proceedings of the International Conference on Future Internet Technologies and Trends*, pp. 215–224, Springer, 2017.
- [36] J. S. Kumar and M. A. Zaveri, "Graph-based resource allocation for disaster management in IoT environment," in *Proceedings of the Second International Conference on Advanced Wireless Information, Data, and Communication Technologies*, pp. 1–12, ACM, 2017.
- [37] O. Rodríguez-Espndola, P. Albores, and C. Brewster, "Disaster preparedness in humanitarian logistics: a collaborative approach for resource management in floods," *European Journal of Operational Research*, vol. 264, no. 3, pp. 978–993, 2018.
- [38] S. Wang, F. Liu, L. Lian, Y. Hong, and H. Chen, "Integrated post-disaster medical assistance team scheduling and relief supply distribution," *International Journal of Logistics Management*, 2018.
- [39] H. Gamitl, J. S. Kumar, and M. A. Zaveri, "Resource monitoring and scheduling for post disaster management using internet of things," in *Proceedings of the 3rd IEEE International Conference for Convergence in Technology*, pp. 1–5, 2018.
- [40] X. V. Doan and D. Shaw, "Resource allocation when planning for simultaneous disasters," *European Journal of Operational Research*, vol. 274, no. 2, pp. 687–709, 2019.
- [41] M. Lawley, S. Reveliotis, and P. Ferreira, "The application and evaluation of banker's algorithm for deadlock-free buffer space allocation in flexible manufacturing systems," *International Journal of Flexible Manufacturing Systems*, vol. 10, no. 1, pp. 73–100, 1998.
- [42] H. Madduri and R. Finkel, "Extension of the banker's algorithm for resource allocation in a distributed operating system," *Information Processing Letters*, vol. 19, no. 1, pp. 1–8, 1984.
- [43] J. S. Kumar, M. A. Zaveri, and M. Choksi, "Task based resource scheduling in IoT environment for disaster management," *Procedia Computer Science*, vol. 115, pp. 846–852, 2017.
- [44] M. A. Zaveri, J. S. Kumar, S. Kumar, and M. Choksi, "Collaborative data processing and resource optimization for post disaster management and surveillance using internet of things," Tech. Rep., Ministry of Electronics and Information Technology, 2016.
- [45] S. K. Pandey and M. A. Zaveri, "Localization for collaborative processing in the internet of things framework," in *Proceedings of the Second International Conference on IoT in Urban Space*, pp. 108–110, ACM, 2016.
- [46] J. S. Kumar and M. A. Zaveri, "Clustering for collaborative processing in IoT network," in *Proceedings of the Second International Conference on IoT in Urban Space*, pp. 95–97, ACM, 2016.
- [47] J. S. Kumar and M. A. Zaveri, "Hierarchical clustering for dynamic and heterogeneous Internet of Things," *Procedia Computer Science*, vol. 93, pp. 276–282, 2016.
- [48] J. S. Kumar and M. A. Zaveri, "Graph based clustering for two-tier architecture in Internet of Things," in *Proceedings of the IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pp. 229–233, 2016.
- [49] J. S. Kumar and M. A. Zaveri, "Clustering approaches for pragmatic two-layer IoT architecture," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 8739203, 16 pages, 2018.
- [50] L. K. Comfort, K. Ko, and A. Zagorecki, "Coordination in rapidly evolving disaster response systems: The role of information," *American Behavioral Scientist*, vol. 48, no. 3, pp. 295–313, 2004.
- [51] Android 4.4 APIs, 2018, <https://developer.android.com/about/versions/android-4.4>.

- [52] G. Svennerberg, *Beginning Google Maps API 3*, Apress, 2010.
- [53] D. C. R. Jain and W. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared systems," Technical Report DEC-TR-301, Digital Equipment Corporation, 1984.

