

Article

Flood Management in Aqala through an Agent-Based Solution and Crowdsourcing Services in an Enterprise Geospatial Information System

Hooshang Eivazy^{1,2,*}  and Mohammad Reza Malek¹ 

¹ Department of GIS, K.N. Toosi University of Technology, Tehran 19697 64499, Iran; mrmalek@kntu.ac.ir

² Department of GIS, Arak University of Technology, Arak 3818146763, Iran

* Correspondence: h.eivazy@arakut.ac.ir; Tel.: +98-9371-985-758

Received: 22 July 2019; Accepted: 26 August 2019; Published: 18 September 2019



Abstract: Propagating crowdsourcing services via a wireless network can be an appropriate solution to using the potential of crowds in crisis management processes. The present study aimed to deploy crowdsourcing services properly to spatial urgent requests. Composing such atomic services can conquer sophisticated crisis management. In addition, the conducted propagated services guide people through crisis fields and allow managers to use crowd potential appropriately. The use of such services requires a suitable automated allocation method, along with a proper approach to arranging the sequence of propagating services. The solution uses a mathematical framework in the context of a GIS (Geospatial Information System) in order to construct an allocation approach. Solution elements are set out in a multi-agent environment structure, which simulate disaster field objects. Agents which are dynamically linked to objects in a crisis field, interact with each other in a competitive environment, and the results in forming crowdsourcing services are used to guide crowds in the crisis field via the crowdsourcing services. The present solution was implemented through a proper data schema in a powerful geodatabase, along with various users with specialized interfaces. Finally, a solution and crowdsourcing service was tested in the context of a GIS in the 2019 Aqala flood disaster in Iran and other complement scenarios. The allocating performance and operation of other system elements were acceptable and reduced indicators, such as rescuer fatigue and delay time. Crowdsourcing service was positioned well in the solution and provided good performance among the elements of the Geospatial Information System.

Keywords: crowdsourcing services; enterprise GIS; multi-agent environment; complex system

1. Introduction

The elimination of natural hazard consequences is usually difficult due to their huge extent [1] in spatial and temporal scale, field recognition [2], and effective management [3]. Although, using some methods, such as data fusion [4] can solve some problems to some extent, issues, such as monitoring, resource management, services, financing [5], ensuring effectiveness of actions taken, reducing reacting time and delivering services, resilience [6], and creating widespread coverage with minimal resources are among the challenges that require a more robust solution. In this regard, crowdsourcing potentials can be regarded as a proper solution to cover the above-mentioned deficiencies [7].

Compared with volunteer resources, crowdsourcing involves more people with less expertise and interests. Although, it fails to include high standards of human resources for emergency services, it is able to fulfill other requirements, such as abundance and spatial distribution. However, managing many uncoordinated people, with few interests and potentials, seems difficult [8]. In order to overcome this problem, a new type of crowdsourcing service was introduced. These IT-based services were

designed to guide the early emergency system rescuers in the form of crowdsourcing resources through the disaster filed.

Spatial crowdsourcing services are proposed with three main segments: Information, activity, and control segments, which are connected to each other by a customized trigger mechanism. In other words, each service segment executes the next segment, as it is presented in Figure 1.

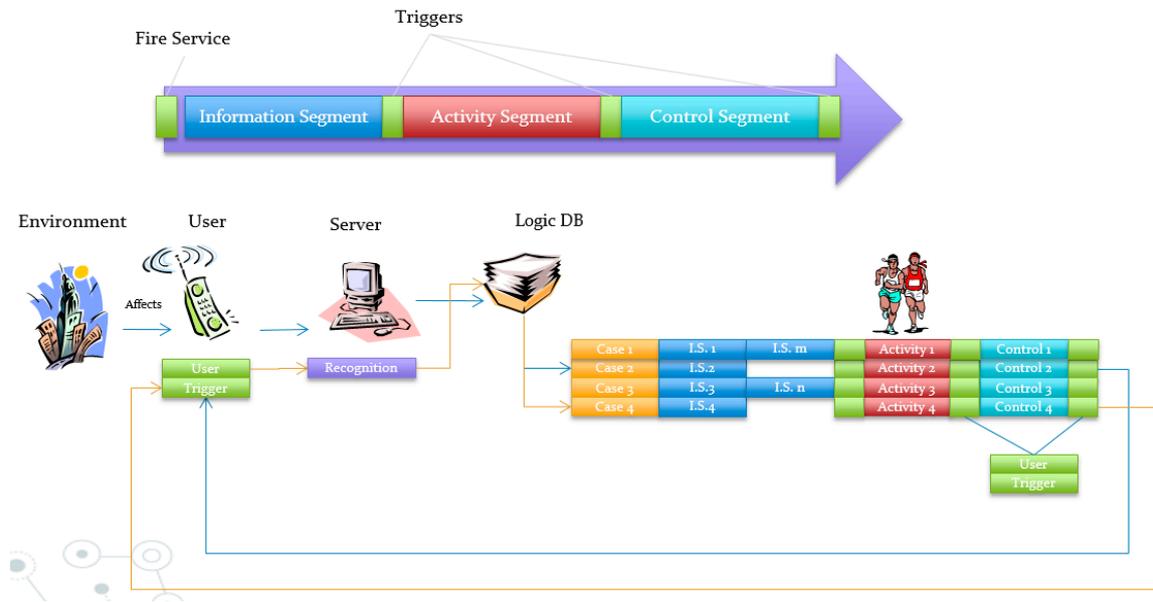


Figure 1. Internal structure of spatial crowdsourcing services and decomposition of crisis problems to atomic simple solutions using crowdsourcing services.

Proposed spatial crowdsourcing system in a basic and simple form would be able to digest the complicated and long-aiding mission of a single individual injured from a critical situation to the safe and stable status. Crowdsourcing services are classified into recognition, rubbing, first medical aid, carrying, capabilities, and orders rescuer the place to go. Then, it guides him what to do and finally takes a confirmation whether the mission has been completely done.

Follow the blue line in Figure 1. After the crisis, and its consequences, an injured person asks for help. After registering on the server, the request goes to a management database. Suppose that at this management database, we can just guide the execution of atomic processes through our services. On the other hand, the management database activates an initial service for each type of request (Figure 1, Case 2). As the service is delivered by a rescuer, the injury situation is improved from the initial state to a better one. In each service, the internal trigger mechanism activates the next part of the service. After the end of the service (Figure 1, Control 2), the external trigger mechanism is activated, and with the user confirmation, the information is sent back to the server (orange line) and from there, the next simple service is selected. By choosing the next simple service, the injury situation gets better. However, the main problem is related to managing issues. Managing many rescuers over disaster field, at different situations, with several skills in a central form, is as difficult as finding the best arrangement of crowdsourcing services related to the injured ones.

Spatial crowdsourcing service has been introduced recently, so there is poor research on its management, and, as a result, there cannot be expected to be many tools available for its administration [9–11]. However, if we consider them as planning units for a larger process, such as emergency rescue process, there are too many famous methods that manage human resources but do not meet our needs.

Methods based on genetic algorithm, fuzzy sets, best fit of human resources, simulations, and other similar ones in allocating and managing human resources, despite special privileges, are also problematic with regard to the problem we are facing. However, according to the prevailing

conditions, existing solutions do not meet our needs. The main reasons are apparent in areas, such as central or distributed mode of solutions, automation, or non-automatic processes, as well as differences in the inherent structure of solutions and problems. For example, the genetic algorithm method is very time-consuming and not practical for many users. The fuzzy sets about our problem, except to provide a general template and idea for the solution, solve no other problem. Likewise, in the problem of human resource management, much attention is paid to the cognitive issues of human resources and the nature of the missions, while in our case, there is no such cognitive principle at all.

In summary, the main problem can be considered in three main fields. The first is how to create a proper combination of crowdsourcing services to solve various problems relevant to individuals, the crisis, and disaster environment. Second, how can we choose the most suitable rescuer for each service? Third, to do such processes, we need a system. How will the structure of such a system look like?

In the present study, a solution and a system was proposed to complete the first two tasks. This solution finds the best rescuers to a given injured person. In addition, it conducts rescuing missions by assigning and composing proper services to suitable rescuers according to the injured situation. Finally, the system, which has been designed based on this algorithm, is briefly presented. Since the words “service” and “system” are frequently used in this paper, the terms “service” and “crowdsourcing service”, or the “system” and the “crowdsourcing system”, may be used interchangeably.

2. Materials and Methods

Since the topic of research, in this article, is crowdsourcing service management, we first mention the crowdsourcing, in brief, and then continue to explore a variety of HRM (Human Resource Management) methods, which are close to our subject.

A very academic definition describe crowdsourcing as an online participative activities. Such activities would be proposed by an organization, an individual, or even a non-profit organization, to a group of people in a heterogeneous condition, via a flexible open call platform. The proposed tasks would vary in complexity and modularity. On the other hand, crowds could bring their money, ideas, data [12], works, and experiences in receiving mutual benefits, such as developing proper software products for users [13]. Crowd also satisfy some of their demands, including professional abilities, and social or economic benefits [14]. Crowdsourcing raises action speed, spatial scale, and functionality extent, or reduce action cost [15]. Volunteer actions, which can be assumed as outsourcing significantly, differs from crowdsourcing because of type processes [16]. Outsourcing is comprised of top-down and bottom-up processes in contrast [17].

However, in spite of the technical issues and the background of the proposed new crowdsourcing services, in this paper, it is preferable to focus more on how to apply this service to crisis management and how to manage such services. On the other hand, given that these services manage human resources, different HRM methods will be considered.

In addition, although the use of crowdsourcing in crisis management [18] has been considered by some researchers, more of these researches look like social networks [19]. However, there are many differences between social services and crowdsourcing services, crowdsourcing has been usually used to gather the information needed to manage the crisis, similar to the research by Ogie et al., for mapping the crisis environment [20]. The use of natural language processing in mapping and cognition of crowdsourcing data has also been addressed [21]. In alike activities, some have examined the degree of reliability of provided spatial crowdsourcing data [22]. While, this kind of service has significant potential for crisis management.

Human resource management practices are critical for organizational success [23]. HRM can be viewed as the core processes of project management in many organizations. However, this is not a simple task and becomes more complex when the number of projects and professionals increases the range of required expertise [24].

Therefore, many distinct approaches have been proposed in the literature to support how to deal with individual software projects by using distinct techniques, such as simulation, genetic algorithms, and fuzzy theory [25–27].

The otero method [28] and BFR (Best Fit Resources) technique determine a fit between the candidates' set of skills and needs, which cannot actually meet all of the tasks required for any injured person during a rescuing mission.

In order to capture the flexibility of human actors, [29] a statistical approach is used to allow for flexible task management and role allocation. They had mind deadlines and escalation procedures, accessed rights and allowed for the temporary promotion and dynamic assignment of actors. However, this method on dynamic aspects requires a lot of data and makes the implementation process in large disaster fields too difficult.

Kang and Jung used the constraints of experts' interviews to allocate human resources to the required activities [27]. They implemented ASA (Accelerated simulated annealing) and COCOMO II estimation model. However, performing this technique becomes really difficult as the number of constraints increase.

Some new concepts declined resource allocation stiffly to production management [30]. Emergency situations are not emphasized. Although, the horizontal ability of human resource would benefit from production processes. Increasing the number of output production is considered the only important parameter.

Buffer management approaches are a class of project resource management, which is related to uncertainty and Critical Chain [31]. Although, the uncertainty forces a centralized approach, resource allocation should be organized in a decentralized way in a multi-agent approach. Agile scrum is a popular flexible project resource management method in IT development. Multidisciplinary teams with a high level of self-organization execute the work in sprints. Scrum-of-scrums is the MPM (Multi-Project Management) approach related to agile scrum [32].

Resource sharing policies form a different class are designed to deal with the inevitable conflict within a matrix organization between project, resource, and portfolio management [33]. Aritua et al. used feedback loops in an allocation process [34]. Also, during the last decade, heuristic Operational Research was regarded as a popular approach for project management, which dedicated among 20% of journals until 2000 [35].

Some believe that two main aspects should be considered in determining the appropriate approach. The first is related to whether a human or an algorithm makes the decision. Another important aspect is to see whether the decisions made are centralized or decentralized. Automated decisions are calculated based on simplified models of reality. These heuristic OR models improve the decision made by incorporating an increasing number of variables. The human decision approach is based on the idea that an algorithm can never incorporate all the situations, which occur in real-life and that humans are much more flexible to adjust themselves to unpredicted situations [36].

According to disaster field situations, a distributed approach is more feasible than centralized ones. In addition, autonomous algorithms are preferred than those made by human decisions. On one hand, numerous rescuers and injured people in disaster fields make the structure matrix too big and too difficult to solve in a centralized system according to their constraints. On the other hand, human-based algorithms do not overwhelm all environment variants.

Regarding practical aspects, WDP (Winner Determine Problem) has been treated by heuristic branch bound algorithm, as well as benchmarking on real world economic models. Although, both declined the solution in preference for centralized solutions.

Competitive Multi-Agent systems propose a dynamic assignment of human resources. Although, they have their own challenges. Finding incentives to diffuse focused resources, the protocols to complete the first price auction, and local games for Nash equilibrium are among the common problems in theoretical aspect [37]. At present, extensive discussions, such as the use of big data, micro data, modeling, and adaptation of the behavior of agent models, with real environments and

validation, are among the main challenges of spatial agent's researches [38]. Although, the use of sophisticated mathematical or probabilistic models simplifies spatial simulation, equally, it reduces the degree of credibility. On the other hand, accessing powerful processing resources to increase the speed of simulator systems is among the many important issues. In this regard, the proper use of simpler mathematical models can increase the reliability of responses, reduce the processing power required, and increase processing speed.

The fact is that this research intersects with four topics, including crisis management, crowdsourcing services, human resource management, and intelligent agents. Although, detailed researches have been carried out in each of these branches, the intersection of these fields is a new issue. In our research, the crowdsourcing service, communicates between these fields. For this reason, in this section, we compare the history of research in each branch separately with the capabilities offered in our service. In this way, we tried to simplify the difficult communication process between four different topics.

3. Solution

In order to conduct a crisis management process, our proposed crowdsourcing service, first, collects aiding requests from the environment. Then, the central system, using the same service, sends aiding requests to appropriate rescuers at the crisis field. But our proposed service has been designed in simple and atomic form. In other words, each service is assigned one rescuer and is designed to perform a simple process. In this way, in order to solve the complex problems of the crisis environment, it is necessary to send the appropriate combination of the crowdsourcing services to the appropriate rescuers. Therefore, our main problem is, firstly, the formation of an appropriate combination of simple atomic crowdsourcing services for solving a complex problem. Second, sending these services to the right rescuers.

To solve these problems, a multi-agent simulation environment was used with customized agents. A multi-agent solution involves a computer environment where several intelligent agents can interact with each other and solve problems that are difficult or impossible with a single part or a central solution. In our case, too, a large number of aiding requests must be solved by a large number of people, with different capabilities and specifications, in a vast area. An intelligent or smart agent refers to an autonomous entity which acts, directing its activity towards achieving goals, upon an environment using observation through sensors and consequent actuators [39].

In this solution, beyond the structure of spatial crowdsourcing services, the most important features of crisis environment was simulated as dynamic objects (Figure 2, part 1), except for rescuers and individuals who asks for requests. Then, a customized agent structure was designed for these two groups (Figure 2, part 2), while focusing on its behavior, especially the search mechanism (Figure 2, part 2). In fact, any rescuer or injured person in the crisis environment will have an object or an agent in the simulation environment.

These agents, then, interact with each other and simulated the environment (Figure 2, part 4) to find the best arrangement of crowdsourcing services and the most suitable rescuers. As the agents find the best rescuer and service, they send the result to the corresponding real rescuers in crisis field (Figure 2, part 6). Three main behaviors of the agent are heavily involved with the simulated environment (Figure 2, part 5). While the search mechanism finds the best rescuer, the guiding behavior takes the main duty of rescuer and navigation information, and learning behavior collects the results for future similar search. Finally, the rescuer decides whether to perform the proposed service (Figure 2, part 6).

Before proceeding to the details, we first review the main components of the solution. We want to tell each intelligent rescuer agent, among the help applicants (agents), who is more suitable for help? What can he do for him or her? To do this, all agents who need help, are ranked against each rescuer agent. The process of scoring and prioritizing, is the main decision mechanism for intelligent agents. This mechanism is based on the effectiveness cone (Section 3.3). To construct this cone, we need different parameters, which later represent the extraction of these parameters (Sections 3.1 and 3.2). The main

idea of this cone, and its constituent parameters, have been extracted from the issue of time-geography. Time-geography or time-space geography is an evolving trans-disciplinary perspective on spatial and temporal processes and events, such as social interaction, ecological interaction, social and environmental change, and biographies of individuals [40]. But why have we used this issue?

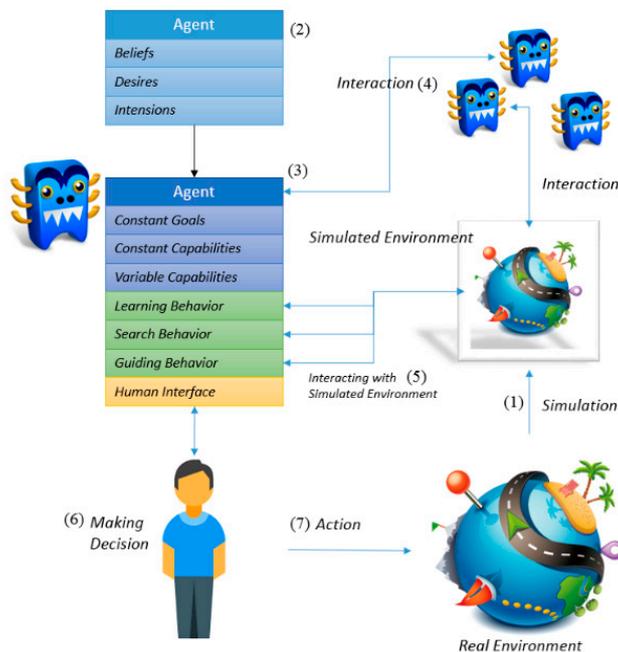


Figure 2. Structure of the proposed agent and its interactions.

One of the main challenges in designing intelligent agents is their mismatch with real-world objects behaviors. On the other hand, the main application of time-geography has been human behavioral studies, which have been successful in this regard [41]. For this reason, time-geography issue contains appropriate developed tools and techniques. It can be concluded that these tools are in good agreement with real human behaviors. Therefore, conceptual or software-based uses of these tools in intelligent agent design can, somewhat, increase the reliability of the intelligent agent operating design. Also, the use of this topic, along with the way we use mathematical tools, greatly reduces the sensitivity of the agent design to the calibration parameters. Now, we can look at the details of the solution.

In the following part, we will look more closely at the internal structure of the agents. Our agent consists of three clearly distinct sections; parameters (Figure 3, parts 2–4), behaviors (Figure 3, parts 14–16), and an interface (Figure 3, part 20). Notice that, in the proposed solution, the most important part of the agent is the decision mechanism, and in this mechanism, the most important part is the effectiveness cone (Section 3.3). The cone in this mechanism, for each rescuer smart agent, gives priority of aiding requests. Section 3 to Section 3.2 provide the required parameters for this cone.

Constant goals (Figure 3, part 2) ensure the agent can start functions and hear messages from the objects and agents in the simulated environment. Therefore, variable capabilities (Figure 3, part 4), such as aiding request positions and types, are updated. It will enter the next section along with the constant capabilities (Figure 3, part 3), such as rescuers' abilities and interests. Figure 3, parts 5 to 13 actually represent the internal structure of the search mechanism of the agent.

The search mechanism of agent is significantly supported by Hagerstrand space constraints. Hagerstrand space is an evolving trans-disciplinary scheme on spatio-temporal events that uses different types of constraints, including psychological, physical, coupling, and authority ones [42]. In order to set this constraint, we use tools created in a tensor space (Figure 3, parts 5–7). Indeed,

instead of simply analyzing spatial events in a crisis environment [43], we use an expert concept space for all kinds of needs and adapt it to our solution.

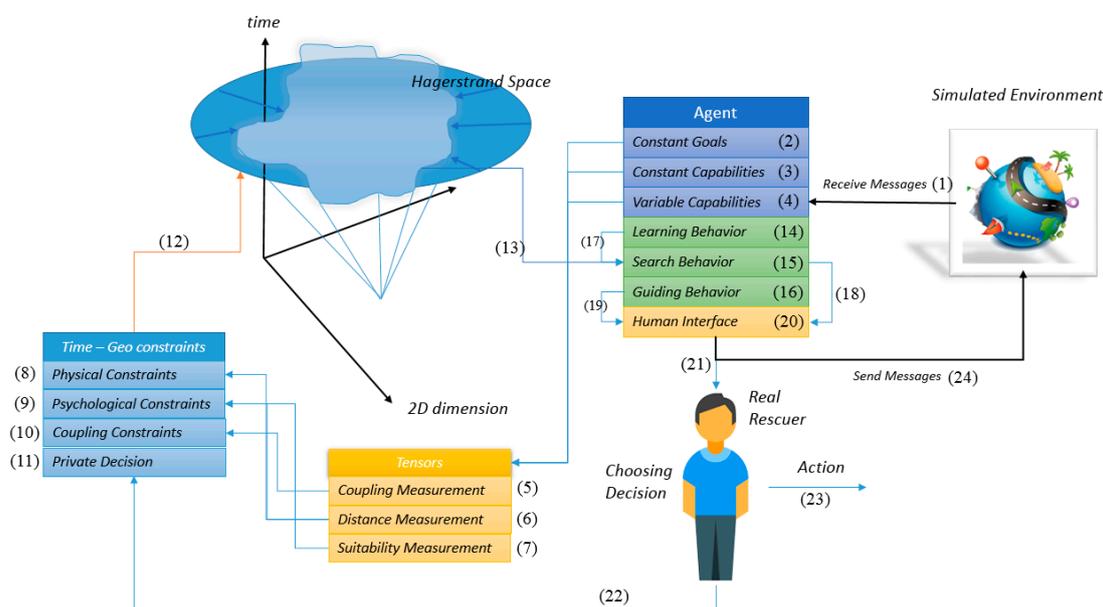


Figure 3. Agent’s search mechanism based on Time-Geography Constraints.

Physical constraints (Figure 3, part 8) consider the possibility of practical action in space-time. This clause in our research is determined by a tool (Figure 3, part 6) that measures the geometric distance between the rescuer and the rescuee. Psychological constraints (Figure 3, part 9) also express the possibility of an action by psychological readiness. In our study, the value of this constraint is expressed by a tool (Figure 3, part 7) that measures the level of fatigue of the rescuer. The coupling constraint (Figure 3, part 10) also expresses the probability that an activity will occur due to the fit of factors in space-time. This constraint in our research examines the proportion between the rescuer and the injured request for the required expertise. The value of this constraint is also measured by another tool (Figure 3, part 5) in tensor space.

These constraints a convex cone in Hagerstrand space (Figure 3, part 12) and then offer a list of appropriate requests for rescuer agent (Figure 3, part 13). This list is the actual output of the agent’s search mechanism (Figure 3, part 18). The list is passed to the human interface (Figure 3, part 20). Then the agent sends it (Figure 3, part 21) to its respective rescuer in the crisis field through the crowdsourcing service. Finally, the rescuer chooses which request to take. This selection returns it to the search mechanism (Figure 3, part 22) and completes it. In this way, the agent, is informed of the real rescuer’s decision, prepares the appropriate messages, and sends them to the simulation environment (Figure 3, part 24).

In the following sections, the mathematical models and tools, that make the search mechanism of the agent, will be considered.

3.1. Mathematical Models and Tensor Tools

The application of numerical models in crisis management is common [44], but in this research, a framework for these models was defined. To implement mathematical models, a framework is used in tensor space (Figure 3, parts 5–7). In order to adapt the problem space to this framework, the important parameters of the problem space are considered the main dimensions of the tensor space. We assume that four types of rescuers serving services include debris removing as d , carrying as c , medical aiding

as m and feeding as f . An injured one declares his required volume in the mentioned topics, based on the following vector, as Equation (1):

$$V_{injured}(d, c, m, f) = r_i \vec{d} + r_{i+1} \vec{c} + r_{i+2} \vec{m} + r_{i+3} \vec{f} \quad r_i \in [0, 10]. \quad (1)$$

The values of r_i are determined by the injured or they express their priority of the value, while other values are set, based on previous knowledge. Then, steps, such as selecting priority, ranking rescuer agents, and sending messages to the candidate as rescuer agents are followed.

In order find the priority of required services, assume that each rescuer agent has a state vector, which indicates its capabilities in four major parameters as shown in Equation (2):

$$V_{Rescuer}(d, c, m, f) = r_i \vec{d} + r_{i+1} \vec{c} + r_{i+2} \vec{m} + r_{i+3} \vec{f} \quad r_i \in [0, 10]. \quad (2)$$

Then, S produced in Equation (3) expresses the suitability of rescuer agents against any given injured ones, which determines which rescuer agents are suitable for any given Injured person, based on their specified tasks:

$$\vec{V}_{Rescuer} \cdot \vec{V}_{Injured} = S. \quad (3)$$

Equation (3) of this tensor tool (Figure 3, part 5), in some way, expresses the quantity of coupling constraint (Figure 3, part 10).

Any given rescuer, in addition to its capabilities, has another extra parameter that indicates his remained power to work as p . The value of the p for injured agents is determined based on his demand power. For example, p equals to 10 for an extreme exhausted injured or rescuer agent.

$$Y = \left[\left(\begin{bmatrix} r_i \\ \vdots \\ r_{i+4} \end{bmatrix}^{Rescuer} \right) - \left(\begin{bmatrix} e_i \\ \vdots \\ e_{i+4} \end{bmatrix}^{Injured} \right) \right] \cdot \left[\left(\begin{bmatrix} r_i \\ \vdots \\ r_{i+4} \end{bmatrix}^{Rescuer} \right) - \left(\begin{bmatrix} e_i \\ \vdots \\ e_{i+4} \end{bmatrix}^{Injured} \right) \right] \quad (4)$$

The Equation (4) gives dot product of two main vector differences. The result is a scalar, which is used to compare the suitability of different pairs and rank rescuer agents for any given injured one. The fifth element of each vector indicates p parameter. The Equation (4) of this tensor tool (Figure 3, part 7), in some way, expresses the quantity of suitability constraint (Figure 3, part 9). r and e , are state vectors of both rescuer and injured agents (1), (2) with an additional parameter for fatigue (p).

If a rescuer agent needs any resources for his mission (for example, if a rescuer wants to temporarily treat an injured person with the help of medicine or medical devices, those medical supplies are considered as a resource), a dyad product will replace the rescuer state vector as in Equation (5), whereas, in another study, it has been shown that the combined effect of a rescuer vector and a source is almost the same as the effect of their dyad product:

$$A_{5 \times 5}^{Trans} = \begin{pmatrix} r_i e_i & \cdots & r_i e_{i+4} \\ \vdots & \ddots & \vdots \\ r_{i+4} e_i & \cdots & r_{i+4} e_{i+4} \end{pmatrix}. \quad (5)$$

Therefore, coupling constraint value (Figure 3, part 10) is modified as Equation (6):

$$(A_{5 \times 5}^{Trans} \times \vec{V}_{Rescuer}) \cdot \vec{V}_{Injured} = S. \quad (6)$$

Note that, S denotes the suitability of a rescuer and an injured based on their fixed properties, while Y expresses their suitability on the basis of variable properties, such as fatigue.

space from raster to vector mode. Finally, “Add Z Information function” [47] extracts these values for rescuer agents. Thus, the injured agents have a real distance to the rescuer agents and can send a message to the suitable ones. Following codes in *arcpy*, the following processes are executed:

```
10 arcpy.sa.CostDistance(injured,cost,500,not)
20 arcpy.InterpolateShape_3d(cosdist,rescuer,rescuer1,5,non,none,none,none)
30 arcpy.AddZInformation_3d(rescuer1,none,none)
40 arcpy.SelectLayerByAttribute_management(rescuer1, "ADD_TO_SELECTION", "'Dist' < 500")
```

Line 10, calculate pixel’s accessing cost to location of injured and based on cost layer, 20 transmits value to rescuer objects, 30 extracts transmitted information from geometry field of objects and 40 runs a spatial query on rescuer class.

However, such a function acts slowly and push a considerable process load to the system. In order to allay this load, pure raster Cost layer should be taken. Then, its watersheds could be extracted by hydrological analysis functions. After modifying the watershed networks geometrically, it can be transformed into a network dataset. Now, the problem is transformed to the vector mode and can be solved by using the proposed solution.

Furthermore, it is possible to use a complex network, that is related to different types of roads. This network, includes weaker paths, such as gravel road or urban ones. Environmental frictions could be assigned to road elements and such extra attributes make the cost attribute of network dataset. Thus, the problem is transformed to vector mode by using this method. Finally, the problem is simply solved via Euclidean distance function when there is no information about raster environment frictions.

3.3. Effectiveness Cone

In time geography methods and researches, the spatio-temporal domain, which is affected by a given point in the same space, is introduced as the effectiveness cone of that point (Figure 3, part 12) [41]. In traditional time geography issues, this cone is usually used as an absolute constraint, while we prefer to use it conceptually. Thus, the whole of it has been used as a criteria for ranking rescuers against any given request.

Equation (7) is used to rank the rescuers against each aiding request,

$$R_{ij} = \frac{s_{ij}}{\sqrt{Y_{ij}^2 + D_{ij}^2}} \quad (7)$$

where, R_{ij} is the rank of rescuer j for request i , s_{ij} is the coupling value between aiding request i and rescuer j , comes from Equation (3), Y_{ij}^2 is psychological value comes from Equation (4) (Figure 5, mark 2) and D_{ij}^2 is the distance between aiding request and rescuer (Figure 5, mark 1), which is calculated by the distance function in Sections 3.2.1 and 3.2.2. So, assuming m requests and n rescuers, we will have the proposed matrix M as Equation (8):

$$M = \begin{pmatrix} R_{11} & \cdots & R_{1n} \\ \vdots & \ddots & \vdots \\ R_{m1} & \cdots & R_{mn} \end{pmatrix}_{m \times n} \quad (8)$$

Which R_{mn} is the rank of rescuer n for request m . At the end of each of the columns of this matrix, is sent to its corresponding rescuer (Figure 3, part 21). Rescuer makes his choice from this list.



Figure 5. Effectiveness cone of given aiding request.

4. Data, Implementation, and Evaluation

In this section, the method of data preparation is discussed first, then the implementation model is expressed in the software environment, and finally the solution evaluation results are examined.

4.1. Data Preparation

Five classes of data are distinguished in our solution; base maps, resource and path maps, as well as rescuer and request maps. These five data classes are the main input of the simulator. Except for base maps directly embedded in the simulator, the main path of the rest of the data starts with the user of the system almost. The user of the system (The upper part of Figure 6) that uses spatial crowdsourcing service may be an applicant for help, a rescuer or any other one from crowds.

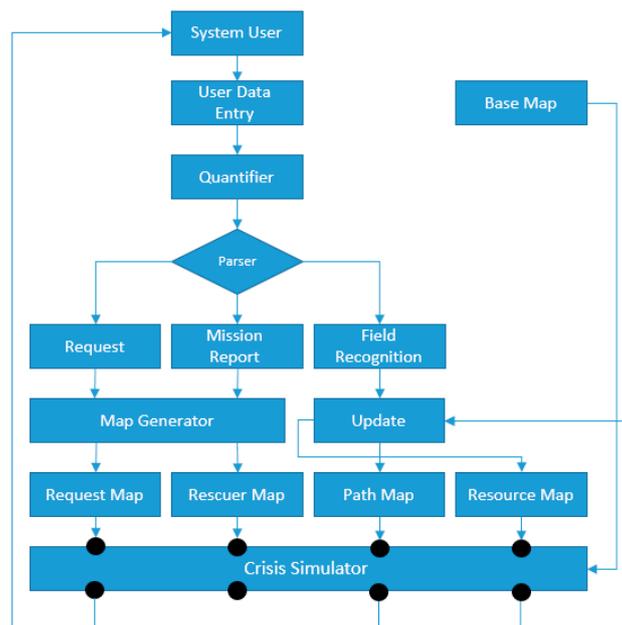


Figure 6. Crowdsourcing data service, crowdsourcing users, and simulator both update maps.

As shown in Figure 6, the system user enters his data into the system through the crowdsourcing service. Since the main processing is done in the mathematical space, the quantifier unit translates some of the input qualitative values into the numerical values required by the simulator. Since this service can accept various information, a parser then discovers the data. Request and report data are directly submitted to the map generator. The field recognition data is entered the update section. Thus four main maps, including request, rescuer, path, and resource maps, are produced. All data comes into the simulator with the base map. In the end, the simulator send choices to the user, and a part of its output is sent to the update unit to be used to update the resource and path maps.

Quantifier unit in Figure 6 uses a specific standard, in order to attach the crowdsourcing service data to other information for use in the simulator. Table 1 summarizes the standard coding in data fields, which was used in Aqala flood relief operation. In the protocol for transferring and storing data, from the user to the server, and vice versa, we used the code contained in the *code* column. These codes are replaced at the output of the user interface software by the domains in the *field content*.

Table 1. Data codes which was used in Aqala flood relief operation.

Data Group	Data Type	Code	Field Content	Domain
Spatial Crowdsourcing Services	Request	1–4	Food & Water/Medical/Shelter/Transfer	$0 \leq \mu \ll 10$
	Mission	5–8	Food & Water/Medical/Shelter/Transfer	$0 \leq \mu \ll 10$
	Field Recognition	1–4	Food & Water/Medical/Shelter/Transfer	$0 \leq \mu \ll 10$
	Mission Report	10	OK/Suspended	0,1
Resources	Foods	20	Food & Water	Long Integer
		22	Drugs	Long Integer
		23	Trucks and instruments	Long Integer
	Device, tools	24	Boats	Long Integer
		25	Generator	Long Integer
		26	Water pump	Long Integer
Path	Street	31	Flooded, depth in centimeter	Long Integer
		32	Traffic jam, waiting in minute	Long Integer
		33	Debris, volume in cubic meter	Long Integer

4.2. Implementation

4.2.1. DBMS Implementation

In the core of the system, the geodatabase, the crisis environment is simulated using relational classes. Each of these classes is somehow connected to the actual objects of the crisis. Some classes have a consistent and non-intelligent behavior and simulate only objects that have an interactive role, while others are smart. At the center of this class is the “Operation” class (Figure 7, part 7), which is equivalent to the spatial crowdsourcing service.

This class is completed in conjunction with the user class (Figure 7, part 11) and extracts the required parameters from classes such as Request (Figure 7, part 2) and Expert (Figure 7, part 9). Allocation class (Figure 7, part 10) is a communication center designed for recording communications between other classes. The User class (Figure 7, part 11) is a parent of two classes with smart behaviors, Rescuers (Figure 7, part 14) and Requests (Figure 7, part 2). These two classes simulate and execute an intelligent behavioral search through built-in mechanisms. The search environment employs two types of Topology (Figure 7, part 13) and Raster space (Figure 7, part 16) in searching for smart agents. The Position class (Figure 7, part 18) generates the position of all the objects in the environment. Of course, this class acts smartly, with its own mechanism on the bank of points, produces instantaneous location for each requestor.

Specification class (Figure 7, part 8) is the main supplier of the parameters required and observed on the service. Resource class (Figure 7, part 14) is one of the non-smart and auxiliary classes of the environment that puts all kinds of resources (Figure 7, part 20) required by the User’s class into

consideration, and adjust their parameters accordingly to the new inventory. The Captions class (Figure 7, part 1) has nothing to do with the main object class, but just to customize the display method and the instant content of the mobile users' GUI.

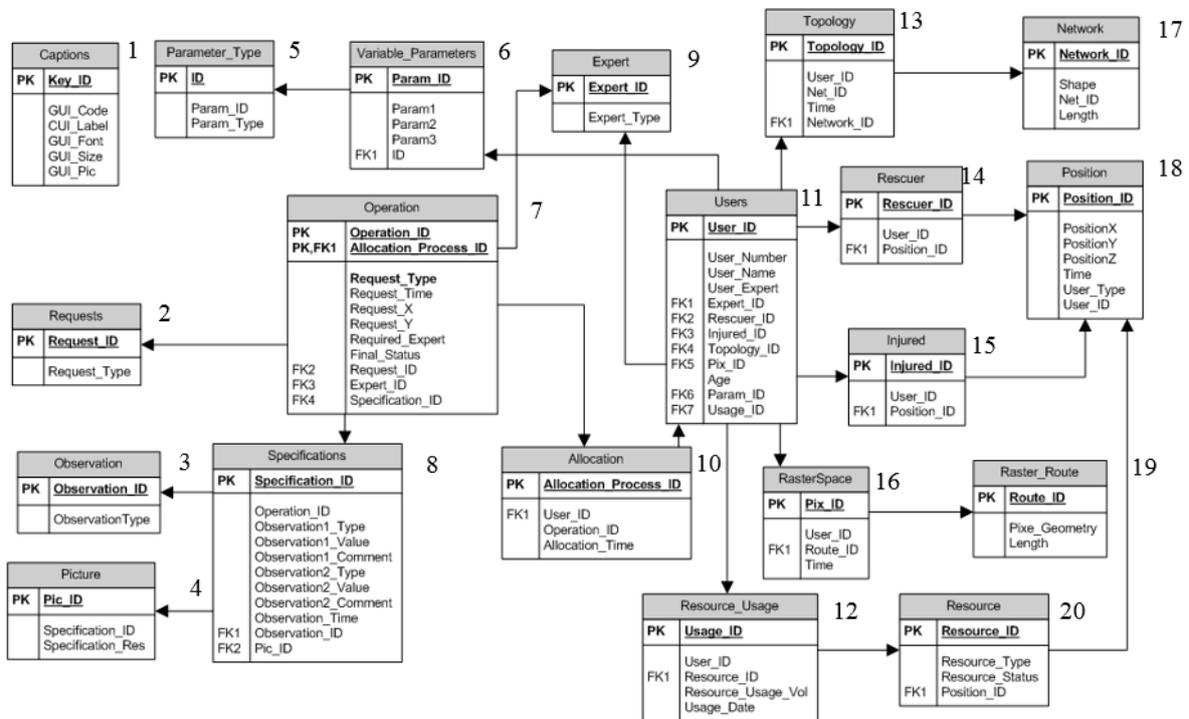


Figure 7. Relational database of system main objects.

4.2.2. Hybrid Programming for Core Development

At the end, an enterprise GIS, including our solution to the allocation problem, was designed. This system uses spatial crowdsourcing service to obtain spatial data from users and re-distribute system decisions made to users. Other tasks, such as management, cartography, and information analysis, are done in the GIS's common practice. But to implement a multi-agent solution in the context of the GIS, a combination of programming methods was used.

Although, traditional techniques in programming such as modular, along with structured- and object-oriented method, involve many benefits, there are some problems with these methods [48]. These methods contribute to the implementation process of huge software construction by using separate and independent modules [49]. In addition, the decomposition of codes can facilitate the development of the most famous software in the world. However, developers should be informed of some important issues for hybrid and complex systems.

Further, these systems should support different plug-in techniques and raise the total speed and design mechanisms to run modules at the requested times. Furthermore, object-oriented programming makes the modelling simple, but it makes the programming too difficult in some cases [50]. Programmers should think about all objects, events, interactions, synchronization time, and the like. However, programming languages have some benefits in developing specific types of projects. Automation software such as CRM [51] can widely spread hybrid programming by using the benefits related to both sides [52]. Using hybrid programming allows developers to deploy the richest tools available in some functional environment, such as a database, along with many written standard codes.

In addition, programming within a database raises the integrity of codes and optimizes the use of data. On the other hand, functional languages are really more simple to be deployed when mathematics is the background structure of the required mechanism. Therefore, the decision was made to do the implementation in a hybrid mode.

to go and what to do (Figure 9, part 10). They use the resources to deploy the relief process in a crisis environment (Figure 9, part 12). This is while the GIS server can provide some information needed by the professional rescue teams or the monitoring crisis management center (Figure 9, part 13 and 14).

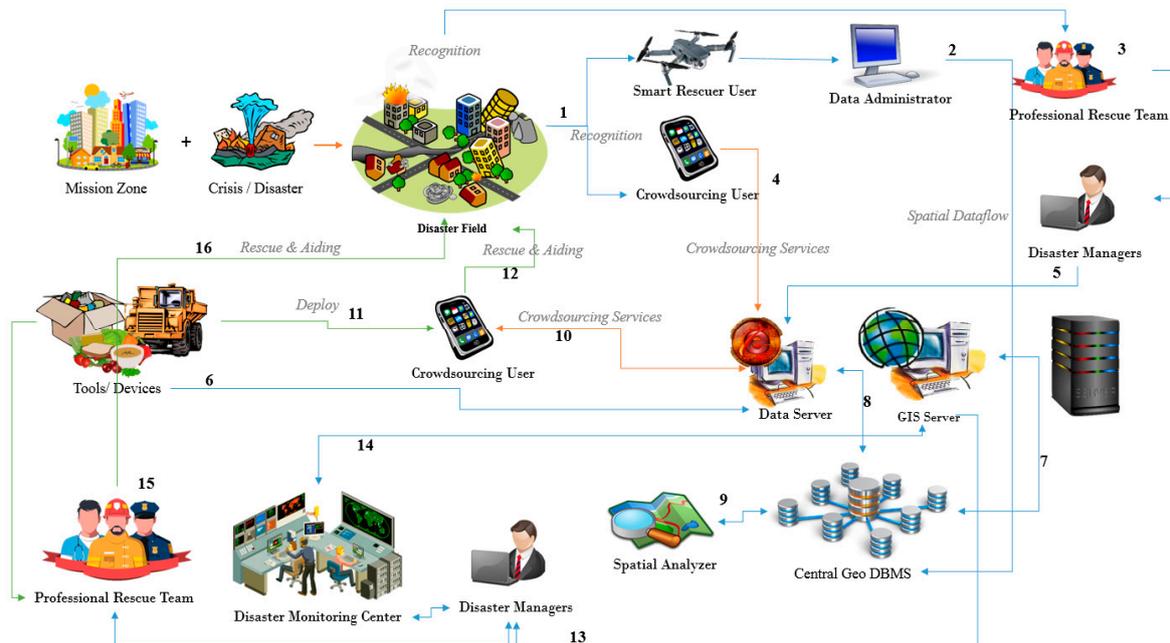


Figure 9. System schematic diagram and dataflow among system elements.

The system has several users, and each user needs a specific cartographic view of the data. These cartographic views are generated by the spatial analyzer (Figure 9, part 9) and are published via the GIS server. Each cartographic view is tailored to a need. As time goes on and the discovery of newer needs, newer views are being made.

Among the system users, the SDE (Spatial Data Engine) user is connected directly to the geodatabase. With access to all spatial and descriptive information, the user can control all of the operations in the system and is in some way a system administrator at the command center. In addition to the capabilities of other users, this user can access spatial classes with a variety of spatial processing with the help of tools, functions, and spatial queries.

For example, all people who has remained constant for a significant period of time before, and after, the crisis and have not made any move can be identified as possible points for rescuing mission. Figure 10a shows these points and the time of stopping since the crisis occurred.

In a similar way, it can monitor the status of resources available at the service stations in real time, in terms of remaining percentage or capacity, in order to compensate for reduced resources. The Figure 10b is an image of the software, in which the residual amounts of water, medicine, food, and equipment are displayed to the logistics managers. Similarly, it categorizes rescuers according to their fatigue in different parts of the city, and locates them on a map or creates crisis density map and many other interesting application maps.

Another system subscriber is the user of the map server on the Internet. This user also has the ability to identify, limit information entry and the display of cartographic information, and, of course, has a very limited tool for analyzing spatial information. This user can use the WFS and WMS services directly on the web and may have a client-side tool or just use the web browser. Figure 10c shows an overview of the region’s different needs and requests with the satellite background image along with main roads.

Perhaps the main user of this system is the crowd of people, which is almost entirely connected to the crowdsourcing services. Different crowdsourcing users need to have the same, simple user interface

on their smart phone to work in this system. This GUI, who uses the Open Street Map service as a background map, is the main interface of the crowd. This GUI, and the subsequent crowdsourcing services, have three main roles. The first role of the interface and service is to register aiding requests. Two combo boxes are used to register the application, and the values are determined by the original server. In other words, the system administrators can adapt domains of these combo boxes to suit the type and the severity of the crises. Figure 10d shows crowd mobile user interface.

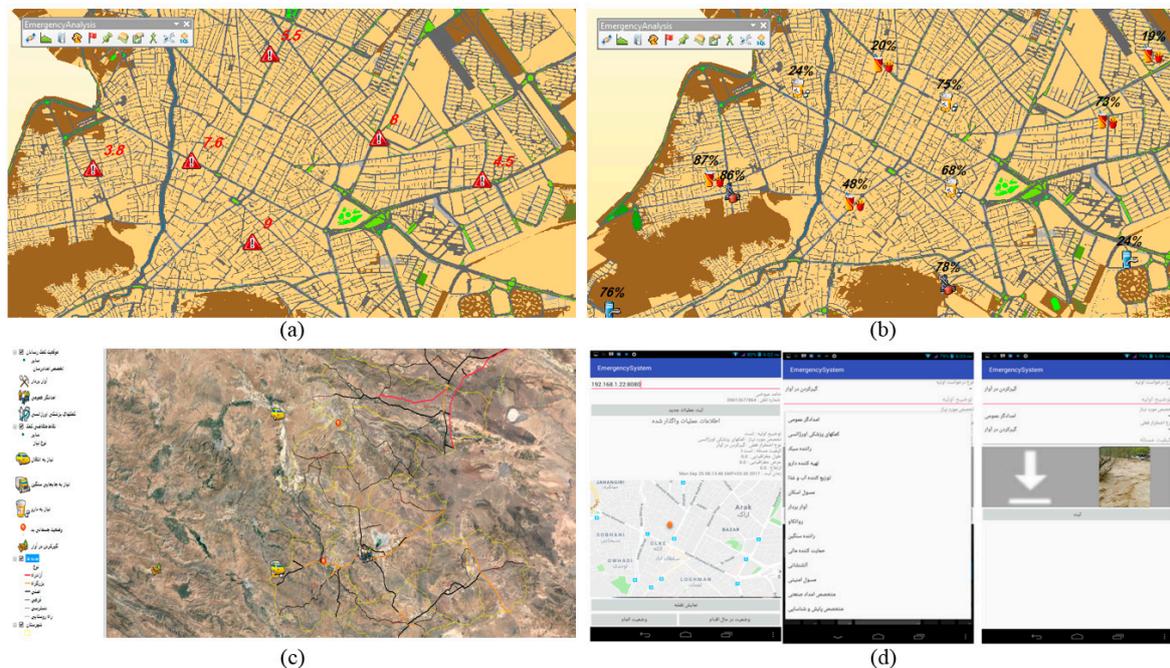


Figure 10. (a) Designed analysis for finding system users with long stopping time since crisis occurred, (b) Displaying the remaining amount of the critical resources in the graphical user environment, (c) A GIS server user, server side, in a web browser with main requests' types, (d) Crowdsourcing mobile user interface and its components.

Using texts to reports situations in crisis field would be difficult. Therefore, it is possible to use icons in GUI instead of using simple buttons with text labels. In addition to the combo boxes, two additional parameters are also taken by this interface, which are proportional to the type of selected combo box cases that express their quantity.

The second role of this interface is crisis field identification. That is, users can report the status of critical areas in the crisis field instead of applying for help. Users can also upload two images for this purpose. The third role of this interface is to guide the rescuers for emergency missions. In other words, the system first identifies the critical users with aiding requests, and then suggests them to the appropriate rescuers, based on the method described above.

A rescue user comes in contact with a list of applicants by connecting to the system and receiving a payroll service. Rescuers control the list and select the right item among them, then the system guides him with its map menu. After reaching the site, if the operation is successful, the rescuer taps the status key and informs the system that the request has been successfully completed. Otherwise, with the touch of the Suspension key, the system will realize that it needs to send other rescuer users to this point.

4.3. Evaluation

Our proposed solution to the allocation problem in crisis management involves actions in the context of an enterprise GIS and the importance of this solution arises in interacting with the

crowdsourcing service and various components of the GIS. In this way, various methods were used to evaluate different parts of the system, service, and solution.

Aqala is a city in Golestan province of Iran and the city center of Aqala County. The city now has about 35,000 inhabitants. The city of Aqala is located on the sides of the Gorgan River. After heavy rains in March 2019, the city and its county was flooded by the Gorgan River. For the first time, the spatial information system was launched on the basis of the proposed solution, and crowdsourcing services was practically tested during the relief operations of the city.

In order to evaluate the performance and functionality of the GIS and crowdsourcing service in the field of spatial data, the system was tested experimentally in the flood of the Aqala. We formed two rescue teams, each team consisting of five people, and two system users were considered for each team. The first team worked with the guidance of the system user in the northern side of the Gorgan River, while for the second team, the system user registered only the operations performed by the team in the southern side of the river. For the first team, before the start of the relief operation, the system's user was registering the requests needed by the people through the system. The first rescue team started solving the problems based on system guidance. But the second team only acted on the basis of their knowledge and commands to the relief operation. Of course, due to the limited time and conditions, we used the system in offline mode. The information of the two user smartphones, and the one on the laptop as the central server, was able to synchronize almost every six hours.

Table 2 shows the summary of collected information and operation results by team 1, compared to team 2.

Table 2. Recorded information of missions, successes and the fails of rescue teams in six days.

			Water/Food	Shelter	Medical	Transport	Total	Percent
Team 1	Day1	Request	703	15	0	18	736	
		Success	517	2	0	0	519	71
	Day2	Request	867	24	0	34	925	
		Success	711	5	0	3	719	77
	Day3	Request	697	89	0	48	834	
		Success	715	19	0	5	739	88
	Day4	Request	743	129	3	15	890	
		Success	706	28	2	11	747	83
	Day5	Request	542	118	7	9	676	
		Success	503	31	5	4	543	80
	Day6	Request	465	94	13	7	579	
		Success	450	25	12	3	490	85
Team 2	Day1	Mission	750	0	0	0	750	
	Day2	Mission	700	0	0	0	700	
	Day3	Mission	700	4	2	0	706	
	Day4	Mission	750	7	3	4	764	
	Day5	Mission	650	6	0	6	662	
	Day6	Mission	650	3	5	7	665	
		Failed	183	78	9	4	274	41
Total Request Team A		4640	Total Request A. without Food			623		
Total Success Team A		3757	Total Success A. without Food			155		
Total Success Team B		4247	Total Success B. without Food			47		

When the second user of the system, in the six-day period, only recorded operations carried out by team two, after the completion of the Sixth Day Relief operation, this user recorded unresolved cases of relief requests from that area.

According to Figure 11 and Table 2, the results indicate that the second team failed to meet about 40 percent of the relief needs. On the other hand, the team's main activity merely involved the

distribution of food without purpose. While, the first team, that operated under the system conduction, had responded to a different range of requests with over 85% coverage.

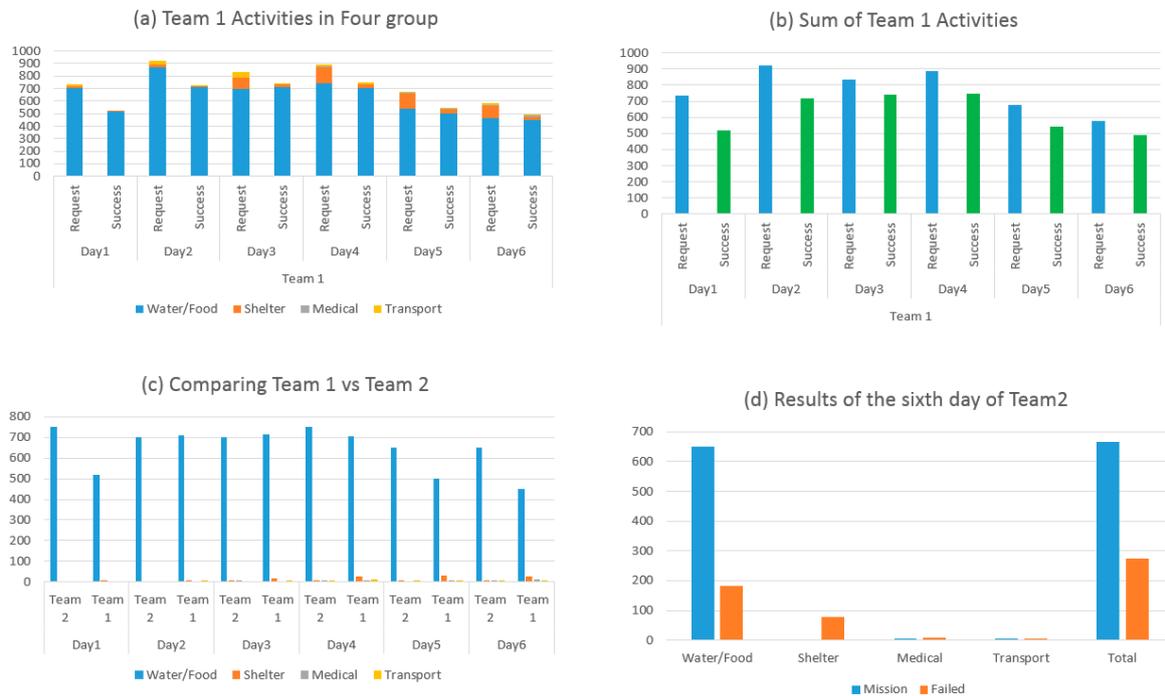


Figure 11. (a) The diverse activities of Team 1, also highlight their success rate in solving the problems in six days (b) Sum of all Team 1 activities and their success rate in six days, (c) Compare the number of activities performed by Teams 1 and 2, (d) Results of the sixth day of Team 2.

Team 1 worked in the northern area of the river with an approximate area of 750,000 square meters and Team 2 in the southern region of the river with an approximate area of 446,000 square meters. Figure 12a, shows the position and density of green relief missions and the remaining red problems in two regions on the sixth day. Team 1, which worked with the system’s guidance in the area, had fewer unresolved points and, on the other, covered more area.

Figure 12b shows the magnitude of water inundation on both sides of the river and the various services provided on the sixth day. First, the map is based on the information obtained from the crowdsourcing service. Thus, the service gives us different developmental capabilities. Second, Team 1 has been able to offer better and further services in the fields of home improvement, sanitation, and transportation.

Figure 12c is a density map of the remaining problems on the sixth day. First, the production of such maps by the GI system is very effective for crisis management, and secondly, Team 1 has managed to reduce the volume of problems in the area by conduction of the system and utilizing the crowdsourcing service. Figure 12d is a service density map by the two teams on the sixth day. Again, Team 1 has managed to provide more and a wider variety of services.

In the next section, in order to evaluate the efficiency of the proposed algorithms and solutions, several scenarios were defined in software and simulation environments.

In the first scenario, a simple competitive multi-agent environment was designed in *AnyLogic* [54]. Further, agents were programmed to touch some constant points as injured in a network space, among which those with more touching won. In order to test the system overall speed, the agents’ numbers were gradually increased in both *AnyLogic* and the developed system. As shown in Figure 13a, the speed of the agent-based model fails as the agent abundance exceeds 23. As illustrated, the real agent-based system, with serious complicated missions, fails even sooner, which proves the functional strength for tackling bulky duties.

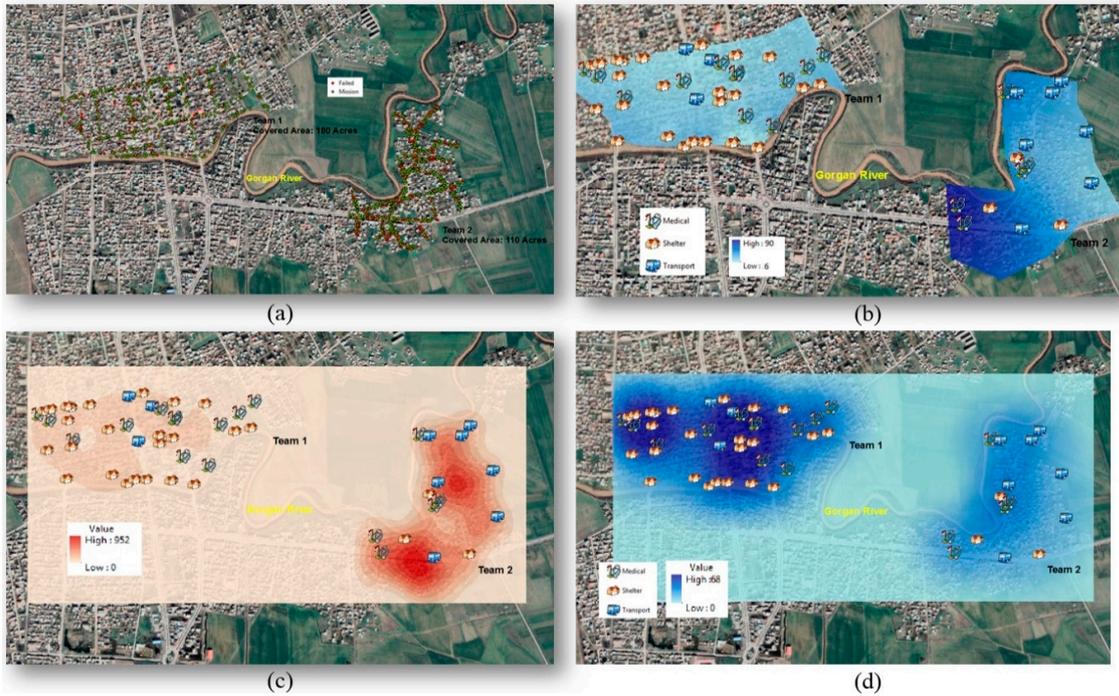


Figure 12. (a) Location and the extent of relief mission on both sides of Gorgan River in Aqala, (b) water inundation rate in two test zone, (c) Unresolved problem density map on the sixth day, (d) Density and diversity map of performed services on the sixth day without considering the food distribution service.

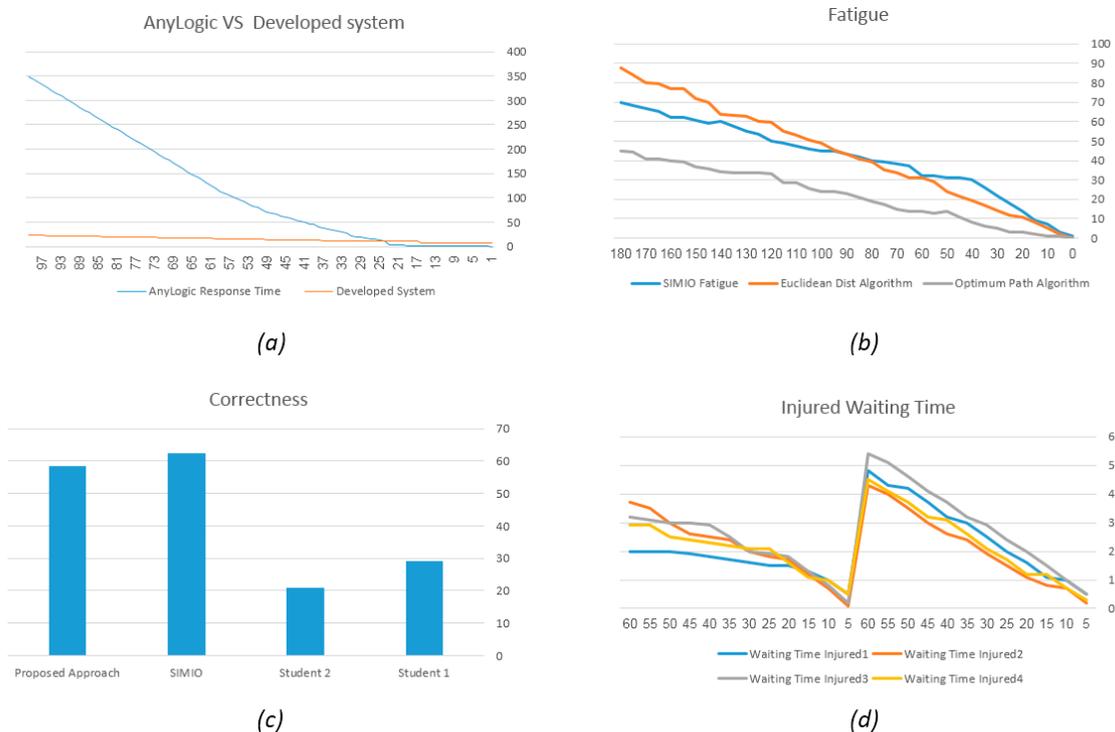


Figure 13. (a) System response time in AnyLogic and developed system; (b) agent fatigue; (c) number of correct choices; (d) Compare total waiting time of injured objects in two cases.

The second scenario was defined to measure the overall fatigue of system elements. In this regard, four medical aiding rescuer agents and 10 injured ones were defined in the proposed system and SIMIO [55]. Then, two tests were run for the proposed system. Regarding the first test, the agents

used the Euclidean shortest path while they used the optimum path algorithm for the second test. SIMIO (Simulation Software system) used its own algorithm to find the optimum path. As displayed in Figure 13b, the designed optimum path decreases the overall fatigue of system elements. In other words, the proposed solution had the best results, compared to the Euclidean algorithm which reveals the worst. In this graph, the x-axis represents the time in minutes, and the y-axis represents the fatigue of agents in percentage.

Regarding the third scenario, four rescuer agents and 12 injured agents in different classes were defined to test the accuracy of rescuer agent selections. Each injured used two services of different classes. An expert rescuer was asked to choose the proper injured agents for any given rescuer agents. In addition, two other university students were asked to choose. Then, all results were compared to SIMIO and developed system selection. As illustrated in Figure 13c, the similarity of decisions was taken by the proposed methods and the SIMIO to expert rescuer was approximately two times more than the volunteer students.

The fourth scenario focused on the total system operation. First, 2 rescuer agents and 4 injured were initialized in the proposed system. Then, two teams including 4 volunteer students participated in in the test. Further, the location of the injured and rescuers were distributed in university campus. One group was asked to the mission just using treasure map of Garmin, while, the second group was conducted by the system. Accordingly, the waiting time were measured according to the chronometers on the injured agents. In relation to the comparison of the injured waiting time, the group conducted by the system could decrease the waiting time approximately to 62%, compared to another group. In other words, considerable time was consumed for the team which just used GPS for recognizing and making decisions (Figure 13d). In this graph, the X-axis represents the total operation time in minutes. The Y-axis, represents the delay time of rescuer agents in minutes.

Figure 13a, displays the effect of increasing the number of agents on the system response time in AnyLogic and developed system. The vertical and horizontal axes represent the number of agents and time in seconds. Figure 13b indicates agent fatigue volume. The blue line represents the fatigue of agents in SIMIO; the orange line belongs to the developed system, while it uses simple Euclidean distance, and; the gray line displays fatigue volume of agents in the developed system by using optimum path algorithm. Figure 13c shows number of correct choices, which are made by agents in the proposed system. It compares the correct choices that have been made by our system, SIMIO and two volunteer students, based on an expert selection. The right half of Figure 13d represents the total waiting time of injured objects while two volunteers just use Garmin treasure map, where the horizontal axis indicates the whole mission time in minutes and the vertical axis represents the total waiting time in minutes. The left half of Figure 13d indicates the waiting time of objects, while two other volunteers were led by crowdsourcing services.

5. Conclusions

We believe that the use of spatial crowdsourcing services can create new developments in spatial information systems. These services can be more powerful than Volunteer Geographic Information in producing data. This service can be used as a tool for collecting spatial information and also serve to guide individuals in an integrated coherent operation. In other words, these services turn spatial information systems from a consultative system to a strategic and actor system in scenes, like crisis management. Using such a service for major problems, such as crisis management in a vast area, utilizes the potential of people to solve the problem in a positive way. Its experimental use in the flood Aqala crisis showed that these types of services could be a new jump in spatial information systems after mobile GIS and VGI (Volunteered Geographic Information).

But the implementation of such a service in the context of the GIS system involves two basic issues; designing and managing automated services. In this regard, the use of multi-agent environments made the solution much simpler. Because designing such environments is far easier than designing a centralized system. The use of tensor space and its models also made the process easier.

On the other hand, designing the process is implemented in the commercial software environment, such as SQLServer, there is more convenience and faster development. For example, a wide range of features, such as classes, views, stored procedures, and triggers, that exist in the database space, were used in agent programming. By using such a solution, the system can easily and quickly generate many agents to act in simulation environment. This is very important in the development of spatial information systems.

Of course, such a method of designing and developing, centered on the database and its tools, reduces overall system speed. Although, similar systems can show a higher process speed, this method will be the only practical way when the number of users increases significantly. Conventional simulation systems are basically unable to work with a large number of users. In addition, the proposed solution for decision mechanism of agents, has a small sensitivity to calibration parameters and has found significant speed through the use of math and tensor tools.

But in the end, we can point to the interaction of proposed solutions with spatial information systems. In this interaction, both parties serve each other. In other words, this solution is implemented in the structure of a spatial information system, and along with other tools in this system, it will further empower it. On the other hand, the solution uses the tools of these systems. For example, the physical constraint in Hagerstrand space is the same distance in both, the Raster and vector mode, and uses tools that are available on these spatial information systems.

Author Contributions: Conceptualization, Hooshang Eivazy and Mohammad Reza Malek; methodology, Hooshang Eivazy; software, Hooshang Eivazy; validation, Mohammad Reza Malek; formal analysis, Hooshang Eivazy; investigation, Hooshang Eivazy; resources, Mohammad Reza Malek; data curation, Hooshang Eivazy; writing—original draft preparation, Hooshang Eivazy; writing—review and editing, Mohammad Reza Malek; visualization, Hooshang Eivazy; supervision, Mohammad Reza Malek; project administration, Hooshang Eivazy; funding acquisition, Hooshang Eivazy

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Xiong, J.; Li, J.; Cheng, W.; Wang, N.; Guo, L. A GIS-Based Support Vector Machine Model for Flash Flood Vulnerability Assessment and Mapping in China. *ISPRS Int. J. Geo Inf.* **2019**, *8*, 297. [[CrossRef](#)]
2. Cheng, C.; Zhang, T.; Su, K.; Gao, P.; Shen, S. Assessing the Intensity of the Population Affected by a Complex Natural Disaster Using Social Media Data. *ISPRS Int. J. Geo Inf.* **2019**, *8*, 358. [[CrossRef](#)]
3. Central Board of Secondary Education. *Natural Hazards and Disaster Management*; Central Board of Secondary Education: Delhi, India, 2006.
4. Jeansoulin, R. Multi-Source Geo-Information Fusion in Transition: A Summer 2019 Snapshot. *ISPRS Int. J. Geo Inf.* **2019**, *8*, 330. [[CrossRef](#)]
5. Fakhruddin, B.S.; Schicka, L. Benefits of economic assessment of cyclone early warning systems: A case study on Cyclone Evan in Samoa. *Prog. Disaster Sci.* **2019**, *2*, 100034. [[CrossRef](#)]
6. Mizutori, M. From risk to resilience: Pathways for sustainable development. *Prog. Disaster Sci.* **2019**, *2*, 5. [[CrossRef](#)]
7. Riccardi, M.T. The power of crowdsourcing in disaster response operations. *Int. J. Disaster Risk Reduct.* **2016**, *20*, 123–128. [[CrossRef](#)]
8. Felice, P.D.; Iessi, M. A Citizen-Sensing-Based Digital Service for the Analysis of On-Site Post-Earthquake Messages. *ISPRS Int. J. Geo Inf.* **2019**, *8*, 136. [[CrossRef](#)]
9. Schimak, G.; Havlik, D.; Jasmin, P. Crowdsourcing in Crisis and Disaster Management—Challenges and Considerations. In Proceedings of the International Symposium on Environmental Software Systems, Melbourne, Australia, 25–27 March 2015.
10. Auferbauer, D.; Tellioglu, H. Centralized Crowdsourcing in Disaster Management: Findings and Implications. In Proceedings of the 8th International Conference, Troyes, France, 26–30 June 2017; pp. 173–182.
11. Poblet, M.; García-Cuesta, E.; Casanovas, P. Crowdsourcing roles, methods and tools for data-intensive disaster management. *Inf. Syst. Front.* **2017**, *20*, 1363–1379. [[CrossRef](#)]

12. Yang, T.; Xie, J.; Li, G.; Mou, N.; Li, Z.; Tian, C.; Zhao, J. Social Media Big Data Mining and Spatio-Temporal Analysis on Public Emotions for Disaster Mitigation. *ISPRS Int. J. Geo Inf.* **2019**, *8*, 29. [[CrossRef](#)]
13. Taj, S.; Arian, Q.; Momen, I.; Zubedi, A. To apply Data Mining for Classification of Crowd sourced Software Requirements. In Proceedings of the ICSIE '19 Proceedings of the 2019 8th International Conference on Software and Information Engineering, Cairo, Egypt, 9–12 April 2019.
14. Estellés-Arolas, E.; González-Ladrón-De-Guevara, F. Towards an Integrated Crowdsourcing Definition. *J. Inf. Sci.* **2012**, *38*, 189–200. [[CrossRef](#)]
15. Prpić, J.; Taeihagh, A.; Melton, J. The Fundamentals of Policy Crowdsourcing. *Policy Internet* **2015**, *7*, 340–361. [[CrossRef](#)]
16. Prpić, J.; Shukla, P. Crowd Science: Measurements, Models, and Methods. In Proceedings of the 49th Annual Hawaii International Conference on System Sciences, Kauai, HI, USA, 5–8 January 2016.
17. Brabham, D.C. Crowdsourcing as a Model for Problem Solving an Introduction and Cases. *Int. J. Res. New Media Technol.* **2008**, *14*, 75–90. [[CrossRef](#)]
18. Harrison, S.E.; Johnson, P.A. Crowdsourcing the Disaster Management Cycle. *Int. J. Inf. Syst. Crisis Response Manag.* **2016**, *8*, 24. [[CrossRef](#)]
19. Das, A.; Mallik, N.; Bandyopadhyay, S.; Bit, S.D.; Basak, J. Interactive information crowdsourcing for disaster management using SMS and Twitter: A research prototype. In Proceedings of the 2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops), Sydney, Australia, 14–18 March 2016.
20. Ogie, R.I.; Clarke, R.J.; Forehead, H.; Perez, P. Crowdsourced social media data for disaster management: Lessons from the PetaJakarta.org project. *Comput. Environ. Urban Syst.* **2019**, *73*, 108–117. [[CrossRef](#)]
21. Murali, S.; Krishnapriya, V.; Thomas, A. Crowdsourcing for disaster relief: A multi-platform model. In Proceedings of the 2016 IEEE Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER), Mangalore, India, 13–14 August 2016.
22. Besaleva, L.I.; Weaver, A.C. *Applications of Social Networks and Crowdsourcing for Disaster Management Improvement*; IEEE: Alexandria, VA, USA, 2013.
23. Zwikael, O.; Unger-Aviram, E. HRM in project groups: The effect of project duration on team development effectiveness. *Int. J. Proj. Manag.* **2010**, *28*, 413–421. [[CrossRef](#)]
24. E Silva, L.C.; Costa, A.P.C.S. Decision model for allocating human resources in information system projects. *Int. J. Proj. Manag.* **2013**, *31*, 100–108. [[CrossRef](#)]
25. Mota, C.M.M.; Almeida, A.T.; Alencar, L.H. A multiple criteria decision model for assigning priorities to activities in project management. *Int. J. Proj. Manag.* **2009**, *27*, 175–181. [[CrossRef](#)]
26. Daher, S.F.D.; Almeida, A.T. Recent patents using group decision support systems: A short review. *Recent Pat. Comput. Sci.* **2010**, *3*, 81–90. [[CrossRef](#)]
27. Kang, D.; Jung, J.; Bae, D.-H. Constraint-based human resource allocation in software projects. *Softw. Pract. Exp.* **2011**, *41*, 551–577. [[CrossRef](#)]
28. Otero, L.D.; Centeno, G.; Ruiz-Torres, A.J.; Otero, C.E. A systematic approach for resource allocation in software projects. *Comput. Ind. Eng.* **2008**, *56*, 1333–1339. [[CrossRef](#)]
29. Donyina, A.; Heckel, R. Modelling Flexible Human Resource Allocation by Stochastic Graph. In Proceedings of the Fifth International Conference on Graph Transformation, Enschede, The Netherlands, 27 September–2 October 2010.
30. Jin, G.; Sperandio, S.; Girard, P. Management of the design process: Human resource allocation in factories of the future. *Insight* **2017**, *20*, 19–23. [[CrossRef](#)]
31. Goldratt, E.M. *Critical Chain*; North River Press: Great Barrington, MA, USA, 1997.
32. Sutherland, J. Future of scrum: Parallel pipelining of sprints in complex projects. In Proceedings of the Agile Development Conference, Washington, DC, USA, 24–29 July 2005.
33. Laslo, Z.; Goldberg, A. Resource allocation under uncertainty in a multi-project matrix environment: Is organizational conflict inevitable? *Int. J. Proj. Manag.* **2008**, *26*, 773–778. [[CrossRef](#)]
34. Aritua, B.; Smith, N.; Bower, D. Construction client multi-projects—A complex adaptive systems perspective. *Int. J. Proj. Manag.* **2009**, *27*, 72–79. [[CrossRef](#)]
35. Kwak, Y.H.; Anbari, F.T. Analyzing project management research: Perspectives from top management journals. *Int. J. Proj. Manag.* **2009**, *27*, 435–446. [[CrossRef](#)]

36. Ponstien, A.; Kusters, R.J. Classification of Human- and Automated Resource Allocation Approaches in Multi-Project Management. *Procedia Soc. Behav. Sci.* **2015**, *194*, 165–173. [CrossRef]
37. Brown, K.L. *Resource Allocation in Multi Agent System*; Stanford University: Stanford, CA, USA, 2003.
38. Heppenstal, A.; Crooks, A. Guest editorial for spatial agent-based models: Current practices and future trends. *GeoInformatica* **2019**, *23*, 163–167. [CrossRef]
39. Wikipedia. Smart Agent Technologies. Wikipedia. 2019. Available online: https://en.wikipedia.org/wiki/Smart_Agent_Technologies (accessed on 4 November 2019).
40. Allen, J. Using network segments in the visualization of urban isochrones. *Int. J. Geogr. Inf. Geovisualization* **2019**, *53*, 264–270. [CrossRef]
41. Thrift, N.; Pred, A. A new beginning: Progress in Human Geography. *Time Geogr.* **1981**, *5*, 277–286.
42. Hägerstrand, T.; Carlstein, T. The two vistas. *Geografiska Annaler: Series. Hum. Geogr.* **2004**, *86*, 315–323.
43. Hecht, R.; Herold, H.; Behnisch, M.; Jehling, M. Mapping Long-Term Dynamics of Population and Dwellings Based on a Multi-Temporal Analysis of Urban Morphologies. *ISPRS Int. J. Geo Inf.* **2019**, *8*, 2. [CrossRef]
44. Kang, K.; Ponomarev, A.; Zerkal, O.; Huang, S.; Lin, Q. Shallow Landslide Susceptibility Mapping in Sochi Ski-Jump Area Using GIS and Numerical Modelling. *ISPRS Int. J. Geo Inf.* **2019**, *8*, 148. [CrossRef]
45. Albano, R.; Sole, A. Geospatial Methods and Tools for Natural Risk Management and Communications. *ISPRS Int. J. Geo Inf.* **2018**, *7*, 470. [CrossRef]
46. Peterson, A. *The Origin–Destination Matrix Estimation Problem*; Linköpings Universitet: Linköpings, Sweden, 2007.
47. ESRI. *ArcToolbox Manual*; ESRI: Redlands, CA, USA, 2018.
48. Stéphane, R. Modular Programming Versus Object Oriented Programming. 2010. Available online: <http://petesqbsite.com/sections/express/issue17/modularversusoop.html>. (accessed on 4 November 2019).
49. Gosling, J.; Joy, B.; Steele, G.; Bracha, G. *The Java Language Specification*; Addison-Wesley: Boston, MA, USA, 2005.
50. Petau, P. *On the Benefits and Problems of Object Oriented Paradigm*; Helsingforce, Swedish School of Economics: Helsinki, Finland, 2005.
51. Webopedia. CRM Software Customer Relationship Management Software. 2018. Available online: https://www.webopedia.com/TERM/C/crm_software.html (accessed on 6 February 2019).
52. Griffin, T.; Trickey, H. Integrity Maintenance in a Telecommunications Switch. *Spec. Issue Database Constraint Manag.* **1994**, *17*, 43–46.
53. Toms, S. *ArcPy and ArcGIS Geospatial Analysis with Python*; Kindle, Ed.; Packt: Birmingham, UK, 2015.
54. Wikipedia. Anylogic. 2018. Available online: <https://en.wikipedia.org/wiki/AnyLogic> (accessed on 27 November 2018).
55. SIMIO. Simio Sync. Simio. 2019. Available online: <https://www.simio.com/index.php> (accessed on 18 March 2019).



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).