ERAM
Evacuation Routing using Ant Colony Optimization over Mobile Ad Hoc Networks

Alejandro Avilés del Moral[1], Munehiro Takimoto [1] and Yasushi Kambayashi [2]
[1]*Department of Information Sciences, Tokyo University of Science, Tokyo, Japan*
[2]*Department of Computer & Information Engineering, Nippon Institute of Technology, Saitama, Japan*

Keywords: Ant Colony Optimization, Global Positioning System, Mobile Ad Hoc Networks, Mobile Agents, Multiagent Systems.

Abstract: This paper proposes a distributed multi-agent framework for discovering and optimizing evacuation routes on demand. We name it Evacuation Routing using Ant Colony Optimization over Mobile Ad hoc Networks (ERAM). Taking advantage of ant colony optimization (ACO) on mobile ad hoc networks (MANETs) composed of smartphones with geo-location capabilities, ERAM aims for adaptability and layout independence, relying exclusively on crowd's knowledge during mass evacuations. Such knowledge is inserted into the system: actively, by users' indication of having reached safe areas on their smartphones; and passively, by smartphones tracking their own movement. In the framework, agents migrate through nodes of a MANET towards safe areas based on an indirect communication mechanism called *stigmergy*, which is a behaviour that social insects show. Once an agent finds such an area, it traces its path backwardly collecting geographical information of intermediate nodes for composing an evacuation route. During the backward travel, agents lay pheromone down while they migrate back based on the ACO algorithm, strengthening quasi-optimal physical routes, and hence guiding succeeding agents. This scenario is analogous to data-packet routing on Internet or resource discovery on P2P networks, except it routes people through physical environments towards safe areas instead.

1 INTRODUCTION

In cases of emergency, people in a building, facility or even whole cities might need to be evacuated. Known routes, however, found to be congested or compromised can lead crowds to panic due to occasional lack of indications to reach a safe area. Therefore, it is desirable to have an infrastructure-less adaptable framework to assist finding routes on demand without any previous specific preparation of roads or buildings in order to increase the survival rate.

We apply swarm intelligence based routing mechanisms that are well known to be useful in virtual environment as shown in (Ducatelle et al., 2010); (Ducatelle, 2007) and (De Rango and Socievole, 2011) for crowd flow guidance in the real world through handheld devices such as smartphones.

Nowadays, smartphones are increasingly relevant in daily life among all kinds of users. This incessant growth is derived from some features such as portability, which traditional computers lack. This feature as well as relatively small size can be applied to solve old problems in new ways. In particular, there are two key features for the implementation of our framework. Whereas the smartphone's wireless connectivity allows the deployment of a swarm intelligent system, the built-in Global Positioning System (GPS) reception can provide information about the physical environment.

The combination of these technologies on a single portable device along with swarm mechanisms assists to find not only main evacuation routes but also alternative ones, while taking in account of the congestion in the routes. The found routes can reflect unpredictable human behavior through dynamically supporting safety policies (Proulx, 2001).

We propose a distributed multi-agent distributed framework for discovering optimized evacuation routes on demand. We henceforth call this framework Evacuation Routing Using Ant Colony Optimization over Mobile Ad hoc Networks

Avilés del Moral A., Takimoto M. and Kambayashi Y..
ERAM - Evacuation Routing using Ant Colony Optimization over Mobile Ad Hoc Networks.
DOI: 10.5220/0004261301180127
In *Proceedings of the 5th International Conference on Agents and Artificial Intelligence* (ICAART-2013), pages 118-127
ISBN: 978-989-8565-38-9

(ERAM). This framework came from AntHocNet (Di Caro et al., 2004), (Di Caro, Ducatelle, & Gambardella, 2005) and (Di Caro et al., 2005), a framework for routing in Mobile Ad hoc Networks (MANETs) and the algorithm for resource discovery in P2P networks presented in (Kambayashi and Harada, 2007) and (Kambayashi and Harada, 2009). Both approaches use Ant Colony Optimization (ACO) providing indirect communication between agents in order to achieve self-organized optimization.

ACO is a biologically inspired framework based on ant foraging behaviors, as proposed in (Colorni, Dorigo, & Maniezzo, 1992) and (Dorigo and Gambardella, 1997). Ant agents laying pheromone down reinforces paths that are more likely to lead towards fruitful nodes. This way, pheromone level indicates the goodness of a node regarding the goal, and by constantly being modified by ant agents, the system get quasi-optimized in run-time. This indirect method is called *stigmergy*.

The structure of the balance of this paper is as follows. In the second section, we describe an evacuation scenario in which ERAM performs the guidance for the users. The third section describes ERAM framework. ERAM consists of several static and mobile agents. The static agents interact with users and store routing information, and the mobile agents effectively work together with *stigmergy* to find optimal evacuation routes. The fourth section demonstrates the usefulness of the framework with the results of numerical experiments on a simulator we built. Finally, the fifth section discusses conclusion and future works.

2 SCENARIO

ERAM presents a novel hybrid solution for network-physical environment routing, a field that has not investigated so far. In contrast with other swarm intelligence based routing frameworks such as the ones proposed in (Kambayashi and Harada, 2007) and (Ziane and Melouk, 2005), which basically focus on data packet routing over networks, ERAM aims to discover physical resources over physical layouts instead of virtual ones. ERAM, however, uses networks as a means to locate those resources, and thus previous researches on ACO routing provide a robust background.

2.1 Scenario Description

As derived from the previous section, in an evacuation scenario resources correspond to physical safe areas that users proactively mark on their smartphones as soon as they are reached, each referred to as node. Paths discovered with ERAM take into account physical layouts. The GPS is used to provide geographic information that will be collected on ERAM as described later. This information is also used to evaluate the congestion of crowd flows and to optimize them.

In order to perform resource discovery and routing, active and passive sources of information are required. Whereas users provide active input when they reach safe areas, smartphones provide passive input by constantly storing GPS tracks.

Finally, optimal routing discovery both in network and in physical environments is achieved by indirect communication through migrations of mobile agents over nodes of a MANET applying the ACO algorithm described in the next section.

In order to make the scenario mentioned above technically feasible, we assume that each node has the following functionalities:

- Built-in Wi-Fi connectivity to construct the MANET with other smartphones.
- Making use of decent GPS signal.

2.2 Scenario Difficulties

A real implementation of ERAM framework, however, implies some difficulties mainly related with the reliability of the information.

One problem, which is easier to handle, is that of relying on human knowledge in a stressful situation to obtain the routes. To deal with this, the framework assumes that, at least, a few people are able to reach a safe area, i.e. shelter, out of the building, etc. and mark it on their smartphone. Then, by the unsupervised usage of the pheromone, those nodes closer to these areas will be more likely to lead people to a secure one.

Another problem, not yet fully addressed in this paper, is the impossibility to avoid fake marking of safe areas. A proposal of solution is weighting safe flags in relation of its geographic density based on the fact that the majority of people will not lie. In this manner, the more safe flags of which nodes are close to, the more likely they are to attract mobile agents, and thus, to produce correct evacuation routes avoiding those leading to fake safe areas. Another way to enforce reliability is a distributed trust protocol, as proposed in (Nakamoto, 2008), which makes almost impossible to fake the system.

Additionally, GPS signal is maybe the most troublesome aspect of the framework. It is hardly available indoors and obstacles like trees or buildings outdoors can easily hinder precision. In order to obtain precise GPS samples, there are some solutions widespread at the moment. One of them is the usage of differential GPS, which relies on static devices with known position, if available. Another one is using external sensors or measurement of Wi-Fi signal to estimate absolute position like the one proposed in (Woodman and Harle, 2008), but this relies on infrastructure or preloaded information. There are also more innovative approaches such as (Bejuri et al., 2011), which propose a taxonomic GPS that would provide precise and ubiquitous positioning.

3 ERAM FRAMEWORK

In this section, ERAM framework is described in detail. It first covers the agents that are involved in the consecution of the goal, conditions of the algorithm to guarantee the negotiability of the evacuation route, and the formal definition of the pheromone calculation.

3.1 Agents

Due to the fact that similarities found in formulations of ERAM's scenario with the one proposed in (Kambayashi and Harada, 2007), the solutions are also similar. Especially, its concept of a set of collaborative static and mobile agents for discovering resources in P2P networks has also been adapted to ERAM. This concept is required to deal with the complexity and uncertainty of what hybrid network-physical scenario presents.

Fig. 1 depicts the different kinds of agents participating in the framework. The following descriptions are brief explanation of their tasks.

1. Information Agent (IA): it is a static agent resides on each node. It is in charge of perceiving physical context as well as storing knowledge about its hosting node. It also acts as an interface with the user, receiving input and providing directions to follow.
2. Node management Agent (NA): it is a static agent also residing on each node. It is in charge of storing knowledge of the network context, namely storing information about other nodes.
3. Goal Agent (GA): it is a mobile agent that floods the network with information about the

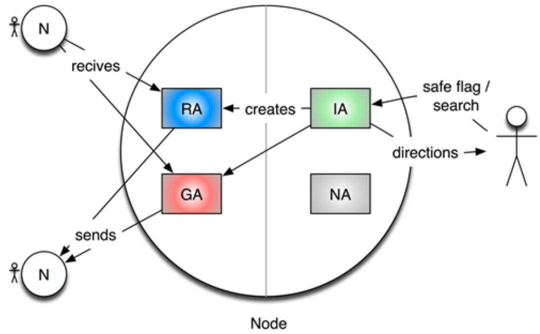

Figure 1: ERAM participators. Nodes host mobile agents (RAs and GAs), and static agents (IA and NA). Users communicate with IA. IA creates RAs and GAs. Mobile agents migrate towards other nodes.

safe area recently reached. An IA on a safe node creates GA.
4. Routing Agent (RA): it is a mobile agent that tries to find safe nodes in order to obtain evacuation routes. IA on a node in search of evacuation routes, which is called swarm node, creates RA.

The interactions between them can be illustrated as shown in Figure 2. We describe the details of each agent below.

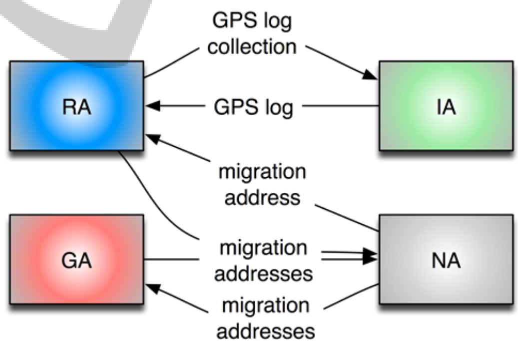

Figure 2: Interactions between agents. Here, IA provides the node GPS log and receives GPS log collections for calculations; NA provides IP address to mobile agents, and both RAs and GAs updates pheromone value on the NA.

3.1.1 Information Agent (IA)

Each node hosts an IA and it keeps track of the GPS log by taking position samples periodically. Whenever another one node comes within Wi-Fi range of the node on which the IA resides, it starts the handshaking protocol in order to initiate session. This can be achieved as shown in SLS (Gajurel and Heiferling, 2009), but storing the whole GPS log instead of a single position. Adjacent connected nodes constantly exchange GPS logs between IAs and all received information is stored in the NA. We describe NA in the next section.

User can interact with the IA by activating the safe flag or by activating the evacuation search. These two modes are mutually exclusive.

- Safe flagged: When a user reaches a safe area, he or she hoists the safe flag in his or her smartphone. Once the safe flag is activated, the IA starts creating GAs periodically in order to flood the network with information about the safe area position.
- Searching evacuation: When a user is in need of evacuation route, he or she sets his or her smartphone in search mode. In this process, the IA produces RAs constantly, making them migrate to search negotiable paths towards safe areas. Once one RA returns from a safe flagged node storing GPS logs of intermediate nodes, the IA composes an evacuation route from them. If the estimated time of the new evacuation route is better, the current one is replaced by it. Finally, the route is provided to the user in a form of directions to be followed.

As pointed in (Rodriguez and Amato, 2010) and (Rodriguez and Amato, 2011), routes composed of pairs of positions should be smoothened to reduce unnecessary loops.

3.1.2 Node Management Agent (NA)

In addition to an IA, every node hosts an NA. The NA keeps information about neighbour nodes such as IP address, GPS log and other data used for pheromone calculation. Each entry is created when the IA successfully initiates session with neighbour nodes, which are updated as mobile agents arrive and give new information.

Table 1: Knowledge encoded within an NA.

IP:port	GPS log	Jumps	SAV	GA id
9.163.251.45:50000	{(3,4), ...}	4	0.8	2435
92.71.112.26:50001	{(0,1), ...}	1	0.1	8431
…	…	…	…	…

As shown by Table 1, an NA encodes the basic knowledge in its matrix to cover the following fields.

- Jumps: the number of hops from a safe flagged node. It is updated by GAs.
- Speed Ahead Value (SAV): this encodes congestion level and it is modified by RAs. It is covered later in detail.
- GA id: the unique identification of the GA that last updated the entry. This is used for stopping GAs from migrating endlessly.

When mobile agents need to migrate, they request the NA to provide them the IP addresses of the nodes to which they migrate next.

3.1.3 Goal Agent (GA)

A safe flagged node's IA creates GAs in order to disseminate the information in the network about the safe area. The GA, therefore, requests the NA information of all the connected nodes, and clones itself on each of them. As the GA migrates through the network, it counts the number of hops it has performed. When it steps into a new node, this information is provided to the NA of that node in order to update the Jumps entry of the neighbouring nodes. This information is used for subsequent pheromone calculation.

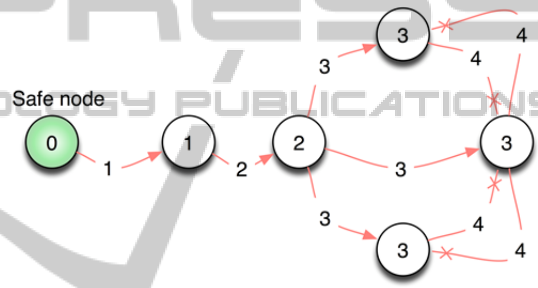

Figure 3: Migration activity of a GA through a simple ad hoc network. Numbers represent Jumps value. Arrows with an X at the end mean the GA is killed.

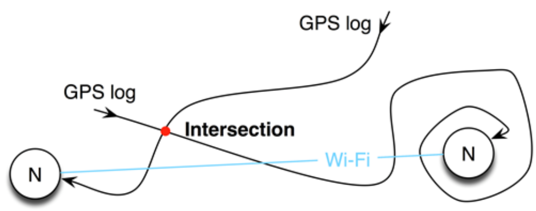

Figure 4: Condition for RA to perform a migration.

The terminating condition of GA's migration is as follows. When it is created, every GA has a unique identification. Thus, if it steps into a node with the same GA id as it has, it means that has been previously visited by one of the other clones of the GA, and therefore it is killed. GA's life span is graphically explained in Figure 3.

3.1.4 Routing Agent (RA)

A routing agent is an agent created by an IA on a node that searches for safe areas. The RA migrates while it satisfies the condition where a GPS log on a new node to which it is migrating has some intersections with the GPS log of the current node as shown in Figure 4. It means that there are some intersections between GPS logs such as a chain, and this is possible because the NA of the hosting node stores the GPS log of nodes surrounding that node.

This condition guarantees that nodes can physically reach the safe areas following paths derived from GPS logs without meeting any obstacles.

In order to optimize this evacuation route, the RA will choose the node with the highest pheromone value at present. As ACO states, this is merely probabilistic as explained in (Di Caro et al., 2006); therefore, there are situations in which random paths are chosen in order to explore other possible evacuation routes.

If a situation in which a sequence of connecting nodes meets the mentioned condition and it reaches a safe flagged node, the RA becomes a backward RA immediately. The backward RA follows the same path as the forward RA but in the opposite direction.

There may be some situations in which the forward RA hits node having an evacuation route already created. In this case the RA will copy the evacuation route and become a backward RA.

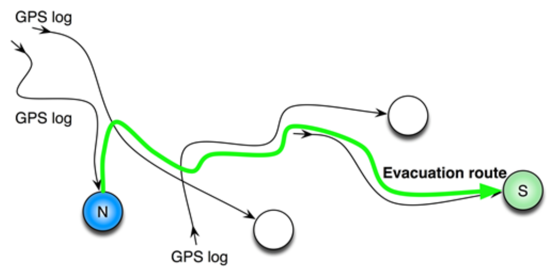

Figure 5: When GPS logs are not exactly intersecting but proximity can be used for calculating the evacuation route.

On its way back, the RA collects the GPS logs of the nodes while laying pheromone down in the NAs on the nodes. This pheromone, represented by Speed Ahead Value (SAV), reflects the congestion present along the path. When the RA hits the original node where it was created, the GPS logs are provided to the IA. Once this happens the evacuation route is established.

3.2 Route Construction

The route composed from GPS logs is expected to be an evacuation route because a chain of intersecting GPS logs ends in a safe flagged node, which is in a safe area.

This is the expected result of an algorithm used to compose the evacuation route, when intersecting GPS logs have been received. The composed route avoids loops and unnecessary paths by deleting those parts that do not lead towards a safe flagged node.

As depicted in Figure 5, sometimes the GPS logs may be not exactly intersecting but should be considered for having physical reachability. In such a case, fuzzy approaches provide flexibility for dealing with uncertainty as well as with constraints that accepts some degree of error.

In our system, instead of binary functions returning whether some intersections exist or not, a radial basis function is utilized to weaken intersection constraints. Notice here that such a radial basis function needs a predefined threshold of distance to be regarded as an intersection between GPS tracks.

3.3 ACO Algorithm

The following equation determines the pheromone level of a node.

$$\mathrm{pheromone(n)} = \frac{\mathrm{speed_n}}{\mathrm{jumps_n}} * \mathrm{SAV_n} \qquad (1)$$

$speed_n$ is the average speed of node *n*, using the GPS track stored in its IA. The length of the time window of the GPS track affects the sensibility of this parameter. This implies that the shorter the length of the window becomes, the fewer samples for calculating the parameter we get. That results in a highly responsive value, causing the last few seconds of the movement to be reflected more precisely.

$jumps_n$ reflects how many migrations are needed to reach the closest safe flagged node. Due to an agents wireless capabilities this is a heuristically determined parameter as it positively estimates how far the node is from a safe area. Even if agents can migrate through obstacles such as walls, human beings will not be able to pass through.

SAV_n stands for Speed Ahead Value. It serves the purpose of estimating congestion ahead. For SAV, having a lower value means that nodes ahead are moving more slowly, therefore the probability of congestion is higher and RAs will be less likely to follow these nodes while seeking a safe flagged node.

As described in (Kambayashi and Harada, 2007), our framework does not implement a pheromone decay system through evaporation. Instead, thanks to RA constantly migrating through the network, this value is frequently updated and serves the purpose of pheromone decay mechanism.

SAV of any safe node is 1. SAV of a non-safe given node *n* starts at 1 and it is recalculated based on: its SAV previous value and both the SAV and average speed from the node straight ahead. The RA carries that information. In this way, knowledge about congestion is propagated backwardly through the network while updating each node through iterations.

The formal calculation is as follows.

$$SAV(n) = w_1 SAV_n + w_2 SAV_{n-1} + w_3 \frac{speed_{n-1}}{speedNorm} \quad (2)$$

Where $0 < SAV \leq 1$, and

$$w_1 + w_2 + w_3 = 1$$

w_1, w_2 and w_3 are predefined weights that determine how SAV will be modified through iterations. Fine-tuning these three constants is a key to obtain a self-optimizing routing system.

- *w1*: represents how slowly SAV is modified through iterations. The higher the weight is, the slower SAV will be affected. Having values near to 1 means that it will hardly change, and near to 0 means that will be overridden every iteration.
- *w2*: represents how much overall congestion ahead from node *n* influences its SAV calculation. Having values near to 1 means that initial SAV will be carried all the way back.
- *w3*: represents how much the speed of the node straight ahead influences on the node SAV. Having values near to 1 means that SAV only represents congestion of the node straight ahead.

In addition, the variable *speedNorm* is a normalizer of speed that keeps the value between 0 and 1.

In summary, the pheromone level of a node is directly proportional to the average speed of movement. The pheromone level is inversely proportional to the number of migrations to the closest safe area. Thus, the pheromone level is penalized by distance from a safe area and congestion such as a pheromone decay mechanism.

4 RESULTS

In order to demonstrate the effectiveness of our framework, we have built a simulator and conducted numerical experiments. The results addressed in this section were acquired using a simulation environment that provides graphical representation of the framework and statistical results for further analysis. We call this simulation environment ERAMsim.

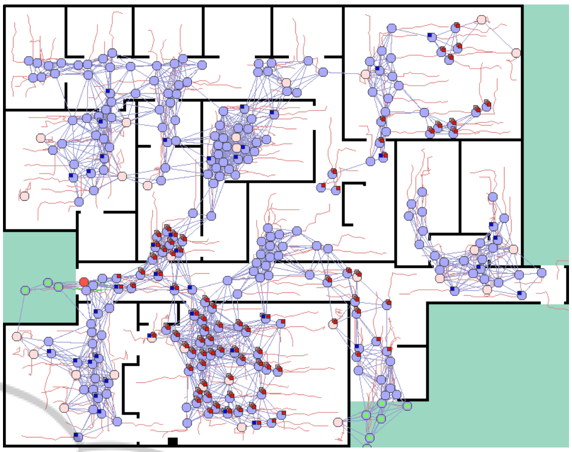

Figure 6: Simulation screenshot with 300 nodes (10% swarm). Blue nodes are informed nodes, pink nodes swarm nodes, and red nodes swarm nodes with evacuation route. Green grounds represents safe area, white ground floor within the building, and black impassable walls. It also shows Wi-Fi connections between nodes, GAs and RAs, and GPS logs in red trajectories.

First, we give a brief explanation of the simulator; second, we describe the simulation settings; and last, we discuss the simulation results.

4.1 Simulator Overview

ERAMsim is a software program developed for running basic simulations of ERAM framework. Providing simplified both physical and network environment simulations, it implements the ACO algorithm as well as the behaviour of individuals trying to reach a safe area in a floor plan, as depicted in Figure 6.

It can be perceived that white ground represents the interior of a building whereas green grounds are safe areas outside the building and black lines are walls. Consequently, the basic unit for space is the pixel, and the time is discrete.

The simulator takes account of physical constraints such as maximum density of people or the impossibility of them to pass through others, producing bottlenecks in doors. It also assumes that there are some nodes that already know where to go, referred as to informed nodes, and others using ERAM exclusively, swarm nodes. As soon as any of them reaches a safe area it becomes a safe node, starting to produce GAs, and then walking away from the building.

4.2 Simulation Settings

Table 2 shows the default values of the main parameters used for running the simulator.

These parameters are clarified as follows:

- 'Total nodes' determines how many individuals-smartphone are participating in the evacuation. 200 nodes provide a density of 4.71%.
- 'Swarm nodes' configures the percentage of the total nodes using ERAM to find their way.
- 'GPS log length' refers to the maximum amount of GPS samples that are stored within a NA with a frequency of 0.25 samples/round.
- 'Wi-Fi range' determines the range, in pixels, of the Wi-Fi signal of the smartphones.
- 'ACO random' determines the chance of a RA to migrate randomly instead of following the best pheromone value.
- $w_{1\sim3}$ are the weights for SAV calculation.
- 'GA/RA cooldowns are the number of rounds that takes for a node to produce the next mobile agent.

Table 2: Parameters default values.

Parameter	Value	Parameter	Value
Total nodes	200	w_1	0.1
Swarm nodes	10% nodes	w_2	0.5
GPS log length	100 samples	w_3	0.4
Wi-Fi range	70 pixels	GA cooldown	50 rounds
ACO random	10%	RA cooldown	5 rounds

We use discrete time, a floor plan with 800px*600px, and nodes represented by circumferences of 6px radius moving at a maximum speed of 2px/round. Approximating these values to meters, nodes are 70cm diameter circumferences, from which it can be derived that a pixel is 5.83cms and they can move up to 1.16 m/s in a 46.7m x 35m physical environment.

Finally, default values have been chosen for the purpose of keeping the performance metric low, which enables the behavior to be easily analyzed through modifying them.

4.3 Simulation Results

The results focus on the success rate of finding the evacuation route rather than on the overall route optimization. This is due to the fact that for measuring the improvement, comparison between virtual evacuation time using ERAM and real evacuation time is required, and in order to obtain the real one, sophisticated human behaviour simulation is needed.

In these results, the performance is measured as the success rate of finding evacuation route in terms of the previously defined variables previously defined. Assuming that all swarm nodes that successfully reach a safe area are caused by the usage of ERAM, success rate of the route discovery algorithm can be formally defined as follows.

$$\text{success rate} = \frac{\text{swarm nodes saved}}{\text{swarm nodes}} \qquad (3)$$

Because of the probabilistic nature of the simulator, in order to get reasonable results, each sample reflects the average of five simulations using the same parameter values.

4.3.1 GPS Log Length

When configuring a real implementation of ERAM, one of the main parameters is the GPS log length. Every node needs to exchange this information constantly with connected nodes in order to reflect lag-free geographic information. This data exchange is critical for the framework but it is also the most bandwidth consuming, and thus, it is crucial to size it optimally.

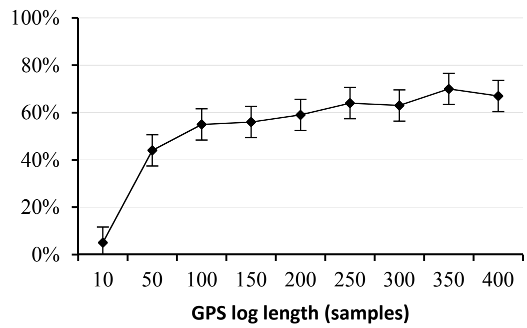

Figure 7: Success rate depending on GPS log length.

As can be seen in Figure 7, the longer the length of the GPS log is, the higher the success rate of route discovery becomes. This is caused by the fact that, it is more probable to find intersections between GPS logs, and consequently for the system to allow more agent migrations. By easing agent migration, route discovery of swam nodes increases and so does the success rate.

Nevertheless, this metric is mainly bounded by two facts, both responsible of the plateau at around 70% success rate in Figure 7. The first one is related to a physical constraint and is impossible to avoid. It is that regardless the GPS log length there might be nodes located where others have never come, making any GPS intersection nonexistent. The second one has to do with technical limitations and is interesting as can be predicted more easily. It is that although there are GPS intersections, if the Wi-Fi signal is much shorter than the GPS log, nodes cannot be aware of it, and therefore migrations will not take place.

4.3.2 Wi-Fi Signal

Studying how Wi-Fi signal influences the success rate is interesting, because it not only determines the bounding the GPS log length, but also it eventually provides minimum technical requirements for the ERAM framework to work properly.

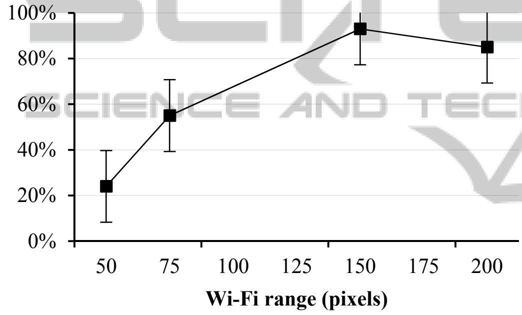

Figure 8: Success rate depending on Wi-Fi range.

Figure 8 shows an improvement in success rate similar to the one exposed for the GPS log length. Again, by increasing the Wi-Fi signal more nodes will be connected in the MANET, and this way, more GPS log intersections will be found by agents.

On the other hand, the GPS log length also bounds the improvement of Wi-Fi signal. Regardless the range of the signal connecting nodes, technically allowing agents to migrate, when GPS logs are not long enough, intersections will hardly occur and then migration constraint will not be satisfied.

With this simulation configuration, having a Wi-Fi signal 1.5 times wider than the GPS log length produces a success rate of 93%.

4.3.3 Density and Swarm Nodes

Finally, we discuss the interest of using ERAM depending on the density of people in the evacuation and how the framework performs when only a few of them know the way to a safe zone.

Note that swarm nodes in this simulation walk towards safe zones only when they have obtained an evacuation route through ERAM. Hence measurements do not concern about 0% or 100% swarm nodes scenarios. Having none swarm node would mean there is nothing to measure, and having all swarm nodes would mean no human knowledge about safe areas and therefore success rate equal to zero.

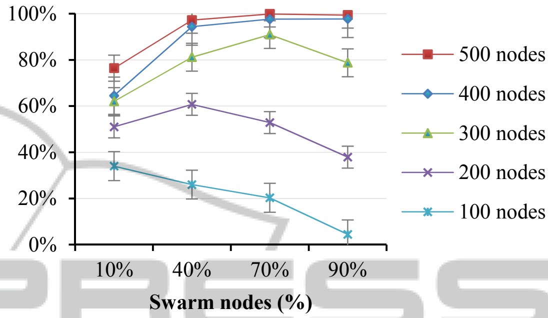

Figure 9: Success rate depending on density and percentage of swarm nodes.

As shown in Figure 9, as density grows the overall success rate increase, highlighting that ERAM presents superior performance for mass evacuations. Given the map the simulation uses, 100 nodes mean a density of 2.36%, and 500 nodes 11.78%.

This improvement is derived from two main facts: first of all, the topology of the MANET mesh is more interconnected; and secondly, more GPS logs stored in the system leads to more GPS intersections.

Another dimension that Figure 9 explores is the impact of swarm nodes percentage. As can be observed, as the number of swarm nodes increases the success rate decreases in low-density settings. This can be explained because the more swarm nodes participating in the evacuation, the less informed nodes, and thus, only a few safe nodes will be produced. In addition, having low densities makes the MANET connectivity too reduced, producing some isolated sub-MANETs with fewer chances to transmit agents towards one of those scarce safe nodes.

When density increases, however, and consequently so does MANET connectivity, RAs laying pheromone down makes, even for highly isolated MANETs, the success rate of this migrations increase up to almost 100% of swarm nodes. Eventually this makes almost all the swarm nodes reach safe areas even when only a few nodes know the way to the safe area.

These results indicate that encouraging the active usage of ERAM (swarm nodes) for evacuations in scenarios such as public buildings, sport events, or cities will provide good results.

It should be clear that this metric is extremely pessimistic due to the simplistic human behavior simulation. As in a real situation a swarm node can become an informed one asking other people, following signs or by visual contact with the safe area.

5 CONCLUSIONS AND FURTHER WORKS

ERAM represents a novel usage of ACO for hybrid environments. Routing people by the passive and active usage of the smartphone opens a new horizon of study. This project provides not only the first formal definition of the framework but also a software tool to test it.

A great amount of uncertainty and imprecision will be present in a real scenario because of the usage of commercial GPS, and unsuitable conditions for geo-location. The simulation results, however, indicate that the direction of our investigation is right. It is now clear that smartphone and ACO are a suitable combination to develop the new safe evacuations systems.

Our experiments on the simulator so far show promising results regarding the evacuation route discovery algorithm, especially for high-density evacuation. This first iteration of ERAM is, however, the initial stage, and still far from achieving the optimal evacuation routing. Further works can be summarized in three main topics: framework refinement, simulator realism, and extensive experimentation.

In regard of framework refinement, first, pheromone calculation of the ACO algorithm needs to find the proper values for weights. One approach to solve this while at the same time making it even more flexible should be applying some sort of neural network to dynamically adapt the weights to the different type of context, i.e. pedestrian or car evacuations. It is also interesting incorporate into the framework solutions for scenario difficulties as described in section 2.

ERAMsim needs further works in order to demonstrate the feasibility of the framework in different scenarios under much more complex environments. For doing so, it will be necessary to adopt more realistic simulation approaches. On the one hand, a realistic network simulation would allow measuring the congestion of agents' interactions produce as well as obtaining the actual time it takes to find an evacuation route. On the other hand, utilizing physical environment simulations as shown in (Rodriguez and Amato, 2010) and (Rodriguez and Amato, 2011) would allow to measure how route optimizing routing improves survival rate.

The simulator, although must be improved, is good enough to show a current state and to explore the limits of this framework by retrieving results beyond the ones shown in this paper.

Concluding, and regarding the feasibility of a future real implementation, we believe that we have enough technological components to implement ERAM. The usage of smartphones is being widespread. We are trying to implement an ad hoc network using solely smartphones. Upon completion of such network, it is possible to have a mobile agent system without explicit network infrastructures. It is also possible to have alternative location system other than GPS. For example, it should be relatively easy to construct infrastructure using RFID rather than GPS that provides the precision needed to provide dynamic routing in controlled environments such as museums for providing new features besides evacuation routing.

REFERENCES

Bejuri, W. M., Mohamad, M. M., & Sapri, M. (2011). Ubiquitous Positioning: A Taxonomy for Location Determination on Mobile Navigation System. *International Journal of Signal & Image Processing , 2* (1), 24-34.

Colorni, A., Dorigo, M., & Maniezzo, V. (1992). "Distributed Optimization by Ant Colonies". *Proc. 1st European Conference on Artificial Life*, (pp. 132-142). Cambridge.

De Rango, F., & Socievole, A. (2011). "Meta-Heuristics Techniques and Swarm Intelligence in Mobile Ad Hoc Networks". In *Mobile Ad-hoc Networks: Applications.*

Di Caro, G., Ducatelle, F., & Gambardella, L. (2006). "An analysis of the different components of the AntHocNet routing algorithm". *Proc. 5th International Workshop on Ant Colony Optimization and Swarm Intelligence (ANTS2006).* Brussels.

Di Caro, G., Ducatelle, F., & Gambardella, L. (2005). "AntHocNet: An adaptive nature-inspired algorithm for routing in mobile ad hoc networks". *EUROPEAN TRANSACTIONS ON TELECOMMUNICATIONS*, *16*, pp. 443-455.

Di Caro, G., Ducatelle, F., & Gambardella, L. (2004). "AntHocNet: an Ant-Based Hybrid Routing Algorithm for Mobile Ad Hoc Networks". *Proc. 8th International*

Conference on Parallel Problem Solving from Nature (PPSN VIII). Birmingham,.

Di Caro, G., Ducatelle, F., & Gambardella, L. (2005). "Using Ant Agents to Combine Reactive and Proactive Strategies for Routing in Mobile Ad Hoc Networks". *International Journal of Computational Intelligence and Applications (IJCIA) , 5* (2), 169-184.

Dorigo, M., & Gambardella, L. (1997). "Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem". *IEEE Transactions on Evolutionary Computation*, *1*, pp. 53-66. Brussels.

Ducatelle, F. (2007). *"Adaptive Routing in Ad Hoc Wireless Multi-hop Networks"*. PhD Thesis, Università della Svizzera Italiana, Istituto Dalle Molle di Studi sull'Intelligenza Artificiale, Lugano.

Ducatelle, F., Di Caro, G., & Gambardella, L. (2010). "Principles and applications of swarm intelligence for adaptive routing in telecommunications networks". *Swarm Intelligence , 4* (3).

Gajurel, S., & Heiferling, M. (2009). "A Distributed Location Service for MANET Using Swarm Intelligence". *Proc. IEEE Mobile WiMAX Symposium*, (pp. 220-225).

Kambayashi, Y., & Harada, Y. (2007). "A Resource Discovery Method Based on Multi-agents in P2P Systems". *Proc. 1st KES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications*, (pp. 364-374). Verlag.

Kambayashi, Y., & Harada, Y. (2009). "Integrating Ant Colony Optimization in a mobile-agent based resource discovery algorithm". *Proceedings of the IADIS International Conference on Intelligent Systems and Agents* (pp. 149-158). Antonio Palma dos Reis.

Nakamoto, S. (2008). *"Bitcoin: A Peer-to-Peer Electronic Cash System"*. From Bitcoin: http://bitcoin.org/bitcoin.pdf

Proulx, G. (2001). "Occupant Behaviour and Evacuation". *Proc. 9th International Fire Protection Symposium*, (pp. 25-26).

Rodriguez, S., & Amato, N. (2010). "Behavior-based evacuation planning". *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, (pp. 350-355). Anchorage.

Rodriguez, S., & Amato, N. (2011). "Utilizing Roadmaps in Evacuation Planning". *Proc. Computer Animation and Social Agents*, (pp. 67-73).

Woodman, O., & Harle, R. (2008). Pedestrian localisation for indoor environments. *Proc. of the 10th international conference on Ubiquitous Computing* (pp. 114-123). New York: ACM.

Ziane, S., & Melouk, A. (2005). "A swarm intelligent multi-path routing for multimedia traffic over mobile ad hoc networks". *Proc. Q2SWinet*, (pp. 55-62). New York.