Ant Colony Optimisation for Planning Safe Escape Routes

Morten Goodwin, Ole-Christoffer Granmo,
Jaziar Radianti, Parvaneh Sarshar, and Sondre Glimsdal

Department of ICT, University of Agder,
Grimstad, Norway

Abstract. An emergency requiring evacuation is a chaotic event filled with uncertainties both for the people affected and rescuers. The evacuees are often left to themselves for navigation to the escape area. The chaotic situation increases when a predefined escape route is blocked by a hazard, and there is a need to re-think which escape route is safest.

This paper addresses automatically finding the safest escape route in emergency situations in large buildings or ships with imperfect knowledge of the hazards. The proposed solution, based on Ant Colony Optimisation, suggests a near optimal escape plan for every affected person — considering both dynamic spread of hazards and congestion avoidance.

The solution can be used both on an individual bases, such as from a personal smart phone of one of the evacuees, or from a remote location by emergency personnel trying to assist large groups.

1 Introduction

Evacuation planning is challenging due to the chaotic and un-organised situation occurring in a crisis situation. Unfortunately, decision makers often have an incomplete picture of hazards and potential escape routes. The situation is further complicated by the fact that people affected are often left alone without any contact with rescue personnel. Further, the chaotic nature of crises situations causes the best escape route to quickly change as hazards such as fires quickly develop.

There is no doubt that decision making in crises situations needs to be made in a timely manner to minimize the potential danger. However, it is difficult people affected to determine what are the best decisions in an evacuation situation. In fact, in most situations the evacuees are not aware of which path to follow for an escape because they either received insufficient information from the rescuers or are unfamiliar with both the architecture and of where the hazards are located. Similarly, the emergency personnel often do not have an overview of where people are located nor which rooms are affected by hazards. This makes evacuation planning particularly difficult [1] [2].

This paper is part of a larger project working on using smart phone technologies in emergency situations. In this project, the smart phones are used to communicate to and from people affected by crises situations. This enables

rescue personnel to get an overview of people escaping and how people move and communication on how to best escape can be given to the people escaping. In addition, the sensor information available in smart phones will communicate presence of hazards using the camera, position using GPS, gyroscope etc. This provides a threat map which is support both for rescue personnel and to automatically determine the best escape routes [3].

The project has three main steps:

1. Collect information.
2. Calculate escape plan.
3. Communicate the plan to the people affected

1.1   Collect Information

Initially in a crisis situation, it is essential to get an overview of the people affected and the hazards present.

The aim is that prior to a crisis situation, for example when people are embark on a cruise ship, they will have the option to download a mobile application for their phone. This application will in a crisis situation utilize the available sensors and communicate it to a central location. This way, the system, including emergency personnel, will be aware of locations of people, whether people are moving, the brightness in each room (indicating hazards such as fire or smoke).

1.2   Determine Plan

Computer and mathematical models have shown to be valuable for escape planning with large complex building with many people [4] [5] [6] [7] [8], but is mainly assuming a static representation of hazards. In contrast, our system will, based on available information, calculate the best escape plan and guide each affected person away from any potential hazard, as well as distributing the people to the most suited escape areas even when the hazards change.

1.3   Communicate Plan

When an adaptive plan is available, it should be communicated to the affected people, which can be done in two main ways. The primary method is for emergency personnel to actively communicate the plan to the affected people through any available means, such as loud speakers and communication directly to each affected person via the smart phone applications. Failing this, the smart phone applications can automatically present the plan using simple visual and verbal steps such as "turn around", "go left" and safely guide people to an escape area.

Unfortunately, even in situations where an optimal escape plan exists and every person affected are aware of the plan, the human mind is so perplexed that not all follow the plan [9]. Most significantly, in a crisis situation factors such as panic spread, people pushing, jamming up and overlooking alternative exits prevent a crowd from following an optimal plan [10] [11]. Therefore, it is important that information about both hazards and people are continuously updated to always provide the best plan.

2   Problem Formulation

Escape planning from a complex building or a large ship can be regarded as a combinatorial optimization problem. In line with common practice [6], we treat the escape as a bidirectional planar graph $G(V, E)$. Each possible location i is connected with a vertex $v_i \in V$, and each potential flow from vertex v_i to v_j is represented by an edge $e_{i,j} \in E$.

In addition we define a function $h(v_i)$ representing the hazard for v_i, so that the function $h(v_i)$ returns probability values representing the likelihood of hazards.

The escape area is a vertex $v_t \in V$ (sink), and the people are located in any vertex $v_s \in V$ (any vertex is a source). Further, all search spaces from v_s to v_t is defined as **S**.

The aim of the application is to find a search space $s* \in \mathbf{S}$ so that $f(s^*) \leq f(s) \forall_s \in \mathbf{S}$, where $f(s) = 1 - \Pi_{V_i \in s}(1 - h(v_i))$. I.e. minimizing the probability that a person encounters a hazard in at least one of the vertexes in the chosen search space.

2.1   Hazards

The hazard functions are populated by both known observations and based on indications and estimations. If a hazard $h(v_i) = 1$, it means that there is a known hazard and vertex v_i is unsafe, and all evacuees should be routed away from the corresponding room. Similarly, $h(v_i) = 0$ means that v_i is a known safe vertex. All other hazards are estimated based on known observations.

Definition of the hazard function is not part of this paper. This paper treats the hazard function as an unknown stochastic function returning a probability of a hazard in the room.

3   Ant Colony Optimisation (ACO)

Problem solving approaches inspired by nature and animals, so called swarm intelligence, have received a lot of attention due to their simplicity and adaptability. Ant Colony Optmisation (ACO) is one of the most popular swarm intelligence algorithms due to its general purpose optimization technique. ACO consists of artificial ants operating in a constructed graph. The ants release pheromones in favorable paths which subsequent ant members follow. This way, the colony of ants will walk more towards favorable paths and in consequence iteratively build the most favorable solution.[12].

ACO was first used to find shortest path from a source to a sink in a bidirectional graph. It has later increased in popularity due to its low complexity and its ability to work in dynamic environments. The flexibility of ACO is apparent as it has successfully been applied in a wide variety of problems such as finding solutions for NP hard problems [13], rule based classification [14], and is shown to be particularly useful for routing in real time industrial and telecommunication applications.

Finding the shortest path in a graph $G(V, E)$ using ACO in its simplest form works as follows. Artificial ants move from vertex to vertex. When an ant finds a route s from the source v_s to the sink v_t, the ant releases pheromones $\tau_{i,j}$ corresponding all edges $e_{i,j} \in s$. The pheromones for all ants m is defined as

$$\tau_{i,j} \leftarrow (1 - p)\tau_{i,j} + \sum_{k=1}^{m} \Delta\tau_{i,j}^k \tag{1}$$

The function is for ant k is defined as

$$\Delta\tau_{i,j}^k = \begin{cases} Q/|s| & if \ e_{i,j} \in s \\ 0 & otherwise \end{cases} \tag{2}$$

where Q is a constant.

The aim of each ant is to walk from v_s to v_t forming the path s. This is achieved by the following rule. When ant k is in vertex i it chooses to go to vertex j with the probability $p_{i,j}^k$ defined as

$$p_{i,j}^k = \begin{cases} \dfrac{\tau_{i,j}^\alpha \eta_{i,j}^\beta}{\sum_{e_{i,l}} \tau_{i,j}^\alpha \eta_{i,j}^\beta} & if \ e_{i,j} \in N(s^p) \\ 0 & otherwise \end{cases} \tag{3}$$

where s^p is the partial solution of s, and $N(s^p)$ are the possible vertexes to visit given s^p. $\eta_{i,j}$ is the inverse heuristic estimate of the distance between node i and j, and α and β are numbers between 0 and 1 to give the relevant importance between pheromones and the heuristics function.

In its simplest form, the $\beta = 1$ and $\alpha = 1$ so that the ants only consider the pheromones — and the heuristic function is ignored, giving:

$$p_{i,j}^k = \begin{cases} \dfrac{\tau_{i,j}}{\sum_{e_{i,l}} \tau_{i,j}} & if \ e_{i,j} \in N(s^p) \\ 0 & otherwise \end{cases} \tag{4}$$

In layman terms, the amount of pheromone released represent quality of the solution. This is achieved by each ant releasing a constant amount of pheromones. Consequently, the shorter the path found, the more pheromone per edge is released. Further, each ant is guided by a stochastic mechanism biased by the released pheromones. Thus, the ants walk randomly with a preference towards pheromones. In this way, the ants incrementally build up promising search space with means that a route s converges towards the shortest route from v_s to v_t, $s*$.

ACO has also successfully been applied for many network applications [15] [16] [17]. It has been empirically shown to have favourable results compared to other routing protocols with respect to short path routing and reduced load balancing. Therefore it seems particularly promising for the finding escape routes.

4   Solution

This section presents the ACO algorithms for finding escape routes in three distinct realistic environments. First, the algorithms interact with a **static environment** where the hazard functions remain unchanged, yet unknown. This resembles classical optimisation problems where the aim is to find a search space s^* so that $f(s^*) \leq f(s) \forall_s \in \mathbf{S}$.

Subsequently, the problem is extended to interact with **dynamic environments** so that the probability of a hazard in v_i, $h(v_i)$, is no longer fixed but changes regularly according to some unknown stochastic function. This shows how well ACO works when environments change, such as fire spreading.

Last, ACO deals with the situation of **control flow**; each edge has a capacity $c(e_{i,j})$. This is a realistic environment where the doors between rooms have a limited capacity, which changes the problem dramatically as it is no longer sufficient to find an optimal solution for one person, but there is a need to consider the system as a whole.

This section presents empirical evidence for ACO working in all the above mentioned environments. In these experiments, all graphs are bidirectional, planar and connected — in line with common practice [6]. Without loss of generality, all experiments are carried out on randomly generated graphs of 1000 vertexes, and 5000 randomly distributed edges, of which 1000 edges are used to make sure the graph is connected.

All experiments are an average of 1000 runs.

4.1   Static Environments

ACO has been used for static routing in many situations before. In these experiments ACO is used in its simplest form, as described in section 3, with a slight adjustment. The constant Q is replaced with a function of s:

$$Q(s) = 1 - \Pi_{v_i \in s}(1 - h(v_i)) \tag{5}$$

I.e. $Q(s)$ represent the inverse hazard probability. The consequence of this is that safe paths are given large amounts of pheromones, and unsafe paths low amount. The pheromone updates are therefore as

$$\Delta \tau_{i,j}^k = \begin{cases} Q(s)/|s| & if \ e_{i,j} \in s \\ 0 & otherwise \end{cases} \tag{6}$$

In the static environment $h(v_i)$ is defined as a random number between 1 or 0, and remain unchanged for each experiment.

Figure 1a and 1b show the behaviour of ACO static environments. Figure 1b shows the behaviour were there in addition to the normal setup the graph is manipulated so that there exists an s so that $f(s) = 0$ — meaning that there always exists a safe path. The optimal solution is calculated using Djikstra's algorithm [18] by considering $h(v_i)$ as basis the cost function for edged $e_{*,i}$.
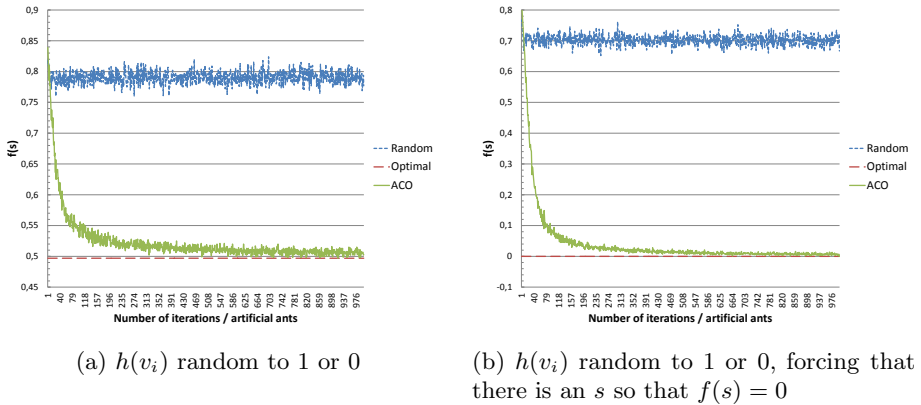
(a) $h(v_i)$ random to 1 or 0

(b) $h(v_i)$ random to 1 or 0, forcing that there is an s so that $f(s) = 0$

Fig. 1. Experiment results in static environments of randomly generated large graph comparing ACO to random and optimal. Hazards probabilities are randomly distributed as 0 or 1.

Figure 1b show the same experiment but with adjustments of hazard probabilities so that there is an s so that $f(s) = 0$ — meaning that there always exists a safe path.

Both experiments show that ACO is able to find the near optimal solution with very few iterations.
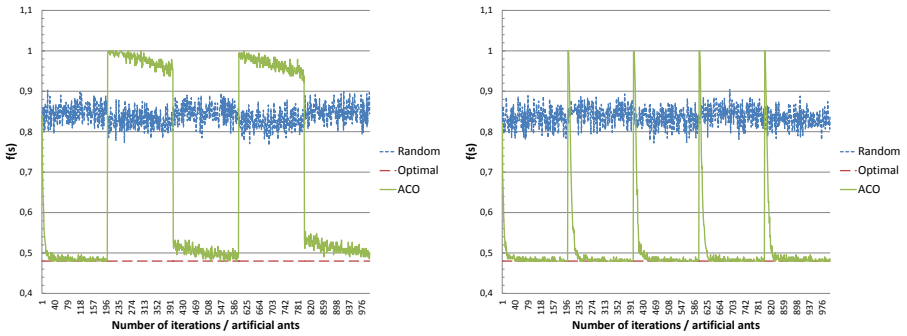
4.2 Dynamic Environments

ACO has also been used for dynamic environments in many situations [19] [20] [21]. This is achieved by letting, for each time step, the pheromones evaporate with a defined probability, typically between 0.01 and 0.20 [12]. The evaporation probability is balance between convergence accuracy and adaptability. I.e. you choose to what extent the ants should work towards a more optimal solution or should be able to adapt to potential other solutions. [21] showed that ACO based routing works well in situations with significant dynamics and continuously broken and newly established connections, which resembles finding an escape route when hazards change.

More specifically, ACO has been used for dynamic environments to detect weakness in networks [22]. This is done by utilizing a mechanism so that problems are reported on a black board. Ants which found a solution to a problem report this on a black board, and update pheromone value on the black board with an inverse probability of how it is solved. This way, collectively, more ants will walk towards problems which have not been solved and thus putting more effort on the unsolved vulnerabilities. Hence, dynamically weaknesses in networks are iteratively solved.

Figure 2a and 2b show ACO in dynamic environments where the $h(v_i)$ hazards are updated every 200th iteration with the following rule:

$$h(v_i) = 1 - h(v_i) \forall_{v_i} \in V \tag{7}$$

Thus, for every 200th iteration the environment is exactly opposite which in turn means that, if an algorithm has learnt an optimal route, the route changes to as far away from optimal as possible. The results show in figure 2a that when the evaporation rate is set to 0, the ACO learns an optimal solution which becomes outdated when $h(v_i)$ changes, and it is not able to adapt to the new optimal solution. Further, every time the hazard probabilities changes the algorithm is further away from the solution. On the other hand, figure 2b shows that when the evaporation rate is set to 0.2, the algorithm is able to quickly adapt to new environments — and is thus able to interact well with dynamic environments.

(a) Evaporation rate set to 0, ACO cannot adapt

(b) Evaporation rate set to 0.2

Fig. 2. Experiment results in dynamic environments. Hazard probabilities, $h(v_i)$, change for every 200th iteration.

4.3   Control Flow

The above situations only consider escapes where there are unlimited capacities — which is not a realistic situation.

While most existing work focus on computer model based escapes focus in finding the shortest safe path for each person [23] [24], this section innovatively extends the ACO to also consider control flow, and in this way avoid congestion of people. Thus an edge can at a certain time step either be full or have room for more people. We achieve this by letting the available capacity of an edge $e_{i,j}$ be defined by $c(e_{i,j})$. Any edge with $c(e_{i,j}) = 1$ cannot be used and is equal to a
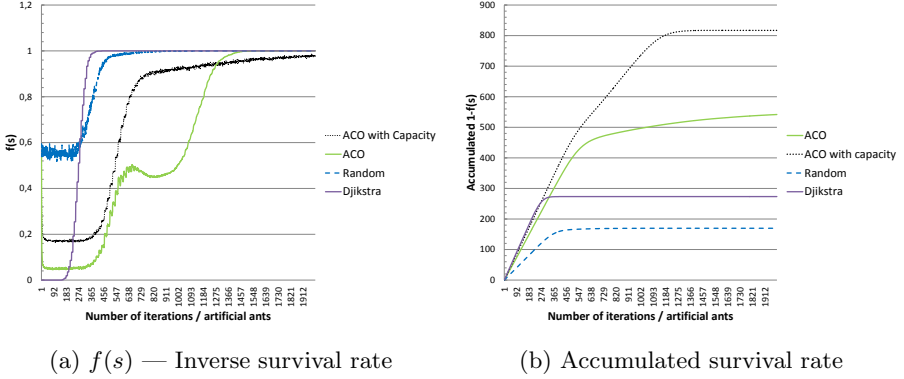
(a) $f(s)$ — Inverse survival rate   (b) Accumulated survival rate

Fig. 3. Experiment results comparing ACO capacity randomly distributed

non existing edge. The aim is therefore to find a set of search paths so that the combined hazard probabilities is minimized:

$$\text{Minimize} \sum_{s \in \mathbf{s}} f(s)$$
$$\text{Subject to} \sum_{e_{i,j} \in s} c(e_{i,j}) \tag{8}$$

ACO has been used in similar situations previously. [25] and [26] used ACO for a best planning, by letting the edged play an active role in the amount of pheromones available. After the artificial ants reach their destination, each edge multiplies the amount of pheromones with the used capacity. This way, the ants have two mechanisms to guide them: (1) The quality level of the path as noted by the amount of pheromones released by previous ants, and (2) the available capacity. [25] showed empirically that in most situation the ACO reached the same path as a brute force approach.

In line with state-of-the-art, we update the function $\Delta\tau_{i,j}^k$ from equation 2 as following.

$$\Delta\tau_{i,j}^k = \begin{cases} \dfrac{Q}{|s|} \dfrac{c(e_{i,j})}{\max(c(e_{i,j}))} & \text{if } e_{i,i} \in s \\ 0 \; \text{otherwise} \end{cases} \tag{9}$$

where $c(e_{i,j})$ is the used capacity of edge $e_{i,j}$ and $\max(c(e_{i,j}))$ is the total capacity available at $e_{i,j}$.

Figure 3a and 3b show the behavior of ACO in the environment which includes control flow. The experiments show that ACO considering capacity has a better over all performance than ACO without capacity consideration. It is noteworthy that Djikstra, which is identical to the algorithm used in previous experiments, does not perform well because it is not able to plan ahead, and is therefore not able to handle control flow well. In fact, both ACO approaches outperform the

much more costly brute force. Lastly, figure 3b shows the accumulated survival rate for each algorithm. This can be read as: How many people will survive if all followed the suggested plan perfectly.

5 Conclusion

This paper presents an application using ACO for finding safe escape routes in emergency situations. ACO is used in three main environments. Firstly, ACO operates in a stationary environment where it quickly reaches a near optimal solution. Secondly, this paper empirically shows that ACO is able to cope with dynamic situations were hazards rapidly change. Lastly, ACO is extended to handle vertex capacity. This enables ACO to find escape routes for groups of people. The application is an essential part of a tool mapping threats and planning escape routes.

For our further work, we plan to investigate ACO in more realistic scenarios, including which setup of ACO works best — such as pheromone distribution and evaporation probabilities. In this paper we assume that all participants can report their location. In practice, only some escapees will be able to provide this — rendering the problem more difficult. We have also assumed that all evacuees will follow a suggested plan. In a real evacuation situation, most people will escape in groups together with their family and friends. Further, when panic spreads people will be less likely to follow plans, which will affect overall results.

We also plan to examine how well ACO works compared to traditional control flow optimisation algorithms, and develop a smart phone application which can be used for safe escape planning.

References

1. Li, Q., Rus, D.: Navigation protocols in sensor networks. ACM Transactions on Sensor Networks (TOSN) 1(1), 3–35 (2005)
2. Li, Q., De Rosa, M., Rus, D.: Distributed algorithms for guiding navigation across a sensor network. In: Proceedings of the 9th Annual International Conference on Mobile Computing and Networking, pp. 313–325. ACM (2003)
3. Radiante, J., Granmo, O.-C., Bouhmala, N., Sarshar, P., Yazidi, A., Gonzalez, J.: Crowd models for emergency evacuation: A review targeting human-centered sensing. In: Proceeding of 46th Hawaii International Conference for System Sciences (2013)
4. Thompson, P.A., Marchant, E.W.: A computer model for the evacuation of large building populations. Fire Safety Journal 24(2), 131–148 (1995)
5. Thompson, P.A., Marchant, E.W.: Testing and application of the computer model simulex. Fire Safety Journal 24(2), 149–166 (1995)
6. Hamacher, H.W., Tjandra, S.A.: Mathematical modelling of evacuation problems– a state of the art. Pedestrian and Evacuation Dynamics 2002, 227–266 (2002)
7. Kim, D., Shin, S.: Local path planning using a new artificial potential function composition and its analytical design guidelines. Advanced Robotics 20(1), 115–135 (2006)

8. Braun, A., Bodmann, B., Musse, S.: Simulating virtual crowds in emergency situations. In: Proceedings of the ACM Symposium on Virtual Reality Software and Technology, pp. 244–252. ACM (2005)
9. Liu, S., Yang, L., Fang, T., Li, J.: Evacuation from a classroom considering the occupant density around exits. Physica A: Statistical Mechanics and its Applications 388(9), 1921–1928 (2009)
10. Wang, J., Lo, S., Sun, J., Wang, Q., Mu, H.: Qualitative simulation of the panic spread in large-scale evacuation. Simulation (2012)
11. Helbing, D., Farkas, I., Vicsek, T.: Simulating dynamical features of escape panic. Nature 407(6803), 487–490 (2000)
12. Dorigo, M., Birattari, M., Stutzle, T.: Ant colony optimization. IEEE Computational Intelligence Magazine 1(4), 28–39 (2006)
13. Gutjahr, W.: A graph-based ant system and its convergence. Future Generation Computer Systems 16(8), 873–888 (2000)
14. Liu, B., Abbas, H., McKay, B.: Classification rule discovery with ant colony optimization. In: IEEE/WIC International Conference on Intelligent Agent Technology, IAT 2003, pp. 83–88. IEEE (2003)
15. Dorigo, M., Stützle, T.: Ant colony optimization: overview and recent advances. In: Handbook of Metaheuristics, pp. 227–263 (2010)
16. Ducatelle, F., Di Caro, G.A., Gambardella, L.M.: An analysis of the different components of the antHocNet routing algorithm. In: Dorigo, M., Gambardella, L.M., Birattari, M., Martinoli, A., Poli, R., Stützle, T. (eds.) ANTS 2006. LNCS, vol. 4150, pp. 37–48. Springer, Heidelberg (2006)
17. Murtala Zungeru, A., Ang, L., Phooi Seng, K.: Performance evaluation of ant-based routing protocols for wireless sensor networks (2012)
18. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numerische Mathematik 1(1), 269–271 (1959)
19. Di Caro, G., Dorigo, M.: Antnet: Distributed stigmergetic control for communications networks. arXiv preprint arXiv:1105.5449 (2011)
20. Di Caro, G., Ducatelle, F., Gambardella, L.M.: Anthocnet: an adaptive nature-inspired algorithm for routing in mobile ad hoc networks. European Transactions on Telecommunications 16(5), 443–455 (2005)
21. Dressler, F., Koch, R., Gerla, M.: Path heuristics using ACO for inter-domain routing in mobile ad hoc and sensor networks. In: Suzuki, J., Nakano, T. (eds.) BIONETICS 2010. LNICST, vol. 87, pp. 128–142. Springer, Heidelberg (2012)
22. Selvan, G., Pothumani, S., Manivannan, R., Senthilnayaki, R., Balasubramanian, K.: Weakness recognition in network using Aco and mobile agents. In: 2012 International Conference on Advances in Engineering, Science and Management (ICAESM), pp. 459–462. IEEE (2012)
23. Zhan, A., Wu, F., Chen, G.: Sos: A safe, ordered, and speedy emergency navigation algorithm in wireless sensor networks. In: 2011 Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN), pp. 1–6. IEEE (2011)
24. Li, S., Zhan, A., Wu, X., Yang, P., Chen, G.: Efficient emergency rescue navigation with wireless sensor networks. Journal of Information Science and Engineering 27, 51–64 (2011)
25. Hsiao, Y., Chuang, C., Chien, C.: Ant colony optimization for best path planning. In: IEEE International Symposium on Communications and Information Technology, ISCIT 2004, vol. 1, pp. 109–113. IEEE (2004)
26. Sim, K., Sun, W.: Ant colony optimization for routing and load-balancing: survey and new directions. Systems, Man and Cybernetics, Part A: Systems and Humans 33(5), 560–572 (2003)