Hindawi
Mobile Information Systems
Volume 2017, Article ID 1328127, 15 pages
https://doi.org/10.1155/2017/1328127

Research Article

ANT: Agent Stigmergy-Based IoT-Network for Enhanced Tourist Mobility

Pablo López-Matencio,[1] Javier Vales-Alonso,[1] and Enrique Costa-Montenegro[2]

[1] *Department of Communication and Information Technologies, Technical University of Cartagena, 30202 Cartagena, Spain*
[2] *AtlanTTIC, Universidade de Vigo, EI Telecomunicación, 36310 Vigo, Spain*

Correspondence should be addressed to Pablo López-Matencio; pablo.lopez@upct.es

Received 28 April 2017; Revised 31 August 2017; Accepted 14 September 2017; Published 8 November 2017

Academic Editor: Paolo Bellavista

In this work, we propose ANT, a network of agents which exchange information in a stigmergy-based fashion. In ANT, each system's actor distributes *pheromones* as a way of indicating places' attractiveness, as well as for building a proper routing path to those sites. The goal of ANT is to improve the chances of discovering points of interest, as well as to reduce the time required for doing so. We have applied ANT to a tourist mobility scenario, where both people and things (events, restaurants, performances, etc.) participate. ANT has achieved notable success in this example case. We find that probability of discovering temporary events and dates improves by more than 35%, while the mean time employed to determine static point decreases by more than a third. We also introduce a mobile-based architecture which performs ANT tasks efficiently and easily for the user.

1. Introduction

In the last decade, the advances in mobile communications have been unstoppable. Cell phones are now widespread, and their regular use is adopted for most of the population [1, 2]. Mobile communications are changing the way people conduct their daily activities such as working, sleeping, cooking, running, reading news, or obtaining information. One important area of influence is tourism, where mobile applications have replaced not only maps and travel guide books but even the human guides. Most of these applications are recommenders that store and classify points of interests (POIs) based on previous user opinions with the downside of losing information on temporary events.

One way of having live data on POIs is through the social network. For example, applications such as Waze [3] and Swarm [4] indicate the state of events, thereby helping participants to make choices. While such systems provide useful information for specific searches or may inspire going to new places, they depend on visual information (e.g., maps, icons, or text) and information collection is for the common benefit of all participants. This constant flow of information may create information overload and distract visitors from exploring and experiencing the atmosphere of a city [5].

An alternative mode of sightseeing entails wandering around the city using a less intrusive guiding system. A visitor may want to feel they are finding interesting new places other people fail to see and share this information. On the other hand, we may want to visit the places that other tourists have encountered and to avoid the places already visited. Our approach combines both goals: serendipitous discovery while providing this information to the rest of participants. The core idea is that a central server provides the user the data to make route decisions. The implementation of these model results is challenging and, in particular, the planning of routes is crucial.

In this paper, we explore a guiding system for tourists based on *stigmergy*, which is a type of communication used by some species of insects, for example, ants [6], for foraging purposes. Stigmergy consists in liberating chemicals denominated *pheromones* while moving and leaving a trace that others can sense. The more individuals follow a trail, the higher amount of pheromones. This concentration of pheromone serves other individuals as a stimulus to follow a particular path. Additionally, a pheromone trail evaporation mechanism enables the search for food in different locations.

A systematic approach of stigmergy can resolve a complex problem via the single action of agents [7]. In our case, the visitor performs a tour carrying a software agent (e.g., on a smartphone). This software has two missions: to upload previously followed paths at interesting locations and to make route decisions. To make this computation, the agent receives a consolidated pheromone distribution of all users from the central server. In these calculations, the server increments the pheromone concentration of the tourist path to reach a POI, allowing others to discover it. Moreover, our model associates a pheromone profile with each agent to avoid any repetition of visits.

The key new features of our proposal are the following:

(1) Efficient. Finding hard or significant locations for each kind of user is difficult to achieve by conventional means but not with our system. It also enables users to bypass previous attractions.

(2) Transparent. The application directs users to interesting sites, avoiding the query-decision cycle of standard applications.

(3) Dynamic. Users can find very short-running events (e.g., a street performance). The system can also recognize a new POI simply by monitoring the agent.

(4) Emphatic. Pheromones act on groups of users according to their likes.

(5) Customizable. Introducing changes in the stigmergy operation (easily configurable by a short set of parameters) facilitates individual adaptation to the environment according to user preferences.

(6) IoT Integrated. Objects and other factors may lead and also govern pheromone liberation actions. For example, a lighting spectacle could liberate pheromones some minutes before it is switched on; a restaurant can liberate different kinds of pheromones at various times for tourists from several countries, and so on. There is a variety of new ways of interacting.

To our knowledge this work is the first to use stigmergy in the design of a tour navigation system. Stigmergy provides distinct advantages over the existing itinerary and route planning systems in discovering unexpected live events and provides a more simple operation and personalization of the tourist interest (e.g., controlling the associated pheromone concentration). The rest of the paper proceeds as follows: Section 2 summarizes the related works on tour planning and routing systems. Section 3 describes ANT architecture and provides an overview of its operation. Section 4 analyzes different mechanisms for tourist path planning, which we investigated numerically using a simulator in Section 5. Finally, Section 6 presents the conclusions.

2. Related Work

Itinerary planning is an ongoing problem that has fostered considerable attention by many researchers. We may categorize the existing works in two main groups, *offline* and *online*, depending on how they adapt their results with the changing conditions during the tour.

The *offline* approaches are adjusted before starting the visit and offer the user an itinerary for a set of POIs. These systems allow the tourist to select the POIs they want to visit. Then they prepare a personalized itinerary [8–11, 20, 21] and in some cases they receive user feedback, which provides the possibility of recommendation [9–11, 21]. The systems presented in [9, 10, 22, 23] have mobility support and are also location-aware, assisting the user with data of the closest POIs. Cellular broadband connections have enabled the development of mobile augmented reality-based (MAR) POI recommendation systems. In addition to an itinerary of POIs, MAR-based systems can deliver interactive information regarding dining, museums, and entertainment and even show virtual paths and directional arrows to facilitate navigation [24, 25].

The works [5, 26] follow a different model: they allow tourists to wander freely around the city and discover new POIs. Once the user is close to a POI the system signals the user (vibration in the case of [5]) or plays a recording with information relating to that POI [26]. These approaches are stand-alone oriented and, on the road, they are absent of central support. Similarly to [5], in [26] the authors use the concept of Audio Bubbles, informative recordings that the user can listen to when close to a POI. Neither [5] or [26] provide a navigation plan, only information about the current POI being visited.

Some other offline systems come with a full set of functionalities and provide an itinerary taking into account a variety of parameters. They can compute a touristic journey based on time scheduling of transport, route distance, user preference, or sightseeing time. The downside with these systems is that they have no live feedback from users, and the associated computation is complicated because they resolve an optimization problem. For example, [12, 27, 28] use genetic heuristics. Reference [29] utilizes simulated annealing and [13] solves a dynamic programming problem. Reference [14] uses a clustering algorithm based on location and tourist preferences and then computes the route using a greedy algorithm on those clusters. Other works iteratively construct the tour plan based on successive refinements of the initial user plan [9, 15, 16, 27] and the solution is approximated via metaheuristics. References [15, 27] use genetic algorithm whereas [16] uses a local search heuristic and [8] a binary search tree heuristic. The common term in the literature for these problems is *tourist trip design problem* (TTDP) or, simply, *Orienting Problem* [30]. In the two cases, the goal is to create a tour trip with the most desirable sites, subject to various budget and time constraints. Note that TTDP and TSP are different problems: the TTDP objective is not the shortest path but finding a suitable route for a particular user. The survey works [30, 31] collect the tourism-centered algorithms to solve this problem. Solving the TTDP is a daunting task (commonly, using metaheuristics) and it needs a lot of information. Our system is much simpler to implement: stigmergy can build a solution with the combination of valid individual routes between POIs and does not need user interaction or additional data. We would point out that stigmergy does not find the shortest solution, despite the fact that it finds a tour which passes through a set of POIs ranked by the concentration of pheromone, that is, the number of recent visits.

In contrast with the precedent systems, some approaches aimed to discover unknown POIs. We have labeled those systems *online*, as most of them were inspired by common features of social network platforms, providing access to user-generated content. Their primary design goal is to allow collaboration among users and support them in their visits, for example, using Twitter (https://twitter.com) or Facebook (https://www.facebook.com) as communication media (moreTourism [32] and e-Tourism [33]), obtaining the users' valuation of different POIs from them (Im feeling LoCo [34], iTravel [17], VISIT [18], and PlanTour [19]), or grouping near-by users for a joint activity (e-Tourism [33], iTravel [17], and Swarm [4]).

Most of these systems consider the location of people and recommend a personalized set of POIs. However, they focus more on recommending places and advertising new plans to the closest contacts than on supporting the navigation of individuals to those places. An illustrative application designed specially to provide information on transportation routes is Waze [3]. Waze is a commercial navigation system where users collaborate by uploading their own experiences as users. This application provides *live* visual data, such as the locations of slow traffic, police controls, speed radars, and roadworks, and computes the best route to a destination. Although all this information may be important during a trip, the purpose of Waze is not to discover interesting new places to go.

ANT stigmergy approach shares with the precedent online systems the capacity to receive feedback from other users (i.e., the most visited POIs) by controlling the pheromone profile of each user (see Section 4). Moreover, ANT can also discover *live* or temporary events and avoid the repetition of POIs. However, we may distinguish ANT from other online-based approaches for its simplistic operation, which does not require user interaction, thereby avoiding distractions and allowing for a more authentic touristic experience. Table 1 compares the ANT system approach with a group of representative references of this section.

Finally, to our knowledge, ANT is the first navigation system for tourism based on stigmergy. There are a few similar approaches, such as papers [35, 36], but their objectives are different from ours.

3. System Description

3.1. Architecture. The main actors of the ANT system architecture (see Figure 1) are the following:

(i) Final user, ANT agent: the ANT software runs on end users' devices (tourists' smart phones, smart watches, etc.). Its mission is to detect when a place is "interesting" for the tourist and to upload the previously followed path to the server when identifying this site. Moreover, it receives a consolidated pheromone map from the ANT server, which is used to make decisions about suitable routes for the user. Section 3.2 describes the technical details of these procedures. Figure 1 displays interesting sites that have been found (marked with blue stars).

(ii) Objects, ANT agent: an intrinsic feature of ANT is allowing objects to interact with the system; for example, these related objects can be promotional events like restaurants, bars, and public places or features. To that end, an ANT software runs at each place. It can run on an IoT device (e.g., a raspberry PI) or even on a smart device, such as a smartphone. They declare a point of interest to the server and associated routes, similarly to the previous case. In this case, a pheromone map is not necessary. An example of a site identified event appears in Figure 1 as the red star.

(iii) ANT server: the server is responsible for generating pheromone information for the agents (i.e., tourists), defining its evolution over time (evaporation and reinforcement processes), and distributing the required information to each agent, for the routing tasks. Section 4 explains the pheromone control procedures.

3.2. Operation Overview. As stated in the Introduction, our approach intends to improve tourists' ability to find new POIs by exploiting the foraging mechanism of ants. Similarly to them, ANT's actors deposit pheromones (indicating site *popularity*). Pheromone mapping is used to decide individual user routes according to different strategies. For that, the system assumes the following rules:

(1) In the absence of *pheromone* a tourist moves at its own will; that is, there is no influence from other ANT agents in path selection.

(2) ANT software keeps track of the paths the tourist has followed, the Route Information (RI), for the last Memory Path Interval (MPI) minutes.

(3) When a visitor remains in or around a place during a POI Detection Interval (POI-DI), the ANT software declares a POI and forwards the RI to the ANT server. The parameter POI Area Distance (POI-AD) refers to the space bound to limit the POI triggering. That is, the user needs to move less than POI-AD during POI-DI.

(4) Upon RI reception, the server checks whether another actor has previously declared this POI. It does so by checking whether another POI is within POI-AD of the current one. If it is a new POI, the server creates a POI Unique Number (POI-UN). Otherwise, the system recovers its POI-UN from the database. Then, the server adds pheromone to the given RI, according to the route ranking procedure (see Section 4). Each

Table 1: Tour itinerary planning systems.

	Online/offline approach	Need of user feedback	Capture of live events	Planning algorithm	Centralized architecture	Mobile support	Location aware	Map support	Personalized itinerary
Gavalas et al. [8]	Offline	No	No	Heuristic for binary tree search (DailyTRIP) to find POIs	Yes	No	No	Yes	Yes
MyMitelene/ Gavalas and Kenteris [9]	Offline	Yes	No	k-Means algorithm to find the POIs for each day	Yes	Yes	Yes	Yes	Yes
Wörndl et al. [10]	Offline	Collect POIs category from Foursquare (https://www.foursquare.com)	No	Dijkstras algorithm to find the shortest route	Yes	No	No	Yes	Yes
RouteMe/ Herzog et al. [11]	Offline	Yes	No	Collaborative filtering based on users' ratings to compute itinerary between two locations	Yes	Yes	No	Yes	Yes
Hornecker et al. [5]	Offline	No	No	—	No	Yes	Yes	Yes	No
Zheng et al. [12]	Offline	No	No	Generic algorithm heuristic to solve TTDP. It takes into account the tourist's state of fatigue	—	—	—	—	Yes
Gionis et al. [13]	Offline	Collect POIs category from Foursquare	No	Dynamic programming algorithm to resolve tour route problem	—	—	—	—	Yes
Awal et al. [14]	Offline	No	No	k-Means algorithm to cluster POIs based on location and user preferences	—	—	—	—	Yes
CT-Planner5/ Kurata et al. [15]	Offline	Yes	No	Genetic algorithm to compute the itinerary of POIs, and Google Maps API to find the best route	Yes	No	No	Yes	Yes
Zhu et al. [16]	Offline	No	No	Local search heuristic to compute a route plan	—	—	—	—	Yes
iTravel/Yang and Hwang [17]	Online	Users exchange POIs valoration on-the-road	Yes	Computes the itinerary of POIs via collaborative filtering technique on users' ratings	Yes	Yes	Yes	Yes	Yes
VISIT/ Meehan et al. [18]	Online	Collect user sentiments from Twitter and Facebook	No	Compute the itinerary of POIs via a combination of decision-making techniques on users' parameters	—	Yes	Yes	Yes	Yes

Table 1: Continued.

	Online/offline approach	Need of user feedback	Capture of live events	Planning algorithm	Centralized architecture	Mobile support	Location aware	Map support	Personalized itinerary
PlanTour/ Cenamor et al. [19]	Online	Collect POI and user data from Minitube (http://www.minube.net) Travel blog and Yelp (https://www.yelp.com)	No	k-Means algorithm to find the POIs for each day	Yes	No	No	Yes	Yes
Swarm [4]	Online	Yes	Yes and generated by the own user	Commercial, n/a	—	Yes	Yes	Yes	Yes
Waze [3]	Online	Yes	Yes	Commercial, n/a	—	Yes	Yes	Yes	Yes
This work: ANT	Online	No	Yes and generated by the own user	Route selection based on the current POI popularity obtained via stigmergy operation	—	Yes	Yes	No	Yes

POI has a particularly associated pheromone type over the map (identified by the POI-UN).

(5) Each Updating Period (UP), the ANT server evaporates pheromone of all kinds at a predefined Evaporation Rate (ER). The next section discusses the evaporation mechanism. If a pheromone associated with a POI-UN disappears entirely, the server releases its associated information.

(6) Each UP, the ANT server communicates tourists with their Individualized Pheromone Mapping (IPM). The IPM provides pheromone mapping of unvisited POIs (once a POI has been visited, visitors are not likely to focus in this place again). Therefore, it will be different for each tourist. The next section describes how the server builds the information into the IPM.

(7) The ANT agent uses the IPM to make route decisions. These decisions will most likely drive the agent to unvisited POIs, which are popular because other tourists have already visited them.

(8) In addition to the carried actions made on POIs, the mobile ANT software will inform and submit path data to the server when a featured event takes place. Notice that these POIs can be temporary, for example, the start of a guided visit to some historic building. In this case, if the server eventually releases a POI, an object can declare a new one (different POI-UN) later in the same location, allowing tourists to visit this interesting reborn site.

The combined operation of the different ANT actors improves the tourist's ability to find POIs if compared with an occasional traveler or even with a local visitor.

Finally, Table 2 resumes the key parameters of the ANT system.

4. Routing Ranking

In this section, we present three methodologies for the mobility of tourists: *Fitness Proportionate Selection* (FPS), *Rank Selection* (RS), and *Gradient Based* (GB). They use pheromone deposited when other agents make a POI discovery to make mobility decisions probabilistically.

To formalize our mobility model, we first envisage city streets as a graph $G(V, E)$. Vertices V are the crossroads where tourists make decisions on their next routes, and the edges E are the streets. Our model uses the following nomenclature:

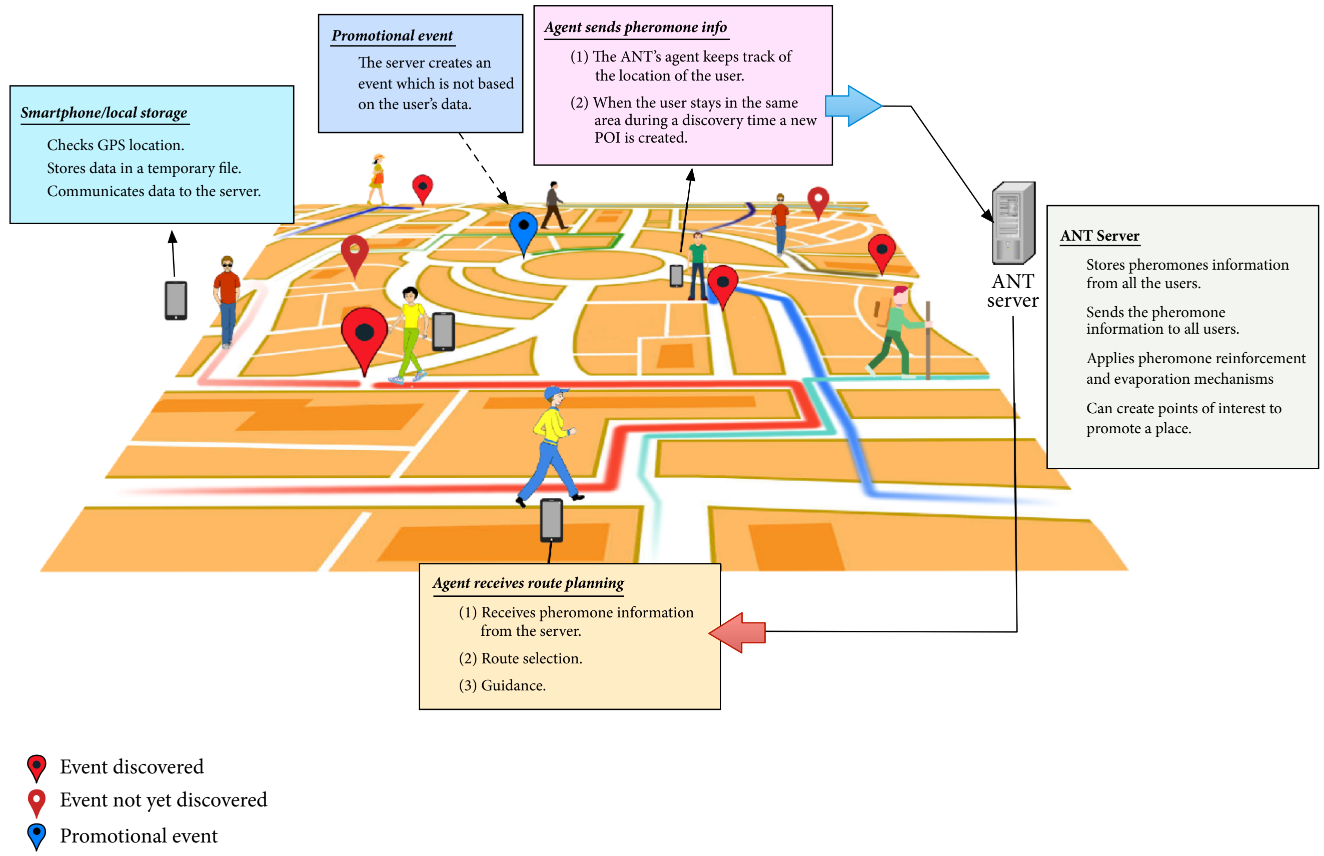
(i) $A = \{a_i \mid i = 1, \ldots, n\}$ is a finite set of tourists (agents).

(ii) $B = \{b_i \mid i = 1, \ldots, m\}$ is a finite set of POIs.

(iii) The edges $E = \{e_{i\to j} \mid i, j = 1, \ldots, |V|\}$ of the graph possess pheromone values for each POI. At edge $e_{i\to j}$ pheromone is $\phi^{\,e_{i\to j}} = (\phi_1^{e_{i\to j}}, \ldots, \phi_m^{e_{i\to j}})$, $\forall \phi \in \mathbb{R}_+$.

Note that POIs have separated pheromone traces over the graph. The reason for this design decision is to differentiate POIs and avoid visiting the same location twice. Therefore, the distribution of pheromones is different for each agent. Specifically, guiding a tourist requires controlling the related amount of pheromone in each arc of the graph, that is, the IPM. For that goal, the server executes the process of pheromone *reinforcement* (related to POIs found) and *evaporation* (at each UP) to enable fast discovery of new POIs.

Pheromone Reinforcement. An agent reaching an unvisited POI sends the stored RI to the ANT server. If the RI has loops, the server is in charge of detecting and eliminating them. The problem with loops is that they may receive pheromone several times, leading to self-reinforcement loops. During the update process, the server deposits an amount of pheromone



Smartphone/local storage

Checks GPS location.
Stores data in a temporary file.
Communicates data to the server.

Promotional event

The server creates an event which is not based on the user's data.

Agent sends pheromone info

(1) The ANT's agent keeps track of the location of the user.

(2) When the user stays in the same area during a discovery time a new POI is created.

ANT Server

Stores pheromones information from all the users.

Sends the pheromone information to all users.

Applies pheromone reinforcement and evaporation mechanisms

Can create points of interest to promote a place.

Agent receives route planning

(1) Receives pheromone information from the server.

(2) Route selection.

(3) Guidance.

FIGURE 1: System overview.

TABLE 2: ANT: list of parameters.

RI	Route information	Run-time generated
MPI	Memory Path Interval	Time from last POI discovered
POI	Point Of Interest	>5
POI-DI	POI Detection Interval	>10 min
POI-AD	POI Area Distance	<50 m
POI-UN	POI Unique Number	Run-time generated
UP	Updating Period	<10 s
ER	Evaporation Rate	<20%
IPM	Individualized Pheromone Mapping	Run-time generated

on edges of the graph that correspond to the RI. In particular, if tourist a_i visits POI b_k, the pheromone is changed as follows:

$$\phi_{b_k}^{e_{i\to j}} \longleftarrow \phi_{b_k}^{e_{i\to j}} + \Delta\phi, \quad \forall e_{i\to j} \in \mathrm{RI}^{a_i}\left(b_k\right). \tag{1}$$

By this rule, a tourist using edge $e_{i\to j}$ increases the probability that future tourists will use the same edge.

An important aspect is the selection of $\Delta\phi$. Unlike TSP problems which commonly use pheromone increments as a decreasing function of the path length, we only aim to attract tourists to the most visited POIs. In this case, the simplest choice is using a constant value for all the agents.

Pheromone Evaporation. Pheromone evaporation can be seen as an exploration mechanism that allows a tourist (agent) to search different locations. Evaporation decreases the pheromone trails of the visited POIs with an exponential speed of value ER. In particular, a tourist a_i that has already visited the set $B^\star$ of POIs, evaporates pheromone by applying the following rule to all edges:

$$\phi_{B^\star}^{E} \longleftarrow (1-\mathrm{ER})\,\phi_{B^\star}^{E}, \tag{2}$$

where ER $\in$ $(0,1]$ is a parameter. Without reinforcement, the pheromone related to $B^\star$ will vanish and become 0. This allows agents to search different locations.

Finally, we would point out that, at every UP, the server generates the tourist's individualized pheromone distribution or IPM required to decide the next road to take. In particular, a tourist a_i at the current end j of an edge must determine which of the next possible edges is using his/her IPM^{a_i}. The possible set of edges is $\mathcal{N}_j^{a_i}$.

The next section explains the methods used to make those decisions.

4.1. Fitness Proportionate Selection. This method (a.k.a. roulette-wheel selection) is a genetic algorithm (GA) to

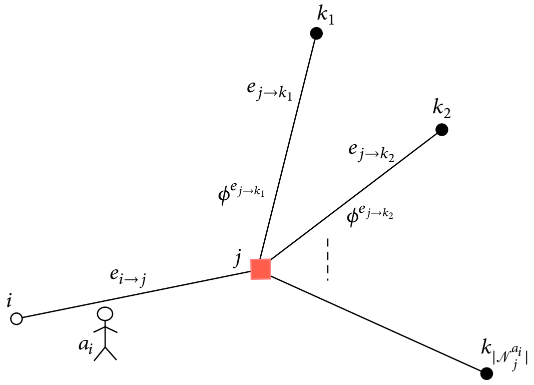


Figure 2: Parameters involved at the tourist decision points.

select the chromosomes of an offspring [37]. Moreover, FPS is commonly used in current research on routing vehicles and transportation logistics, for example, assigning routes to different carriers to minimize costs [38, 39], or to maximize the number of deployable targets using a combination of terrestrial and aerial transportation [40], or even the typical TSP problem [41]. Similarly to them, FPS suits our need to select the next street when there are several alternatives (see Figure 2).

In FPS, each possible next link is assigned a probability in proportion to its fitness function, which we define as follows:

$$\begin{aligned} f : \mathbb{R}_+^m &\longrightarrow \mathbb{R}_+ \\ \phi &\longmapsto \sum_{i=1}^{m} \phi_i. \end{aligned} \tag{3}$$

Our fitness function $f(\cdot)$ calculates the pheromone of each road. That is, popular roads have a larger probability of being taken than the less popular ones. A tourist with low or no interest in a POI (e.g., one already visited) that is popular on one road is highly unlikely to take that path. Therefore, our f represents the *effective* pheromone of an event.

The implementation is as follows:

(1) Evaluate the fitness, $f_j^{a_i}$, of each edge of $\mathcal{N}_j^{a_i}$ for tourist a_i at decision point j.

(2) Compute the probability, $p_k^{a_i}$, of selecting each member of $\mathcal{N}_j^{a_i}$:

$$p_k^{a_i} = \frac{f_k^{a_i}}{\sum_{\forall l \in \mathcal{N}_j^{a_i}} f_l^{a_i}}. \tag{4}$$

(3) Compute the cumulative probability:

$$q_k^{a_i} = \sum_{l=1}^{k} p_l^{a_i}, \quad \forall k \in \mathcal{N}_j^{a_i}, \tag{5}$$

where p_l's has been previously ranked from lower to higher values.

(4) Generate a uniform random number, $r \in (0, 1)$.

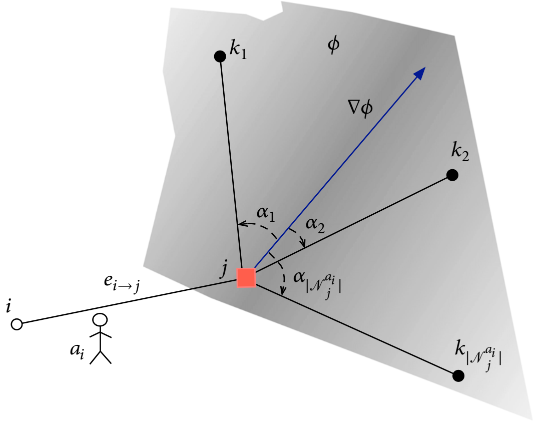


Figure 3: Gradient scheme of decision.

(5) If $r < q_1$ then select the lowest ranked path, or else select the path k such that $q_{k-1} < r \leq q_k$.

4.2. *Rank Selection.*

FPS method almost always selects the road with the highest probability when its fitness value is much greater than the other alternatives. For example, if a road has the best fitness value corresponding to 90% of all the probabilities, then the likelihood of other roads being selected will be very slim. In this case, tourists might have small chances of discovering new POIs, more so if they are temporary.

Rank selection first sorts the roads, and then every alternative receives fitness according to its ranking. The worst will have fitness 1, second worst 2, and so on. The best will have fitness N (number of paths at the crossroad).

After this, all the paths have a chance of being selected. Although this method can lead to slower convergence, it allows for expansion of the search space and discovering new POIs not visited yet by any other tourist.

4.3. *Gradient Based.*

This strategy reflects the behavior of many relevant phenomena in nature, where individuals move towards a major concentration or a greater measure of some element. For example, animals perform migratory tasks by leveraging environmental cues such as nutrient [42] and thermal gradients [43], or magnetic fields [44]. Even at a cellular level, there is a reaction that triggers lymphocyte chemotaxis towards the tumor to form a lymphocyte barrier to contain tumor progression [45].

In our case, we want to direct people towards major concentrations of tourists, which correspond to the direction of the gradient vector of the scalar field of pheromone. We represent this pheromone in the plane by a function $\Phi : \mathbb{R}^2 \to \mathbb{R}^2$, which is continuous and differentiable. Therefore, at a decision location j a tourist a_i follows the next procedure:

(1) Compute the gradient vector at point j (Figure 3) of the pheromone values on the map, $\nabla\Phi(j)$.

(2) Compute the angles $\alpha_1, \ldots, \alpha_{|\mathcal{N}_j^{a_i}|}$ between the gradient vector and each possible path.

(3) Get the weights

$$w_k = \frac{\pi - \left|\alpha_k\right|}{\sum_{\forall l \in \mathcal{N}_j^{a_i}} \left(\pi - \alpha_l\right)}, \tag{6}$$

$k = 1, \ldots, |\mathcal{N}_j^{a_i}|$, which are inversely proportional to the angle; that is, the lower the weight assigned to a path, the closer that path to the direction of the highest concentration of pheromone.

(4) Rank w_k's and take the path with the lowest value.

In our approach, these computation results are daunting because we look for different events and because tourists have different interests. Thus, we must deal with the effective pheromone, f, for each event, which increases the computation load. For the sake of simplicity, we compute

$$\max_{\forall k \in \mathcal{N}_j^{a_i}} f_k^{\mathcal{N}_j^{a_i}}\left(j\right) \tag{7}$$

so the corresponding path to this value is considered the closer path (in degrees) to $\nabla\Phi(j)$.

5. Evaluation and Results

To test the proposed ranking methods we used Netlogo agent based simulator [46], which is particularly well suited for modeling complex systems (e.g., natural and social phenomena) developing over time. Programmers can instruct hundreds or thousands of agents, all operating independently. This feature makes it possible to explore the connection between the micro-level behavior of individuals and the macro-level patterns that emerge from their interaction. Also, Netlogo can import ESRI shapefiles (.shp), vector files which allow the simulation of *walker* agents over a map.

For our trials, we chose the town center of Ghent (Belgium). Ghent is a tourist town with many visitor sites, such as old buildings and museums and, like many old towns, has an entangled network of walkways, making it a suitable example due to its complexity. For our purposes, we used a *squared* area of N, S, E, and W each 525 m from location point 51.05660 N, 3.721500 E. Figure 4 represents a map with walkways and building footprint of this area, obtained with the OSMnx [47] Python module. Paths (white lines) on this map are similar to a maze, which is a challenge for ANT.

We assumed that tourists move along these lines and turn back when they reach the limits of the map. In the absence of pheromone, that is, without any guiding information from others, we assumed a random-walk model; that is, next edge is selected at random with a uniform distribution. The goal of our evaluation was to compare the performance, regarding event detection probability and discovery time of the random-walk mobility model with the schemes provided by ANT and described previously in Section 4. We assumed that events happen in the map and that when tourists reach one, they will remain around it for a *period of visit* (PV). To implement this mechanism for each user, we defined an event attraction function, which is initialized to its maximum value when the user comes closer to that event (at less distance than the POI-AD parameter). Afterwards, this attraction function decays exponentially and is used to modulate the IPM received by the server *before* the user triggers the corresponding POI. We would point out that this attraction modulation takes place only for simulation purposes, but it is not part of the route ranking mechanism. If the user remains in the event for a given time, the POI will be declared as stated in Section 3.2. From this time on the IPM no longer contains pheromone associated with the currently visited event, and the simulated tourist continues reaching for new events.

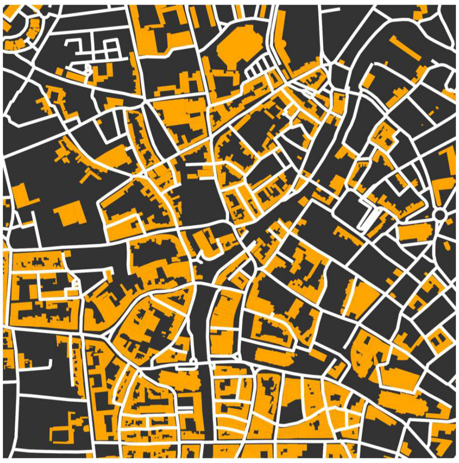


Figure 4: Simulation map (Ghent, Belgium).

Next, we describe the different tests performed in this simulated environment.

5.1. Static Simulation. First, we evaluated a *static scenario*, in which events are at fixed map locations. In this case, we measured the time taken to visit each event for the first time. We sorted these times by precedence and characterized their distributions, as well as the joint one. From these distributions, we also computed the mean first passage time (both individual and global). Moreover, we obtained the POI detection ratio, that is, for each tourist, the number of discovered POIs from all the existing POIs during the tourist's journey.

Table 3 shows the configuration parameters of our simulator. The simulation goal for each tourist was to find 5 events and to save the time instant of each discovery, or *discovery sample*. The program ran a permanent population of 100 tourists. At the fifth POI (i.e., 5 discovery samples), the visitor left the scenario, and a new one started the search for new locations. Visitors stayed in a POI for 8 minutes on average. Finally, the simulation stopped when there was a total of 625000 discovery samples.

We ran simulations for each ranking method. Table 4 shows the main statistics obtained by each method, and Figure 5 plots their pdf fit of their sample times. Note that, for our purposes, time is in minutes. In addition, for each tourist, we number the events by discovery order, that is, events 1, 2, 3, 4, and 5.

Table 3: Configuration parameters in static scenario.

Tourists	100
Tourist pace	5 Km/h
Speed variation	0.3%
Number of POIs	5
Pheromone increment	1 unit
ER	10%
MPI	<2 h
POI-DI	8 min
POI-AD	40 m
PV	15 min
Total visiting tourists	125000
Total discovery samples (each. mob. model)	625000

At the beginning of a simulation, the streets were empty of pheromone feedback. Therefore, agents first decisions were random and as a consequence, the medians (Md) of time to reach the first event are similar in the four methods. Moreover, their distributions have modes below 20 minutes (see event 1 in Figure 5). However, on the next passage times (i.e., event 2 and forward), the sample mean starts to be greater with no stigmergy than those obtained with it; this trend increases in the next passage times (3, 4, and 5). As we expected, no stigmergy motion gives the worst mean passage times of the five events. We would point out, for example, that the time spent to arrive at event 4 is >13 hours and to event 5 >23 hours. These results seem logical if we think that the tourists are visiting other POIs (i.e., the ones they found and those declared by other people) but, also, because they select streets at random.

In contrast, the stigmergy methods obtain means at least 50% lower than without stigmergy, and these differences increase during the simulation. We observe how tourists using gradient system have the shortest mean passage times. They discovered the five events in about 5 hours, which is an outstanding result considering that each visit lasts 15 minutes and the tourists likely make more than five visits in their journey. Therefore, the means of the fifth event in Table 4 show that with gradient the speed of discovery is 77.3% greater (on average) than with no pheromone, 39.4% compared to rank selection, and 40.9% better than FPS. FPS and rank method result in similar speeds, but rank slightly improves FPS (2.5% faster).

To better characterize the discovery times we also observe how *disperse* they are, that is, their standard deviation s with each method. The pdfs of Figure 5 give us a picture of the distribution of the passage times along the different events. We see that standard deviation rises with each event discovered (approximately at the same scale on methods using stigmergy). In the best case (gradient method), s is 28 minutes at event 1 and about 3:45 hours at event 5, whereas FPS has 52 minutes at event 1 and 5:30 hours at event 5. The three stigmergy methods show s values comparable to their $\overline{x}$, meaning that tourists who assisted with stigmergy discovered events much faster than others. However, no stigmergy-based tourists have the greatest variability in their sample times, with s values that double (at each passage time) the largest s on a stigmergy method (FPS). Consequently, the no stigmergy pdf result is almost *flat* at event 2 and its tail, for all events, is higher than with stigmergy methods. Therefore, tourists without stigmergy might have a high probability of large discovery times. Besides, the difference $\overline{x}$-Md indicates the existence of high sample times that increase the mean time. The greater this difference, the more possibilities of their existence. Tourist motion without stigmergy provides the greatest $\overline{x}$-Md values.

Also, we analyzed the intervisit period (i.e., the time spent moving between consecutive visits). Figure 6 contains four plots for each routing method. In each plot, we represent two consecutive journey times of the tourist (the current journey on x-axis and the precedent one on y-axis). The diagonal line is the locus of equal values. We observe that the probability of having longer trips increases on each journey because the highest values (i.e., white colored points) fall below the line; the longer the distance of those values from the diagonal, the greater the difference between the preceding journey times. Hence, the mean values increase with each journey; thereby, also, the number of samples plotted below the line is higher indicating greater s. Consequently, tourists might have longer journey times. This result is more significant on tourists not assisted by stigmergy. It is worth noting that the times to discover the first event might have a similar probability on all routing methods. This result reaffirms the fact that tourists spend similar times to find the first event because of the initial absence of pheromone. Therefore, Figure 6 corroborates the results of Table 4 and Figure 5 in which tourists assisted by gradient method achieve the shortest mean times of discovery.

Finally, we computed the percentage of POIs detected (POI detection ratio) during the simulation of each tourist. Let us recall that the server will declare a new POI if a visitor stays around POI-AD distance (40 m) during POI-DI time (8 min). Besides, if the POI has no visits, then its associated pheromone vanishes and the server will remove its POI-UN. Therefore, these POIs are active during the time they receive visits, and a tourist will have a limited time to discover them. Despite this constraint, the simulations indicate that tourists assisted by pheromone reach a high number of POIs. Similarly to the other tests performed, gradient shows the highest ratio, 70.13%, and the ratio of FPS and rank selection is slightly lower (see Table 4). In contrast, tourists not assisted by pheromone only visit a small number of POIs (4.61%). Without pheromone, the POIs will only be active in the simulator when tourists are visiting them.

These results confirm the influence of stigmergy in the improvement of the discovering abilities of tourists in a static scenario. In fact, the gradient method is the one where stigmergy has the bigger influence on mobility decisions, followed by FPS, and ranking selection.

5.2. Dynamic Simulation. Next, we tested our methods in a *dynamic scenario* of events. In this scheme, the events appear and disappear at random times and locations. In this

Table 4: Statistics for datasets of time discovery samples (in minute time unit) of POIs 1 to 5, and the percentage of POIs detected by each tourist.

		No stigmergy	FPS	Gradient based	Rank selection
Event 1	$\overline{x}$	94.04	45.41	27.94	40.60
	s	137.98	52.73	27.18	47.41
	Md	38.40	26.13	20.53	23.33
	$\overline{x}$-Md	55.64	19.28	7.41	17.27
Event 2	$\overline{x}$	265.87	119.37	69.59	111.18
	s	236.14	88.03	48.75	83.98
	Md	199.60	97.07	57.47	89.20
	$\overline{x}$-Md	66.27	22.30	12.12	21.98
Event 3	$\overline{x}$	499.87	212.88	124.10	198.84
	s	350.66	131.26	77.38	124.49
	Md	417.47	186.27	103.33	169.60
	$\overline{x}$-Md	82.41	26.61	20.77	29.24
Event 4	$\overline{x}$	835.11	339.06	196.85	319.26
	s	510.23	202.35	114.35	182.25
	Md	724.13	302.67	166.80	279.20
	$\overline{x}$-Md	110.98	36.39	30.05	40.06
Event 5	$\overline{x}$	1417.80	544.18	321.34	530.68
	s	797.25	334.79	225.85	367.32
	Md	1254.40	480.13	268.93	460.93
	$\overline{x}$-Md	163.36	64.05	52.40	69.75
Mean POI detection ratio		4.61%	69.46%	70.13%	67.08%

case, our goal was to get the greatest number of tourists to visit every POI. Therefore, this dynamic problem is far more difficult than the static one, because tourists can only find the interesting locations during their *lifetime*.

The parameters used in the simulator are shown in Table 5. The events appeared on the map following a Poisson random variable of rate λ and were on a location *(live)* during an exponential variable time. This configuration enables a random number of ongoing events that vary over time (see Table 5). The evolution of the system showed approximately ten dynamic events throughout the simulation. Also, the server generated new POIs if a tourist wandered around a location in a time POI-DI (8 min). These POIs were active as long as pheromone was associated with them or as long as they had visitors. The period of visit (PV) of each POI was 15 minutes. In this scenario, the goal was to count the number of tourists visiting each dynamic event and to record the total once it disappeared in the simulator; we also computed the ratio of the number of POIs detected by a tourist and the total number of POIs during de simulation. Finally, we ran 380 simulations with 300 samples (i.e., dynamic events) in each run, making a total of 114000 event samples for each routing method. Figure 7 shows the histograms for each scheme and Table 6 a comparison of their numerical results.

The mean number of tourists' visits moving without stigmergy is 34.55, which is remarkable, since we know that they decide without any assistance or reference guide. Because tourists select paths at random, they can exploit the dynamic number of events and reach them without assistance. Despite this remarkable result, the data in Table 6 indicate that events receive more visitors (on average) if tourists select routes assisted by stigmergy than without it, 14.02% with FPS, 24.23% with the rank selection, and 35.80% with the gradient method.

Another remarkable result is that stigmergy is useful for discovering new POIs, but not all dynamic events receive visits. Some POIs may end up with no visitors because the latter are following the pheromones of some other POIs. This issue is common to all routing methods but is more evident for FPS and gradient. The reason is that, in FPS and gradient, pheromone concentration has more leverage on the decision of path than rank selection and, for that, POIs are more likely to go unvisited (see Figure 7). In the case of the gradient, tourists deterministically follow the path of the region with more pheromone; with FPS tourists are more likely to take the street with more pheromone. The histograms show a likelihood of about 15% of POIs with $[0, 2.5]$ tourists using gradient, whereas with FPS 25% of POIs have $[0, 2.5]$ tourists. The value of $\overline{x}$-Md corroborates this result; $\overline{x}$-Md is negative (see Table 6), which indicates that there are more samples above the mean than below.

The number of unvisited POIs can be huge with FPS and gradient. In contrast, and precisely because decisions depend mainly on pheromone concentration, FPS and gradient are also able to guide a large number of tourist to the POIs; therefore, some POIs receive a high percentage of visits. For example, when using gradient, POIs have a 51.5% probability of receiving ≥60 tourists; with FPS, POIs have a 34%

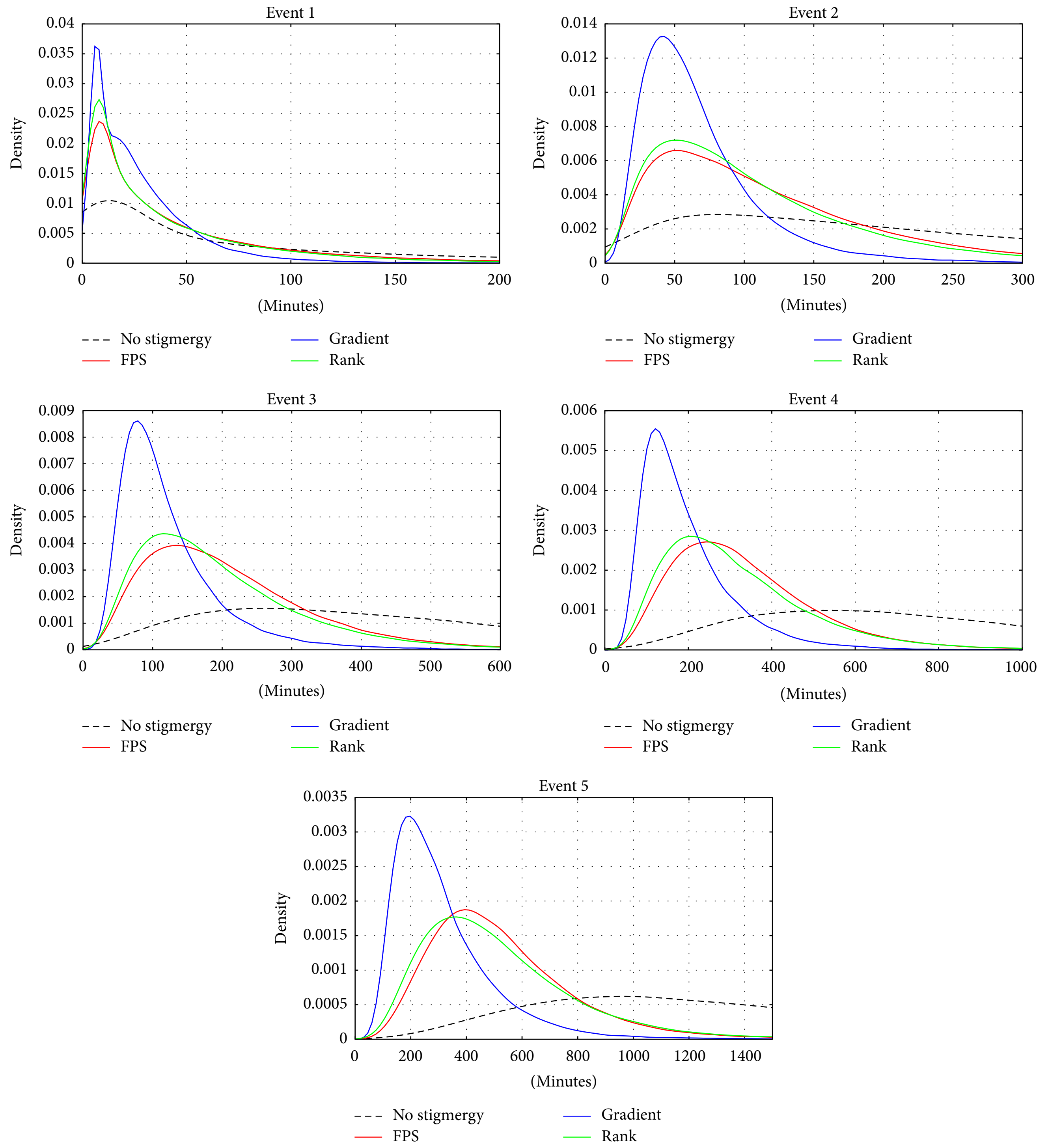


Figure 5: Estimation of the pdf for datasets of discovery samples of POIs 1 to 5 obtained with configuration of Table 3.

probability of having more than ≥60 tourists. Therefore, both gradient and FPS are convenient if we want to have POIs with a significant percentage of visitors.

In short, gradient and FPS methods achieve distributions of visits per POI with two separated zones of high probability: low and high number of visits (see Figure 7). We can corroborate this result by examining their deviations. The s values of FPS and gradient are high when compared to rank and no stigmergy, indicating that their samples are dispersed around the mean.

On the other hand, the two distributions of probability, no stigmergy and rank selection, have bell-shaped histograms. Consequently, the probability of visits to POIs is concentrated at some positive values greater than 0, that is, there is reduced

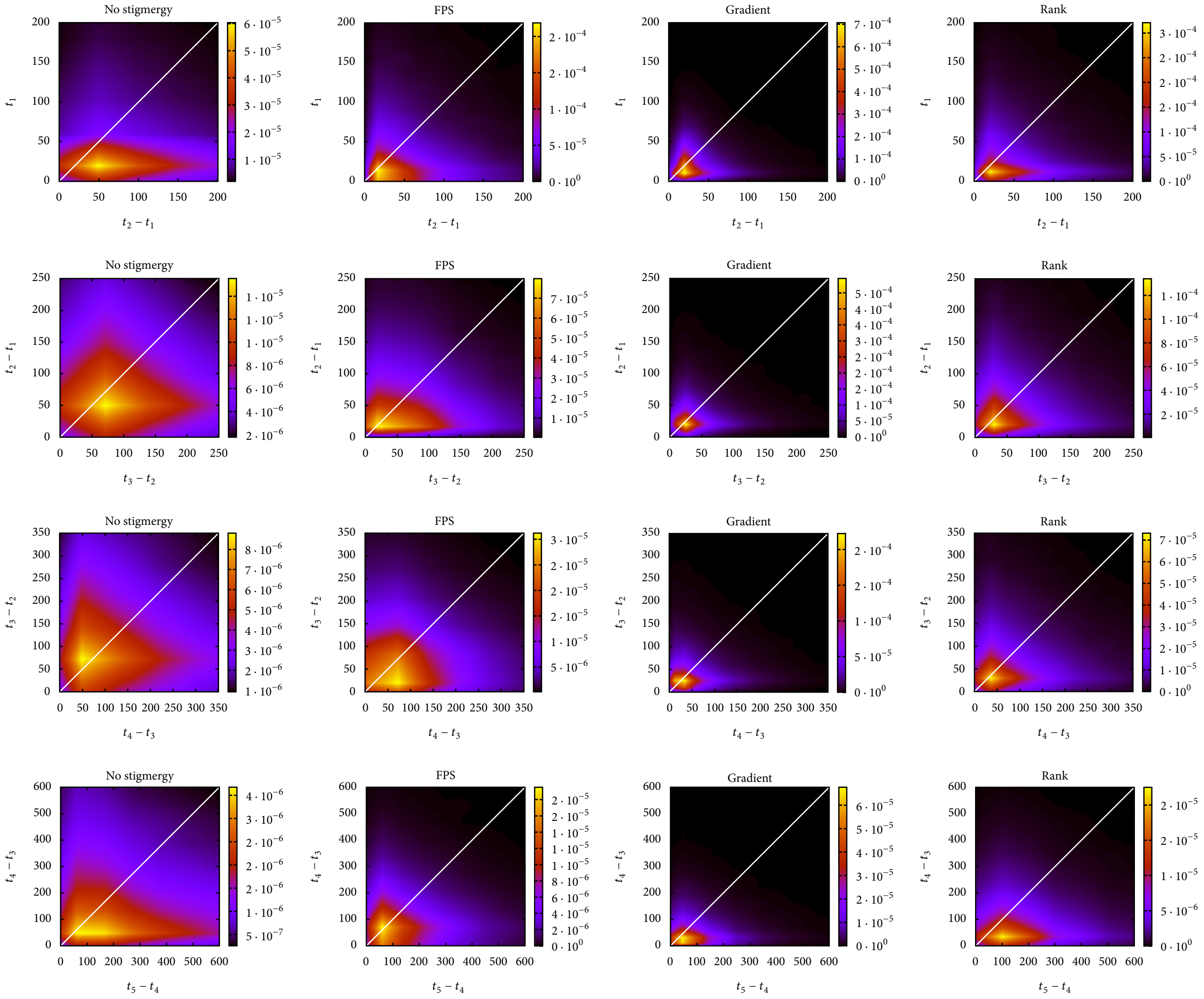


Figure 6: Time differences pdf between consecutive event's discoveries: t_1 (first), t_2 (second), t_3 (third), t_4 (fourth), and t_5 (fifth event); the diagonal line are points where $t_{i+1} - t_i$ are equal to $t_i - t_{i-1}$, $i = 0, \ldots, 4$, and $t_0 = 0$ is the start time of the tourist visit.

expectation of having POIs without visits. In particular, with no stigmergy, POIs have a mean number of visits of 34.55, and the maximum value (mode) of probability (>15%) corresponds to the bar centered at 25 visits. Based on the mean values of Table 6, rank method is between FPS and gradient, whereas, by contrast, rank selection has a reduced probability of undiscovered POIs (i.e., with 0 visits). In the case of rank routing, the alternative paths at crossroads have less different fitness value than with FPS. Therefore, the selections are less deterministic than with FPS, and tourists do not always choose the path with the highest pheromone concentration.

Finally, Table 6 shows the mean number of POIs discovered by tourists. Similarly to the static environment, with dynamic events tourists can discover more POIs assisted with stigmergy than without it. In particular, FPS is 54% and gradient 55%, which doubles the result of not using stigmergy, 27%. In comparison with the static events scenario, stigmergy methods achieve a lower ratio. The reduction of the discovery ratio is because of the continuous changes of the pheromone trails, due to the dynamic appearance of POIs. However, the detection ratio on no stigmergy tourists is higher than in the static simulation. The computation of the ratio is similar in the dynamic and static scenario: at the fifth event discovered the simulator records a sample of the ratio; this occurs faster in the dynamic scenario (events may appear in the path of the tourist). Because of this, the number of POIs discovered can, for example, only be one but the number of total POIs was just three, in which case the sample of the ratio is high (this example, 33.3%). In the static scenario, events are at permanent locations, and unguided tourists take more time to reach the fifth event and compute the detection ratio than in the dynamic scenario. For this reason, in the static scenario the number of total POIs is high, and thus the mean ratio remains small (4.61%).

A high percentage of POIs in FPS and gradient receive a large number of visits (some POIs with almost 100% of visitors), because of the indirect communication among

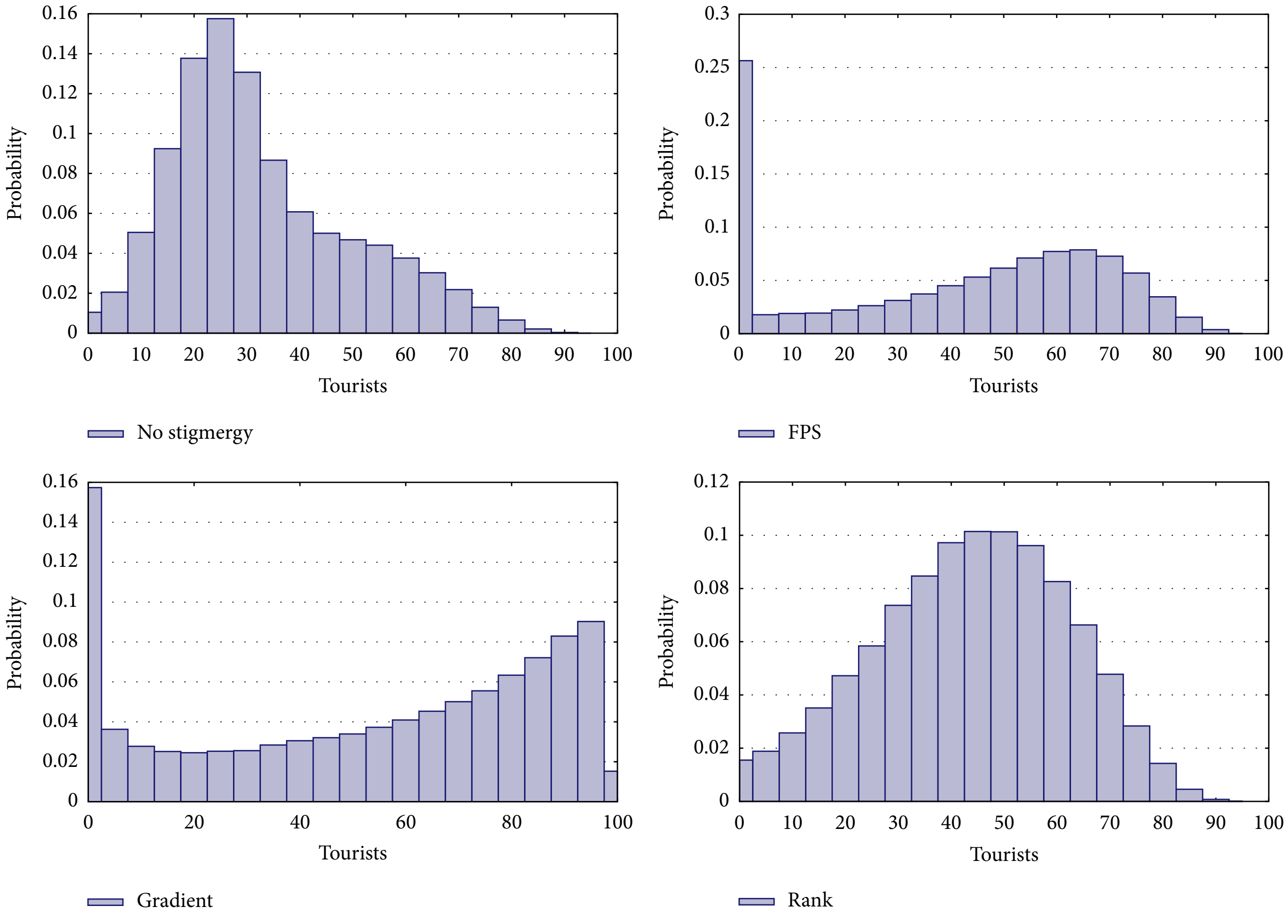


Figure 7: Distribution of visits per POI samples in the dynamic scenario; dataset obtained with simulation parameters of Table 5.

Table 5: Configuration parameters for the dynamic simulation.

Tourists	100
Tourist speed	5 Km/h
Speed variation	0.3%
Initial number of dynamic events	7
Pheromone increment	1 unit
ER	15%
MPI (max)	1 h
POI-DI	8 min
POI-AD	40 m
PV	15 m
Event decrease factor	1.5%
Events arrival rate	Poisson $\lambda = 0.005$
Lifetime of events	Exponential $1/\mu = 55$ min
Events-per-simulation	300
Total event samples (each. mob. model)	114000

tourists through stigmergy. The downside is the increase in the events that end with no visits. On the contrary, ranking selection performs better to find most of the locations (at least for a certain number of tourists) and many POIs have a high percentage of visits. Therefore, ranking selection has the advantage of motion without stigmergy, because it can find almost any event, and the benefit of stigmergy-based methods, because it can guide more tourists to locations of interest.

Table 6: Statistics for visits per POI and percentage of POIs detected by each tourist.

	No stigmergy	FPS	Gradient based	Rank selection
$\overline{x}$	34.55	40.18	53.81	45.59
s	17.17	29.14	34.40	18.47
Md	31	47	61	47
$\overline{x}$-Md	3.55	−6.82	−7.19	−1.41
Mean POI detection ratio	27%	54%	55%	41%

6. Conclusions

This paper presents ANT, a navigation system for tourists. The problem with current applications is that they do not offer real time information or they have to be continuously monitored by the user. The ANT system allows tourists to discover new places and share them with other people, while providing navigation to those locations. Moreover, the system reduces interaction with the user to avoid distraction from the experience of exploring a city.

ANT enables unique features for tourist navigation by using an artificial stigmergy algorithm. It allows the discovery of highly active temporary functions, beyond the functionality of most current navigation applications. The core of ANT is a route ranking mechanism, able to make decisions about the most likely directions. We have studied three techniques: Fitness Proportionate Selection, Rank Selection, and Gradient. Stigmergy-based routing obtains prominent results compared with nonstigmergy discovery. In particular, stigmergy has improvements of 77.3% in the time to discover five events, and of 38% in the number of visits per POI. In the two scenarios (static and dynamic) gradient excels compared to other schemes offering the shortest discovery times and highest event detection probability. In the dynamic event scenario, we have found that the rank selection method allows tourists to discover most of the events. The results also corroborate the benefits of stigmergy to detect live promoted POIs.

For our future work, we aim to implement a prototype and test it in the field with users. Moreover, we intend to develop haptic devices which enable seamless navigation without the need to look at any display. We will also attempt to evaluate user grouping through clustering techniques to weight pheromone IPMs to support differentiated user groups.

Conflicts of Interest

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

Acknowledgments

This research has been supported by the project AIM, ref. TEC2016-76465-C2-1-R (AEI/FEDER, EU).

References

[1] Eurostat, "Almost 8 out of 10 internet users in the eu surfed via a mobile or smart phone in 2016," 2016, http://ec.europa.eu/eurostat/documents/.

[2] Pew Research Center, "Internet & Technology. Mobile fact sheet," 2017, http://www.pewinternet.org/fact-sheet/mobile/.

[3] Waze, "Traffic maps and navegation by means of social interaction," 2017, https://www.waze.com.

[4] "Heat maps for Foursquare application," 2017, https://www.swarmapp.com/.

[5] E. Hornecker, S. Swindells, and M. Dunlop, "A mobile guide for serendipitous exploration of cities," in *Proceedings of the 13th International Conference on Human-Computer Interaction with Mobile Devices and Services, Mobile HCI 2011*, pp. 557–562, swe, September 2011.

[6] P.-P. Grassé, "La reconstruction du nid et les coordinations interindividuelles chez *Bellicositermes natalensis* et *Cubitermes sp.* la théorie de la stigmergie: Essai d'interprétation du comportement des termites constructeurs," *Insectes Sociaux*, vol. 6, no. 1, pp. 41–80, 1959.

[7] M. Dorigo, E. Bonabeau, and G. Theraulaz, "Ant algorithms and stigmergy," *Future Generation Computer Systems*, vol. 16, no. 8, pp. 851–871, 2000.

[8] D. Gavalas, M. Kenteris, C. Konstantopoulos, and G. Pantziou, "Personalized routes for mobile tourism," in *Proceedings of the 2011 IEEE 7th International Conference on Wireless and Mobile Computing, Networking and Communications, WiMob'2011*, pp. 295–300, chn, October 2011.

[9] D. Gavalas and M. Kenteris, "A web-based pervasive recommendation system for mobile tourist guides," *Personal and Ubiquitous Computing*, vol. 15, no. 7, pp. 759–770, 2011.

[10] W. Wörndl, A. Hefele, and D. Herzog, "Recommending a sequence of interesting places for tourist trips," *Information Technology & Tourism*, vol. 17, no. 1, pp. 31–54, 2017.

[11] D. Herzog, H. Massoud, and W. Wörndl, "Routeme: A mobile recommender system for personalized, multi-modal route planning," in *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*, pp. 67–75, ACM, 2017.

[12] Zheng. W., Liao. Z., and J. Qin, "Using a four-step heuristic algorithm to design personalized day tour route within a tourist attraction," *Tourism Management*, vol. 62, pp. 335–349, 2017.

[13] A. Gionis, T. Lappas, K. Pelechrinis, and E. Terzi, "Customized tour recommendations in urban areas," in *Proceedings of the 7th ACM International Conference on Web Search and Data Mining, WSDM 2014*, pp. 313–322, usa, February 2014.

[14] M. A. Awal, J. Rabbi, S. I. Hossain, and M. M. A. Hashem, "A hybrid approach to plan itinerary for tourists," in *Proceedings of the 5th International Conference on Informatics, Electronics and Vision, ICIEV 2016*, pp. 219–223, bgd, May 2016.

[15] Y. Kurata, Y. Shinagawa, and T. Hara, "CT-Planner5: a computer-aided tour planning service which profits both tourists and destinations," in *Proceedings of the Tourism Recommender Systems*, vol. volume 15, RecSys, 2015.

[16] C. Zhu, J. Q. Hu, F. Wang, Y. Xu, and R. Cao, "On the tour planning problem," *Annals of Operations Research*, vol. 192, pp. 67–86, 2012.

[17] W.-S. Yang and S.-Y. Hwang, "ITravel: a recommender system in mobile peer-to-peer environment," *The Journal of Systems and Software*, vol. 86, no. 1, pp. 12–20, 2013.

[18] K. Meehan, T. Lunney, K. Curran, and A. McCaughey, "Context-aware intelligent recommendation system for tourism," in *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom '13)*, pp. 328–331, IEEE, San Diego, Calif, USA, March 2013.

[19] I. Cenamor, T. de la Rosa, S. Núñez, and D. Borrajo, "Planning for tourism routes using social networks," *Expert Systems with Applications*, vol. 69, no. ISSN 0957-4174, pp. 1–9, 2017, http://www.sciencedirect.com/science/article/pii/S0957417416305693.

[20] F. Ricci, B. Arslan, N. Mirzadeh, and A. Venturini, "Itr: a case-based travel advisory system," in *Proceedings of the European Conference on Case-Based Reasoning*, pp. 613–627, Springer, 2002.

[21] A. Yahi, A. Chassang, L. Raynaud, H. Duthil, and D. H. Chau, "Aurigo: An interactive tour planner for personalized itineraries," in *Proceedings of the 20th ACM International Conference on Intelligent User Interfaces, IUI 2015*, pp. 275–285, usa, April 2015.

[22] L. Baltrunas, B. Ludwig, S. Peer, and F. Ricci, "Context relevance assessment and exploitation in mobile recommender systems," *Personal and Ubiquitous Computing*, vol. 16, no. 5, pp. 507–526, 2012.

[23] F. Martínez-Santiago, F. Ariza-López, A. Montejo-Ráez, and A. Ureña-López, "GeOasis: a knowledge-based geo-referenced tourist assistant," *Expert Systems with Applications*, vol. 39, no. 14, pp. 11737–11745, 2012.

[24] J. Rasinger, M. Fuchs, T. Beer, and W. Höpken, "Building a mobile tourist guide based on tourists' on-site information needs," *Tourism Analysis*, vol. 14, no. 4, pp. 483–502, 2009.

[25] P. Kourouthanassis, C. Boletsis, C. Bardaki, and D. Chasanidou, "Tourists responses to mobile augmented reality travel guides: the role of emotions on adoption behavior," *Pervasive and Mobile Computing*, vol. 18, pp. 71–87, 2015.

[26] D. McGookin, S. Brewster, and P. Priego, "Audio bubbles: Employing non-speech audio to support tourist wayfinding," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 5763, pp. 41–50, 2009.

[27] R. A. Abbaspour and F. Samadzadegan, "Time-dependent personal tour planning and scheduling in metropolises," *Expert Systems with Applications*, vol. 38, no. 10, pp. 12439–12452, 2011.

[28] J. Karbowska-Chilinska and P. Zabielski, "Genetic algorithm solving the orienteering problem with time windows," *Advances in Intelligent Systems and Computing*, vol. 240, pp. 609–619, 2014.

[29] S.-W. Lin and V. F. Yu, "A simulated annealing heuristic for the multiconstraint team orienteering problem with multiple time windows," *Applied Soft Computing*, vol. 37, pp. 632–642, 2015.

[30] P. Vansteenwegen, W. Souffriau, and D. Van Oudheusden, "The orienteering problem: a survey," *European Journal of Operational Research*, vol. 209, no. 1, pp. 1–10, 2011.

[31] D. Gavalas, C. Konstantopoulos, K. Mastakas, and G. Pantziou, "A survey on algorithmic approaches for solving tourist trip design problems," *Journal of Heuristics*, vol. 20, no. 3, pp. 291–328, 2014.

[32] M. Rey-López, A. B. Barragáns-Martínez, A. Peleteiro, F. A. Mikic-Fonte, and J. C. Burguillo, "moreTourism: mobile recommendations for tourism," in *Proceedings of the IEEE International Conference on Consumer Electronics (ICCE '11)*, pp. 347-348, Las Vegas, Nev, USA, January 2011.

[33] I. Garcia, L. Sebastià, and E. Onaindia, "On the design of individual and group recommender systems for tourism," *Expert Systems with Applications*, vol. 38, no. 6, pp. 7683–7692, 2011.

[34] N. S. Savage, M. Baranski, N. E. Chavez, and T. Höllerer, "I'm feeling LoCo: A location based context aware recommendation system," in *Advances in Location-Based Services*, pp. 37–54, Springer, 2012.

[35] E. Forcael, V. González, F. Orozco, S. Vargas, A. Pantoja, and P. Moscoso, "Ant Colony Optimization Model for Tsunamis Evacuation Routes," *Computer-Aided Civil and Infrastructure Engineering*, 2014.

[36] O. Zedadra, H. Seridi, N. Jouandeau, and G. Fortino, "S-MASA: A stigmergy based algorithm for multi-target search," in *Proceedings of the 2014 Federated Conference on Computer Science and Information Systems, FedCSIS 2014*, pp. 1477–1485, pol, September 2014.

[37] K. Sastry, D. E. Goldberg, and G. Kendall, "Genetic algorithms," in *Search methodologies*, pp. 93–117, Springer, 2014.

[38] S. Liu, "A hybrid population heuristic for the heterogeneous vehicle routing problems," *Transportation Research Part E: Logistics and Transportation Review*, vol. 54, pp. 67–78, 2013.

[39] C. Gahm, C. Brabänder, and A. Tuma, "Vehicle routing with private fleet, multiple common carriers offering volume discounts, and rental options," *Transportation Research Part E: Logistics and Transportation Review*, vol. 97, pp. 192–216, 2017.

[40] H. Savuran and M. Karakaya, "Efficient route planning for an unmanned air vehicle deployed on a moving carrier," *Soft Computing*, vol. 20, no. 7, pp. 2905–2920, 2016.

[41] N. M. Razali, J. Geraghty et al., "Genetic algorithm performance with different selection strategies in solving TSP," in *Proceedings of the world congress on engineering*, vol. 2, pp. 1134–1139, 2011.

[42] C. Torney, Z. Neufeld, and I. D. Couzin, "Context-dependent interaction leads to emergent search behavior in social aggregates," *Proceedings of the National Acadamy of Sciences of the United States of America*, vol. 106, no. 52, pp. 22055–22060, 2009.

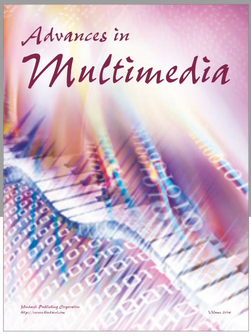
[43] P. De Frenne, B. J. Graae, F. Rodríguez-Sánchez et al., "Latitudinal gradients as natural laboratories to infer species' responses to temperature," *Journal of Ecology*, vol. 101, no. 3, pp. 784–795, 2013.

[44] J. L. Kirschvink, "Sensory biology: Radio waves zap the biomagnetic compass," *Nature*, vol. 509, no. 7500, pp. 296-297, 2014.

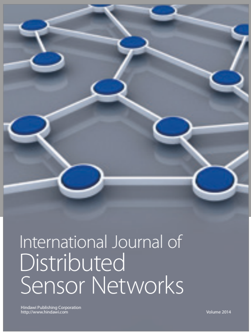
[45] R. A. Ramos, J. Zapata, C. A. Condat, and T. S. Deisboeck, "Modeling cancer immunotherapy: assessing the effects of lymphocytes on cancer cell growth and motility," *Physica A: Statistical Mechanics and its Applications*, vol. 392, no. 10, pp. 2415–2425, 2013.

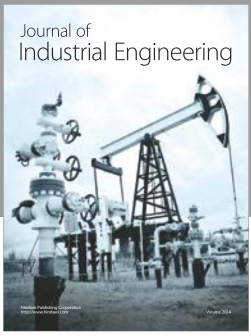
[46] U. Wilensky, *Netlogo. Center for Connected Learning and Computer-Based Modeling*, Northwestern University, Evanston, IL., USA, 1999.

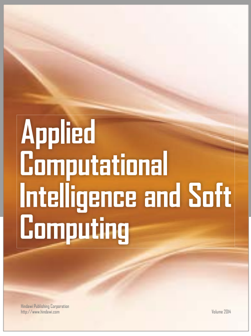
[47] G. Boeing, "OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks," *Computers, Environment and Urban Systems*, vol. 65, pp. 126–139, 2017.

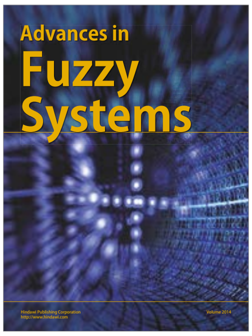

Advances in
Multimedia

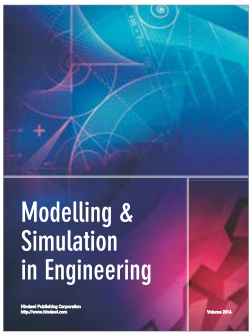
The Scientific World Journal

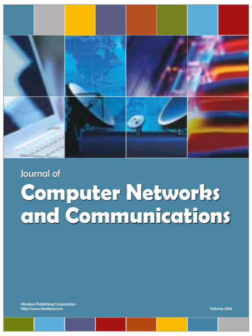

International Journal of
Distributed Sensor Networks


Journal of
Industrial Engineering

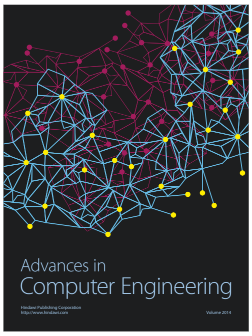

Applied Computational Intelligence and Soft Computing


Advances in
Fuzzy Systems

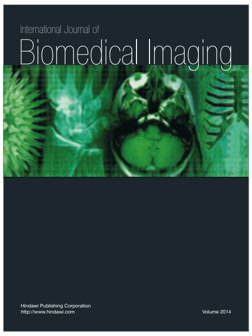

Modelling & Simulation in Engineering

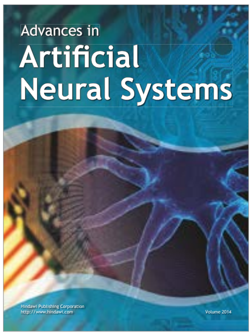

Journal of
Computer Networks and Communications

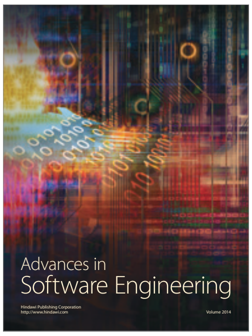
Advances in
Artificial Intelligence


Advances in
Computer Engineering

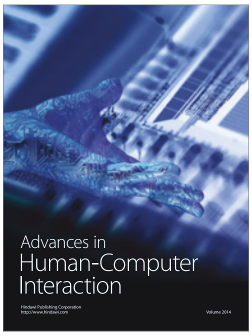
International Journal of
Computer Games Technology

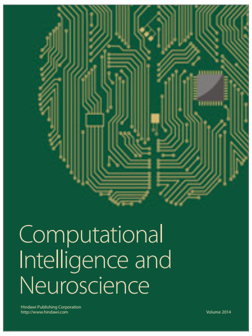

International Journal of
Biomedical Imaging


Advances in
Artificial Neural Systems


Advances in
Software Engineering

Journal of
Robotics


Advances in
Human-Computer Interaction


Computational Intelligence and Neuroscience

International Journal of
Reconfigurable Computing

Journal of
Electrical and Computer Engineering