Bio-inspired Multi-Agent Collaboration for Urban Monitoring Applications*

Uichin Lee†, Eugenio Magistretti♯, Mario Gerla†, Paolo Bellavista‡
Pietro Lió§, Kang-Won Lee⋆

†UCLA ♯Rice University ‡University of Bologna
§University of Cambridge ⋆IBM Research
†{uclee,gerla}@cs.ucla.edu, ♯emagistretti@rice.edu
‡pbellavista@deis.unibo.it, §Pietro.Lio@cl.cam.ac.uk, ⋆kangwon@us.ibm.com

Abstract. Vehicular sensor networks (VSNs) provide a collaborative sensing environment where mobile vehicles equipped with sensors of different nature (from chemical detectors to still/video cameras) inter-work to implement monitoring applications such as traffic reporting, environment monitoring, and distributed surveillance. In particular, there is an increasing interest in proactive urban monitoring where vehicles continuously sense events from streets, autonomously process sensed data (e.g., recognizing license plates), and possibly route messages to vehicles in their vicinity to achieve a common goal (e.g., to permit police agents to track the movements of specified cars). MobEyes is a middleware solution to support VSN-based proactive urban monitoring applications, where the agents (e.g., police cars) harvest metadata from regular VSN-enabled vehicles. Since multiple agents collaborate in a typical urban sensing operation, it is critical to design a mechanism to effectively coordinate their operations to the area where new information is rich in a completely decentralized and lightweight way. We present a novel agent coordination algorithm for urban sensing environments that has been designed based on biological inspirations such as foraging, stigmergy, and Lévy flight. The reported simulation results show that the proposed algorithm enables the agents to move to "information patches" where new information concentration is high, and yet limits duplication of work due to simultaneous presence of agents in the same region.

Key words: Vehicular Ad Hoc Networks (VANET), Vehicular Sensor Networks (VSN), Bio-inspired Data Harvesting, Multi-agent Coordination

1 Introduction

Vehicular Ad Hoc Networks (VANETs) are becoming increasingly popular and relevant to the industry due to recent advances in inter-vehicular communication technologies and decreasing cost of communication devices. Unlike a typical MANET, the networking components in a vehicle have a plenty of computing

* This research is supported through participation in the International Technology Alliance sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defense under Agreement Number W911NF-06-3-0001, and; by ARMY MURI under funding W911NF0510246. Corresponding author: Kang-Won Lee, E-mail address: kangwon@us.ibm.com

and storage capacity. Thus, VANETs are considered one of the most promising forms of MANETs outside the military domain and have recently stimulated promising research ranging from safe cooperative driving to entertainment support and distributed data collection.

In this paper, we are interested in urban sensing for effective monitoring of environmental conditions and social activities in urban areas using vehicular sensor networks (VSNs). Differently from traditional wireless sensor nodes, vehicles are not typically affected by energy constraints and can easily be equipped with powerful processing units, wireless communication devices, GPS, and sensing devices such as chemical detectors, still/video cameras, and vibration/acoustic sensors. We particularly envision *proactive* urban monitoring services where vehicles continuously monitor events from urban streets, maintain sensed data in their local storage, process them (e.g. recognizing license plate numbers), and route messages to vehicles in their vicinity to achieve a common goal (e.g. to allow police agents to pursue the movements of specific cars). However, this requires the collection, storage, and retrieval of massive amounts of sensed data. In conventional sensor networks, data are dispatched to "sinks" and are processed for further use (e.g., Direct Diffusion [1]), but that is not practical in VSNs due to the sheer size of generated data. Moreover, it is impossible to filter data a priori because it is usually unknown which data will be of use for future investigations. Thus, the challenge is to find a completely decentralized VSN solution, with low interference to other services, good scalability, and tolerance to disruption caused by mobility and attacks.

To that purpose, we designed and implemented MobEyes, a novel middleware that supports VSN-based proactive urban monitoring applications [2]. In MobEyes, each sensor node performs event sensing, processing/classification of sensed data, and periodically generates data summaries with extracted features and context information tagged with timestamp and position information. Summaries are then disseminated to other regular vehicles such that mobile agents, e.g., police patrolling cars, move and opportunistically harvest summaries from neighbor vehicles. As a result, agents can create a low-cost opportunistic index which enables them to query the completely distributed sensed data storage, thus answering questions such as: which vehicles were in a given place at a given time? which route did a certain vehicle take in a given time interval?, and which vehicles collected and stored the data of interest? Unlike MobEyes, CarTel [3] utilizes opportunistic connectivity via roadside access points to send queries about sensed data and to return replies "on-demand," instead of "proactive" data collection, which should be definitely preferred in presence of constraints on query resolution latency.

Multiple agents can collaborate in harvesting relevant data, processing them, and searching for key information. It is critical to design a mechanism to effectively coordinate and geographically separate the operation of multiple agents, while allowing them to seek most productive fields in a totally distributed matter. However, multi-agent harvesting is a very challenging problem due to the dynamic nature of the target environment (e.g. continuous creation and movement of metadata) and the scale of operations (e.g. harvesting region ranging over multiple city blocks) without a priori knowledge of the location of the critical information. Incidentally, we note that social animals (ranging from bacteria to vertebrates) solve a similar problem of *foraging* to find a good food source quite efficiently using a simple communication mechanism in a fully distributed manner with lightweight and lazy coordination.

Given this observation, the primary goal of this paper is to design a novel multi-agent coordination mechanism for MobEyes harvesting agents by taking inspirations from biological systems. We realize that each species may have inched towards foraging optimality for specific tasks and various constraints (e.g., habitat niches, animal size and speed, environment, etc.). Therefore we design a mechanism by encompassing different animal foraging and behavioral ecology strategies, instead of focusing on single animal species. The natural scene examples inspiring MobEyes multi-agent coordination include: (a) Foraging behavior of *Escherichia (E.) coli* bacteria that operate in distinct modes of locomotion based on the level of nutrient concentration [4, 5]; (b) Lévy walk behavior of many biological organisms and groups, e.g., albatrosses and fishing boats, to improve food search over large-scale regions [6, 7]; and (c) Stigmergy found in ants and other social insects that use various types of pheromones to signal nest mates with potential conflicts, e.g., a sort of "no entry" sign [8, 9].

Based on this study, we propose a novel harvesting strategy, called *datataxis* (á la chemotaxis of E. coli bacteria), that guides the agents to stay and acquire metadata on "information patches," the regions where newly created and not-harvested metadata are concentrated (based on a simple metric for metadata density estimation per road segment). MobEyes agents adapt their behavior by following a 3-state transition diagram that sometimes forces them to change their area of exploration by using Lévy walk-inspired movement patterns that are considered suitable for the large scale of the typically targeted regions. To avoid harvesting work duplication, agents exploit stigmergy-inspired techniques for conflict resolution to prevent from useless concentration of agents in the same region at the same time.

We validate the performance of our proposed data harvesting scheme via extensive simulations where we use a realistic Manhattan mobility model and compare the harvesting efficiency of our datataxis foraging (DTF) with random walk foraging (RWF), biased random walk foraging (BRWF), and an idealized preset pattern foraging (PPF). From this study, we show that the proposed DTF balances the movement of multiple agents and distributes them effectively without the need of centralized and intrusive coordination protocols.

The remainder of the paper is organized as follows: Section 2 presents a background on the MobEyes urban sensing architecture; Section 3 reviews the foraging behaviors in nature and presents our algorithm for multi-agent coordination; Section 4 presents a simulation-based performance evaluation of various agent coordination approaches; finally Section 5 concludes the paper.

2 MobEyes Vehicular Sensing Platforms

We present the MobEyes solution using one of its possible application scenarios: collecting information from MobEyes-enabled vehicles about criminals who spread poisonous chemicals in a particular section of the city (say, a subway station). We assume that the criminals use vehicles for the attack. In this scenario,

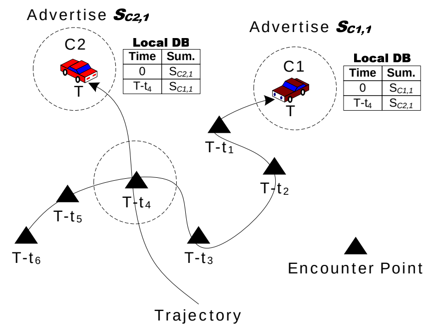


Fig. 1. MobEyes single-hop passive diffusion

MobEyes will help detect the criminal vehicles and permit tracking and capture. Here, we assume that the vehicles participating in MobEyes are equipped with cameras and chemical detection sensors. Vehicles continuously generate a huge amount of sensed data, store it locally, and periodically produce short *metadata chunks* obtained by processing sensed data, e.g., license plate numbers or aggregated chemical readings. Metadata chunks are aggregated in a summary packet that is opportunistically disseminated to neighbor vehicles, thus enabling metadata harvesting by the police to create a distributed metadata index which permits to find a set of vehicles storing data of interest for forensic purposes such as crime scene reconstruction and criminal tracking.

Any regular node periodically advertises a new summary packet with generated metadata to its current neighbors to increase the opportunities for agents to harvest the summaries. A packet header includes a packet type, generator ID, locally unique sequence number, packet generation timestamp, and generator's current position. Each packet is uniquely identified by the generator ID and its sequence number pair, and contains a set of metadata locally generated during a fixed time interval. Neighbor nodes receiving a packet store it in their local metadata databases. Therefore, depending on the mobility and the encounters of regular nodes, packets are opportunistically diffused into the network of vehicles, yet metadata diffusion is time and location sensitive. MobEyes can be configured to perform either single-hop passive diffusion (only the source advertises its packet to current single-hop neighbors) or k-hop passive diffusion (the packet travels up to k-hop as it is forwarded by j-hop neighbors with $j < k$). Figure 1 depicts the case of two sensor nodes, $C1$ and $C2$, that encounter with other sensor nodes while moving (the radio range is represented as a dotted circle). A black triangle with timestamp represents an encounter. For ease of explanation, we assume that there is only a single encounter, but in reality there may be multiple encounters with any nodes that happen to come within the dotted circles. $C1$ and $C2$ periodically advertise a new summary packet $S_{C1,1}$ and $S_{C2,1}$ respectively where the subscript denotes $\langle ID,\ Seq.\# \rangle$. At time $T - t_4$, $C2$ encounters $C1$, and thus they exchange those packets. As a result, $C1$ carries $S_{C2,1}$ and $C2$ carries $S_{C1,1}$.

In parallel with diffusion, MobEyes metadata harvesting may take place. The MobEyes police agent collects summary packets from regular nodes by periodically querying its neighbors. The goal is to collect all the summary packets generated in a specific region. Ideally, a police node should harvest only those summary packets that it has not collected so far. To focus only on missing packets, a MobEyes authority node compares its list of summary packets with that of each neighbor (i.e., a set difference problem), by exploiting a space-efficient data structure for membership checking, i.e., a Bloom filter [10]. A MobEyes police agent uses a Bloom filter to represent its set of already harvested and still valid summary packets and includes this filter when broadcasting a harvest request message [2]. Given this, each neighbor node prepares a list of missing packets. After random back-off, one of the neighbors returns those missing packets to the agent. The agent sends back an acknowledgment with a piggybacked list of returned packets and, upon listening to or overhearing this, neighbors update their lists of missing packets.

Note that each vehicle can piggyback the current position into its summary advertisement, and thus, Last Encounter Routing (LER) can be supported at no extra cost [11]. Enhanced LER with the carry-and-forward to address intermittent connectivity plays a key role in MobEyes when an agent tries to retrieve the actual data, or to send a dump request to the target vehicle.

3 Multi-Agent Information Harvesting

Multiple agents can collaboratively search a given area of interest to collect desired information more rapidly. We design an algorithm to coordinate and control multiple agents to harvest target data as efficiently as possible. In particular, we are interested in designing a simple algorithm that does not involve a tight, close range control of agents' movement, since the latter would incur heavy communication overhead. At the same time, we want the algorithm to be efficient; ideally, we want our algorithm to perform similarly to a centralized coordination algorithm, in terms of data harvesting efficiency (i.e., how fast can we collect all of the interested data) and the control efficiency of agents' movement (i.e., how much redundant data was collected by multiple agents). In addition, we want the algorithm to be able to be self-organizing and adaptive to the dynamics of the environment, such as the changes in the movement patterns, the densities, and the data carried by VSN vehicles. Also some part of the network may exhibit intermittent connectivity; hence, we require our algorithm to be delay tolerant and robust to temporary disconnections.

3.1 Biological Inspirations for Data Harvesting

The main reason for us to look at biological inspiration comes from the observation that the animals and insects encounter a similar problem: they often coordinate their efforts to effectively collect food without prior knowledge of food sources; yet they are known to solve the problem quite effectively, if not optimally [12]. Accordingly to the foraging theory, animals are presumed to search for nutrients and obtain them in a way to maximize the ratio of energy intake over the time spent for foraging. Foraging constraints also shape division of labor in animal societies. This applies to both vertebrate societies where foraging tends to be associated with hunting and is based on individual recognition, and invertebrates (insect) societies which are characterized by a great deal of redundancy. In this section, we review key foraging behaviors in nature that are applied to tackle our problem.

Stigmergy: MobEyes data harvesting is directly related to the food foraging problem solved by stigmergy [9]. Ants need to find routes to possibly ephemeral food sources in an effective manner. Since it is not immediately obvious how long the current site will remain as a valid foraging site, they have to solve a dynamic problem of remembering a rewarding source while exploiting newly discovered food sites. In many cases, the nutrients are distributed in *patches*, and the main issue of foraging is finding such patches, deciding how long it will take before depleting and leaving food sources. The foraging patterns in ants change with increasing prey/food size, showing all stages intermediate between an individual and a mass exploitation of food resources. This suggests that social insects process information and solve problems in a complex environment, while keeping some parsimony at the level of the individuals' decision rules [8]. It has been known that ants can optimize their foraging by selecting the most rewarding source via the following methods. Physical contacts and other forms of direct communication, e.g., via sound or vibrations, are limited both spatially and temporally; only neighbors in the vicinity can receive the signal. On the contrary, *pheromone trails* are long lasting and can be considered a wide broadcast that slowly dissipates in time. Different types of pheromones have evolved in ants. First, there are long-lasting pheromones, used to maintain the spatial organization of ant networks, and volatile pheromones, used to quickly mark routes leading to current food sources. For instance, the pygidial gland of the Ponerine Army Ant *Leptogenys distinguenda* produces a long-lasting trail pheromone (that lasts about 25 minutes), which guides the ants back to the trail or the colony when they are detached from the trail network [13]. Second, there is a short-live *repellent* pheromone, which effectively serves as a no-entry signal.

Chemotaxis of *E. coli*: Another biological foraging behavior that we consider in the context of information harvesting is the chemotactic (foraging) behavior of many bacteria, for example *E. coli* [4]. *E. coli* is representative of a large, widespread class of bacteria, and is present everywhere in the environment and also in the lower intestines of mammals including humans. *E. coli* gets its locomotion from a set of rigid flagella that enables the bacteria to swim. When their flagella turn clockwise, bacteria tumble and do not move to any particular direction. On the other hand, when flagella turn counter-clockwise, the bacteria will swim in a directional movement. The sensors of E. coli are receptor proteins that are stimulated by the binding of molecules in the environment. Based on the level of nutrients (or attractants) a bacterium will move in different modes. More specifically, when an *E. coli* is in some substance without food or noxious substances, its flagella will alternate between moving clockwise and counterclockwise so that the bacterium will alternate between tumbling and swimming. This alternation will move the bacterium in random directions. We can consider this movement mode a *search* for food. If the nutrients have homogenous concentration, the bacteria will exhibit a search behavior but with increased run length of swimming and decreased tumble time. In effect, they will search for nutrients more aggressively when they are in a nutrient environment. Finally, when the bacteria detect a change in the concentration level of nutrition, they will swim along the gradient of concentration toward the most nutrition rich area, and spend less time tumbling. If somehow, an *E. coli* encounters a region where nutrient gradient does not increase after the swim, it will return to the baseline search mode to look for higher concentrations.

Lévy Walk: There is a growing agreement that foraging and movement patterns of some biological organisms may have so-called "Lévy-flight" characteristics. Lévy random walks, named after the French mathematician Paul Pierre Lévy [6], are known to outperform Brownian random walks when the precise location of the targets is not known a priori but their spatial distribution is uniform. A Lévy flight is comprised of random sequences of movement segments, with lengths l, drawn from a probability distribution function having a power-law tail, $p(l) \sim \ell^{-a}$ where $1 < a < 3$. Such a distribution is said to have a "heavy" tail because large-length values are more prevalent than within other random distributions, such as Poisson or Gaussian. Viswanathan et al. demonstrated that $a = 2$ constitutes an optimal Lévy-flight search strategy for locating targets that are distributed randomly and sparsely [14]. Under such conditions, the Lévy search strategy minimizes the average distance traveled and presumably the average energy expended before encountering a target. The strategy is optimal and results in space filling paths, if the searcher is exclusively engaged in searching, has no prior knowledge of target locations, and if the average spacing between successive targets greatly exceeds the searcher's perceptual range.

3.2 Bio-inspired Multi-Agent Coordination in MobEyes

In MobEyes, vehicle mobility is exploited for effective and inexpensive metadata dissemination, i.e., regular cars carry-and-forward metadata to harvesting agents. Therefore, metadata are likely located where the number of vehicles is greater. As an indicator of information concentration, we define the *information density* as the number of metadata carriers, i.e., regular cars actually transporting metadata, in a road segment. We note that our algorithm does not need to depend on this specific metric and can work with any information density metric that can be profitably measured. Like *E. coli* bacteria, our goal is to find a *patch* that contains a large number of "useful" metadata carriers with information not yet harvested by either the same or a cooperating harvesting agent. As a first level approximation, a promising solution for agents is to mimic the foraging behavior of *E. coli* by estimating the gradient of *information density* and moving to a direction where this gradient increases (á la the swim of *E. coli* in a solution with nutrient gradient), while performing a random search when there is no specific gradient (á la the tumble of *E. coli* in a homogeneous environment). We name this bio-inspired behavior of harvesting agents as *datataxis* (inspired by the chemotaxis of *E. coli*).

The key for effective datataxis is to estimate vehicle density in a decentralized way with minimum overhead. To achieve this goal, we propose to divide any road into a set of uniquely identifiable unit distance segments (or "road segments"). Any urban area can be represented as a set of road segments. While MobEyes regular nodes are in a specific road segment, they estimate density of that segment by simply counting the number of their neighbors: this per-

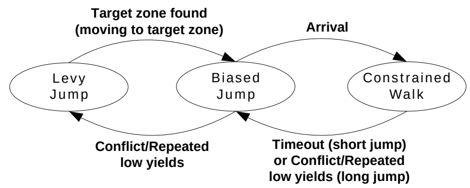


Fig. 2. Agent state diagram

segment density estimation is advertised by the vehicles on that road segment via the regular MobEyes summary broadcast process. Each vehicle only advertises the density information for the road segment it is currently on. In that way, the density information is locally computed and updated. Agents can collect per-road segment density samples, by exploiting the regular MobEyes protocol for summary harvesting, with no additional communication overhead.

However, the model of a simple *E. coli* behavior for all cooperating agents is insufficient to realize effective harvesting of monitoring metadata in urban environments. We have extensively explored bio-inspired coordination behaviors to identify, evaluate, and adopt the most suitable differentiated working modes to obtain high harvesting coverage with minimum overhead. In our design, MobEyes agents operate in one of the following three modes: (a) the Lévy Jump (LJ) mode, (b) the Biased Jump (BJ) mode, and (c) the Constrained Walk (CW) mode. The LJ/BJ modes are considered as the exploration stage to find the best possible location to start a more focused search, whereas the CW mode can be considered as the exploitation stage where agents try to harvest as much as possible by carefully and finely controlling their movements. Figure 2 presents a transition diagram consisting of the three possible states of operation by MobEyes harvesting agents.

First of all, a MobEyes agent starts with the LJ mode and searches for dense areas with vehicles. In the Lévy jump literature, it is known that the jump distance following a power law distribution with the exponent of 2 is known to be optimal for non-destructive foraging, i.e., a foraging scheme where agent can "productively" visit the same place many times [15]. Recall that since vehicles move in the urban grid, it may be very possible that after a while the same area may become "productive" again. The key idea of the LJ mode is that agents can choose a long distance with some probability, due to the heavy tail of the power law distribution. Thanks to the long jumps, the area covered by the agents will be much larger than the area that would have been covered by only random walk movement patterns [15]. Since the network size is finite in our model, we use a truncated Lévy jump distribution: $f(d) = \frac{d_{max} d_{min}}{d_{max} - d_{min}} \frac{1}{x^2}$ where we set the d_{max} as the network diameter and d_{min} as the communication range. The angle of a jump from the current location is selected randomly. For each jump, the agent steers its movement towards the road segment that minimizes the distance to the new jump location. However, for a given location, it may not be feasible to jump toward a certain direction. For instance, if an agent is located at the bottom left corner of the network, a jump is feasible toward the first quadrant. The key idea of a Lévy jump is to have a long jump with some probability for efficient exploration. Thus, we modify the angle selection such that we only consider the region that can span a chosen distance. In the previous example, the jump direction is chosen from the first quadrant.

Once the agent finds a dense area above a certain threshold, the agent changes its operation state to the BJ mode so that it can move toward that location. The target location is the mid-point of the densest road segment, which is also set as the reference point of the CW mode that will be used by the agent as described below. The agent steers its movement towards the road segment that minimizes the distance to the determined reference point (i.e., a simple greedy movement).

When entering the CW region (the circular area with center the reference point and radius R), the agent switches its mode to the CW mode and starts harvesting metadata within that region. The default choice in MobEyes is to automatically set the distance parameter R as a function of the number of agents and the size of the overall search area. MobEyes supports two operating sub-modes for an agent in the CW state. First, the agent follows the road segment that maximizes the positive per-segment density change. In this case, since we exclude the current road segment from the candidate road segment for the next movement, it is possible that the rate change may be negative. If this occurs, the harvesting agent chooses the road segment that minimizes the change. Second, the agent can follow a biased random walk along a set of road segments in the vicinity; the set consists of the segments with density greater than a configurable threshold. If the explored urban area has the shape of a long strip, staying within a CW region could be inefficient. For this reason, the MobEyes agent periodically performs short range jumps to explore the nearby area after CW duration T_{cw}, thus changing its reference point. To avoid the worst case of continuous jumping around a region where there is not much gain, after a configurable threshold, the agent performs a long jump to a random direction, and switches its mode to the LJ mode to collect the density information again as in the initial phase (i.e., repeated low yield case). This behavior is repeated until the harvesting procedure has ended.

One crucial issue in multi-agent harvesting is to coordinate the movements of cooperating agents. Ideally, we want the agents to direct themselves in the richest information areas while not stepping other agents' toes. In other words, each agent coverage area should be non-overlapping with the others and, when agents encounter each other, one of them should be able to quickly move to a different non-overlapping region. To this end, similar to the pheromone trail left by ants, a harvesting agent leaves a trail on the regular vehicles while collecting metadata. The trail information will contain the ID of the collecting agent and the timestamp of data collection. Thus, agents can detect a conflict via metadata harvesting. For conflict resolution, an agent with lower ID will perform a long jump to a random location that is outside the CW region of the conflicting agents. If it finds an information patch, the constrained random walk begins; otherwise, the LJ mode will be initiated, and the overall process starts over.

4 Evaluation

We evaluate the proposed metadata harvesting algorithm by simulation using ns-2.[1] Mobile nodes communicate using IEEE 802.11 with fixed bandwidth of

[1] http://www.isi.edu/nsnam/ns

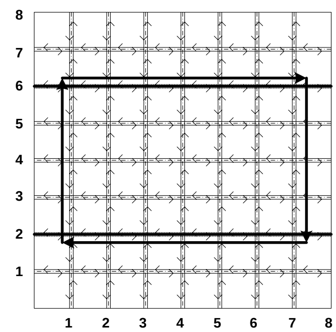
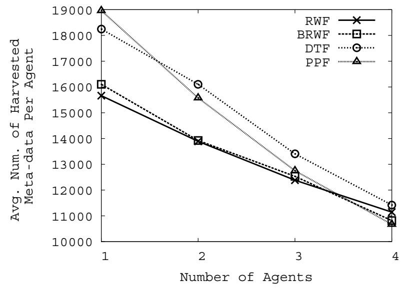


Fig. 3. Street map used for the Manhattan mobility model: Horizontal streets 2 and 6 (marked with thick solid lines) are initially populated (dense streets). The regular mobility pattern (clockwise directional cycle marked by thick areas) is traveled by agents in the PPF strategy.

11Mbps and nominal radio range of 250m. Vehicles move in a fixed region of size $2400m \times 2400m$ according to the Manhattan mobility model (MT) from [16]. In MT, nodes are moving on the streets defined by a map (Figure 3). At each intersection, vehicles make independent decisions about the next direction; the choice of direction (straight, left, right) is equally probable. We use 7x7 grids (each grid segment is set to 300m to avoid interference between nearby streets). We populate two horizontal streets, Street 2 and Street 6, with vehicles by controlling transition probability (i.e., make left or right turns with probability 0.1, and go straight with probability 0.8). When nodes reach the boundary of the simulated region, they bounce back by inverting their direction (modeled by forcing U-turn with probability 1). If this happens, we reset the node and treat it as a new incoming node that carries no meta-data. We consider the number of nodes $N = 200$, and the maximum speed $v = 20$m/s. We fix the speed of harvesting agents to a constant (10m/s).

We evaluate the following foraging schemes by agents: (a) Random Walk Foraging (RWF), (b) Biased Random Walk Foraging (BRWF), (c) Preset Pattern Foraging (PPF), and (d) Datataxis Foraging (DTF). Agents in RWF randomly choose road segments and harvest metadata from encountered vehicles. Agents in BRWF operate similarly except that they choose road segments based on a defined transition probability that is biased by knowledge about "food sources" (i.e., Street 2 or 6). In PPF, we define a preset mobility pattern representing that the agents are fully aware of the movement patterns of others, and thus, we configured the agents to move around the rectangular path that includes Streets 2 and 6. The PPF foraging strategy represents the optimal agent movement in our scenario since the agents will cover the most popular streets using this mobility pattern. DTF implements our proposed scheme for agent movement while harvesting metadata. An agent explores a region while in the Lévy Jump mode to estimate meta-data density per road segment, and switches its mode to the CW mode. After moving to an information patch, an agent stays there for 300s (in CW mode). An agent performs short jumps within a CW region, where the radius of the CW region is set to 600m. If it finds another region after the



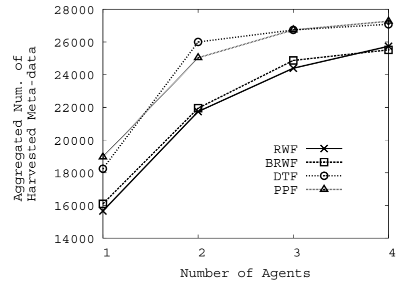


Fig. 4. Number of harvested summaries per agent

Fig. 5. Aggregated number of harvested summaries

jump, a conflict is detected, and it performs a long jump, where the maximum jump range is set to 900m. The number of agents used in the simulation varies from 1 to 4 nodes.

We set the summary packet advertisement period of regular nodes and the harvesting request period to 3s in all the simulations. A new summary is generated based on a Poisson process with rate $\lambda = 1/10$ (i.e., on average it is generated every 10s). We measure the performance in terms of the total number of harvested summary packets. If multiple agents are used, we also calculate aggregate harvesting rate. When calculating the harvesting rate, we only count the number of distinct summaries harvested. For a given scenario, we report the average values of 30 runs, each of which takes 1500s (i.e., 25 minutes).

Simulation Results: Figure 4 reports the impact of the number of foragers on harvesting performance, by showing the number of harvested summaries per agent. In general, the graph shows that the value decreases as the number of agents increases because we only count unique summaries. We note that BRWF shows only a slight improvement over RWF. This stems from the fact that once the agents in BRWF deviate from the popular streets, it takes a long time for them to return to productive areas. The performance of DTF is consistently better than RWF and BRWF, and quite close to PPF. Recall that PPF is a foraging strategy specifically and statically optimized to our target deployment scenario and is expected to represent a quasi-optimal solution. Thus, we find that our DTF algorithm is efficient, not far from the performance achievable via static knowledge of the characteristics of the considered deployment environment. Figure 5 shows the total number of distinct summaries harvested by all the agents. In this plot we also find that the aggregate harvesting ratio of DTF is much better than both RWF and BRWF, and very close to PPF.

5 Conclusion

In this paper, we presented a novel data-harvesting algorithm for urban monitoring applications. The proposed algorithm has been designed based on biological inspirations such as (a) foraging behavior of *E. coli* bacteria, (b) stigmergy found in ants and other social insects, and (c) Lévy flights found in foraging and general movement patterns. The proposed algorithm called *datataxis* enables the

MobEyes agents to move to “information patches” where new information concentration is high. This algorithm is guided by a practical metric based on local efficient estimates of information density per road segment. In our data foraging strategy, an agent starts with a random walk until it encounters an information patch; then it performs a constrained walk to move toward a higher density region. When an agent encounters some other agents in the same region it moves to another region by using a conflict resolution algorithm that has been inspired by Lévy jump, so that harvesting agents' work is not duplicated. Our simulation results showed that datataxis effectively balances the movement of agents and distributes them appropriately, performing better than other commonly used harvesting strategies.

References

1. Intanagonwiwat, C., Govindan, R., Estrin, D.: Directed Diffusion: a Scalable and Robust Communication Paradigm for Sensor Networks. In: ACM MOBICOM'00, Boston, MA, USA (2000)
2. Lee, U., Magistretti, E., Zhou, B., Gerla, M., Bellavista, P., Corradi, A.: MobEyes: Smart Mobs for Urban Monitoring with a Vehicular Sensor Network. IEEE Wireless Communicationsg **13**(5) (Sept. 2006)
3. Hull, B., Bychkovsky, V., Chen, K., Goraczko, M., Miu, A., Shih, E., Zhang, Y., Balakrishnan, H., Madden, S.: CarTel: A Distributed Mobile Sensor Computing System. In: ACM SenSys, Boulder, CO, USA (Oct.-Nov. 2006)
4. Passino, K.M.: Biomimicry for Optimization, Control and Automation. Springer-Verlag (2005)
5. Kim, T., Jung, S., Cho, K.: Investigations into the Design Principles in the Chemotactic Behavior of Escherichia Coli. Biosystems (2007)
6. Levy, P.: Theorie de l'addition des variables aleatoires. Gauthier-Villars, Paris (1954)
7. Reynolds, A.M., Frye, M.A.: Free-Flight Odor Tracking in Drosophila Is Consistent with an Optimal Intermittent Scale-Free Search. PLoS ONE **2**(4) (2007)
8. Detrain, C., Deneubourg, J.L.: Complexity of Environment and Parsimony of Decision Rules in Insect Societies. Biol. Bull. **202**(3) (Jul. 2002) 268–74
9. Theraulaz, G., Bonbeau, E.: A Brief History of Stigmergy. Artificial Life **5**(2) (1999) 97–116
10. Bloom, B.H.: Space/Time Trade-offs in Hash Coding with Allowable Errors. CACM **13**(7) (Jul. 1970) 422.426
11. Grossglauser, M., Vetterli, M.: Locating Nodes with EASE: Mobility Diffusion of Last Encounters in Ad Hoc Networks. In: IEEE INFOCOM, San Francisco, CA, USA (Mar.-Apr. 2003)
12. Stephens, D., Krebs, K.: Foraging Theory. Princeton University Press, NJ, USA (1986)
13. Jackson, D.E., Ratnieks, F.L.: Primer: Communications in Ants. Current Biology **16**(15) (2006) 570–574
14. Bartumeus, F., da Luz, M.G.E., Viswanathan, G.M., Catalan, J.: Animal search strategies: a quantitative random-walk analysis. Ecology **86** (2005) 3078–3087
15. Viswanathan, G., Buldyrev, S.V., Havlin, S., da Luz, M.G.E., Raposo, E.P., Stanley, H.E.: Optimizing the success of random searches. Nature **401** (1999) 911–914
16. F. Bai, N.S., Helmy, A.: The IMPORTANT Framework for Analyzing the Impact of Mobility on Performance of Routing for Ad Hoc Networks. Ad Hoc Networks Journal - Elsevier **1** (Nov. 2005) 383–403