Hindawi
Complexity
Volume 2020, Article ID 8880730, 12 pages
https://doi.org/10.1155/2020/8880730

Research Article

Self-Administered Information Sharing Framework Using Bioinspired Mechanisms

Huibo Bi, Yanyan Chen, Wen-Long Shang, Chengcheng Song, and Wenbo Huang

Beijing Key Laboratory of Traffic Engineering, College of Metropolitan Transportation, Beijing University of Technology, Beijing 100124, China

Correspondence should be addressed to Wen-Long Shang; shangwl_imperial@bjut.edu.cn

Received 7 August 2020; Revised 6 October 2020; Accepted 18 October 2020; Published 30 October 2020

Academic Editor: Zhile Yang

The promising potential of distributed and interconnected lightweight devices that can jointly generate superior information-collecting and problem-solving abilities has long fostered various significant and ubiquitous techniques, from wireless sensor networks (WSNs) to Internet of Things (IoT). Although related applications have been widely used in different domains in attempting to collect and harness the ever-growing information flows, one major issue that impedes the further advancement of WSNs or IoT-based applications is the restricted battery power. Previous research mainly focuses on investigating novel protocols to save energy by reducing data traffic with the aid of optimal or heuristic algorithms. However, data packet behaviours and significant parameters involved are mostly preconfigured in a supervised-learning fashion rather than using an unsupervised learning paradigm and therefore may not adapt to uncertain or fast-changing environments. Hence, this paper concentrates on optimising the behaviours of data packets and significant parameters in a widely tested routing protocol, namely, Cognitive Packet Network (CPN), with the aid of several bio-inspired algorithms to increase the efficiency of energy usage and information acquisition. Two novel packet behaviours are introduced, and an on-line parameter calibration scheme is proposed to realise packet time-to-live (TTL) adjustment and rate adaptation. The simulation results show that the introduction of the bioinspired algorithms can improve the efficiency of information sharing and reduce the energy consumption.

1. Introduction

Energy-efficiency-related studies in the combination with machine learning approaches have been a long-standing focus in computer-aided systems such as vehicular systems [1–3], surface vessel systems [4, 5], manufacturing systems [6, 7], evacuation systems [8, 9], and communication systems [10, 11]. Among these systems, a key component would be information sharing system, which is used to disseminate useful information among users without being inundated by the less useful information [12]. Owing to their long term monitoring ability and easy access to immediate environment, WSNs and the recent emerging IoT are the ideal framework for information sharing and actually have been involved in a vast number of information-sharing related applications such as emergency management, transportation surveillance, medical care, and field monitoring. Since WSNs and IoT are generally formed by battery-powered devices, various protocols are designed to improve the energy efficiency in information acquisition and transmission. However, in most previous protocols, significant parameters such as packet rate or time-to-live of packets are commonly fixed or preconfigured and can cause unnecessary power utilisation in complex and fast-changing environments. Moreover, the configuring and routing algorithms in the optimal protocols are mostly computationally complex and time-consuming. The recent review in [13] has highlighted the immense potential of self-aware computer networks which can self-configure and self-adapt with the aid of software defined networks and reinforcement learning techniques. Therefore, in this paper, we propose several bioinspired reinforcement learning like algorithms to adaptively reconfigure the key parameters of an IoT-based information sharing system and adjust search behaviours of packets to realise energy efficiency and also improve the efficiency in sensing interested events. These optimisations are constructed on top of a widely-used routing protocol, namely, Cognitive Packet Network (CPN). To evaluate the performance of the mechanisms, we employ a simulated IoT-based emergency navigation system to obverse the energy expenditure and the efficiency in delivering emergency messages for evacuees in fire-related disaster scenarios.

The remainder of the paper is organised as follows: Section 2 presents the related work on parameter optimisation in WSNs and IoT-based applications as well as the bioinspired algorithms in information sharing systems. Then, we recall the concept of CPN in Section 3 and introduce the bioinspired algorithms in Section 4. The simulation tool and assumptions of experiments are depicted in Section 5, and the results are presented in Section 6. Finally we draw conclusions in Section 7.

2. Related Work

2.1. Parameter Optimisations in WSNs and IoT. A vast amount of specifically designed routing protocols have been proposed in various WSNs and IoT based applications due to power constraints and long-term unattended operations. The key factor that impacts the design of WSNs and IoT-related protocols is the underlying data delivery patterns, which can be classified into three types: continuous pattern, event-driven pattern, and query-driven pattern or hybrid [14, 15]. In continuous pattern, sensor nodes gather sensory data and transmit to sink nodes or users periodically; in event-driven pattern, sensor nodes gather sensory data and transmit to sink nodes or users when an event occurs; in query-driven pattern, sensor nodes gather sensory data and respond to users when receiving a query. In this paper, we focus on the literature related to continuous pattern since the performance of the latter two patterns is heavily affected by specified scenarios. The study in [16] models the life time of a WSN as a function of data transmission rates among sensor nodes and then formulates the data packet routing problem as a linear programming problem; by solving this problem and maximising the life time of the WSN with a proposed flow augmentation algorithm, desired data transmission rates can be obtained. Similarly, the work in [17] proposes a distributed energy efficient routing algorithm for a WSN to maximise the operation time before the first node drains out of power; this routing problem is constructed as a linear program with the objective of maximising the minimal operation time of all sensor nodes in the WSN; by solving this problem with dual decomposition and subgradient algorithms, the optimal information flow rates among different nodes are determined. One major limitation of the above algorithms is the assumption that all data-routing requests are preknown. The research in [18] and the extended work [19] proposes a data transmission rate control algorithm for WSNs with respect to the node congestion level and data priority; the data transmission rate of a node is adjusted based on two factors: (1) the difference between its current input rate and its maximum allowable rate; (2) the priority of a node, which is determined by the data class priority and the geographical priority; the transmission rate of the WSN is calculated from the sink node to other nodes in a backpropagation manner. The work in [20] designs a multispeed and multipath packet delivery mechanism for WSNs to guarantee the quality of service (QoS) in timeliness and reliability domains; the speed of packets is calculated by the division of geographical distance between nodes by the transmission delay, and a compensation mechanism is triggered to drop packets probabilistically when the speed is below a threshold; the reliability of a path is determined by the packet loss probability of traversed nodes, and a multipath routing mechanism is used to multicast packets to multiple neighbour nodes to improve the reliability. By making use of the spatial and temporal redundancy and correlation of the sensory data (e.g., sensors within an area are considered as spatially correlated and data transmitted within a time period is temporally related), the work in [21] utilises a context-aware entropy filtering model to regulate the transmission rate of nodes with respect to their information type, information priority, and correlation with previous data; to achieve energy efficiency, a node can be adapted among various states including active, dormant, nomadic, stand-by, and discovery based on the periodic and event packet queue length of itself and ancestor nodes. The research in [22] employs a Brownian motion model [23] to evaluate the appropriate time-to-live of packets in a wireless network where packets are sent to search the destination without any priori knowledge; common phenomena such as packet loss are considered and the results indicate that a judicious choice of time-to-live of packets can effectively minimise the average packet travel time. The work in [24] proposes a game theoretical algorithm to self-configure the transmission power of IoT devices; instead of maximising quality of service (QoS) of IoT devices, a constrained Nash equilibrium is introduced to achieve satisfactory QoS levels to reduce energy consumption. The study in [25] proposes a lightweight context-aware service discovery algorithm to search services in a highly dynamic IoT network; instead of considering the service discovery problem as an optimisation problem, a plain mode-switching algorithm is used to adjust the transition behaviours of nodes among advertising, updating, and idle modes.

In summary, previous approaches mainly focus on optimise one or several key parameters (e.g., packet rate) in the routing process. The employed optimisation algorithms tend to configure the targeted WSN or IoT from a global point of view. These algorithms can therefore achieve energy efficiency at the expense of high communication or computation cost. To the best of our knowledge, there was not a comprehensive mechanism to self-configure all the key parameters in a network.

2.2. Bioinspired Algorithms in Information Sharing-Related Systems. As an immense source of inspiration for system design, various biological systems such as nervous system, evolutionary system, ant colony, immune system, and cell organization [26] have been investigated to optimise information sharing-related systems. For instance, the work in [27] proposes a genetic algorithm-aided one-class support tucker machine to detect outliers in large-scale IoT sensory information; in comparison with vector-based anomaly detection methods, anomalies in high-order sensory data can be accurately detected while retaining the origin structure of the data. Inspired by the growth process of the biological brain, the study in [28] proposes a self-cognitive learning framework for general system optimisation; a genetic algorithm is utilised to gradually improve the learning ability of the framework over generations. The work in [29] presents an automatic learning algorithm to predict future events in smart spaces; a reinforcement learning like "autobiographical memory" is proposed to generate cause/effect relationships of users and the environment. The research in [30] employs an inter-cell signaling model and the Kuramoto synchronization model to develop adaptive and flexible server node behaviours for load-balancing in a heterogeneous server system; the unnecessary load transfers among server nodes are avoided and the parameter conditions for the stability and convergence of the system are given and proved. The review in [31] presents various genetic algorithms, ant colony algorithms, bee life algorithms, particle swarm algorithms, immune algorithms, and human brain-inspired fuzzy logic algorithms that are utilised in routing and information-sharing of Vehicular Ad Hoc Networks; compared with traditional counterparts, bio-inspired algorithms are more efficient and robust in large-scale networks. The work in [32] employs a bioinspired mechanism, namely, "reaction-diffusion," to optimise the spectrum allocation for information sharing in a cognitive IoT network; IoT nodes distributedly form clusters which are determined by the competition of the activator and the inhibitor in the reaction-diffusion mechanism. The research in [33] adopts a feed-forward neural network model to a wireless sensor-actuator network (WSAN) for evacuation routing; all physical nodes in the WSAN deploy a neural network with identical topology: an input layer, a hidden layer, and an output layer; the input layer receives the latest two coordinates of a pedestrian, and a suggested direction is subsequently generated by the output layer; the neural networks are trained with the backpropagation algorithm in standard situations and are deactivated when an emergency happens. The work in [34] employs a genetic algorithm to minimise total evacuation time, travel distance, and number of congestion encountered in an evacuation process; non-domination sorting [35] is used as no priori knowledge and is available to determine the weight of the three goals; the initial chromosomes are paths found by the k-th shortest path algorithm [36] and are incrementally evolved to feasible solutions through crossover and mutation operations. The study in [37] employs a variation of particle swarm optimisation (PSO) to search routes and adjust velocity during evacuation; occupants are viewed as particles to search exits; once an exit is discovered, all the other particles will move towards it while keep their moving inertia to expand searching space; if more than one exit is found, particles will choose the nearest exit as the destination. Inspired by the bee colony foraging behaviour, the work in [38] uses bee colony optimisation to displace evacuees in hazardous areas to safe areas during an emergency evacuation; hives, food sources, and bees represent safe areas, hazardous areas, and evacuees, respectively; evacuees select a safe area with regard to "attractiveness" which is determined by the distance to the area and the distribution of people in hazardous areas; once an evacuee determines a target, it will recruit other evacuees by sharing information of the devoted area.

3. Cognitive Packet Network

The Cognitive Packet Network (CPN) [39–41] is a quality of service- (QoS-) driven protocol that was originally proposed for route-finding in large-scale packet networks as well as tailoring diverse QoS for end users in multimedia networks. CPN is composed of CPN nodes in which a mailbox (MB) is maintained to store the discovered routes. CPN contains three types of packets, smart packets (SPs), acknowledgements (ACKs), and dumb packets (DPs). SPs are responsible for path-finding and information-gathering. Each SP carries a cognitive map as well as executable code and can control its own behaviours. The search direction of SPs is determined by either Random Neural Networks (RNNs) [42–47] maintained at each CPN node or a random walk behaviour controlled by "drift parameter." The drift parameter is defined as the possibility for a SP to choose the next hop at random over the RNN's advice. When a SP reaches the destination, it will generate an ACK to bring back all the collected sensory information. DPs carry the payloads and always follow the top-ranked route in the mailbox discovered by SPs. In the context of emergency navigation, evacuees are considered as DPs.

4. Bioinspired Mechanisms

Millions of years of evolution has made the animal foraging behaviours become near-optimal solutions of autonomous search and resource allocation. These behaviours are commonly constructed on top of simple but reliable mechanisms and therefore suit contexts with limited computational capacity and energy. The study in [48] presents a comprehensive literature review of biomimetic models based on animal research. Inspired by this work, in this paper, we investigate efficient algorithms and policies to optimise packet behaviours as well as significant parameters used in CPN.

4.1. Optimise Smart Packets' Behaviour. Smart Packets play a vital role in the route discovery and information collection of CPN, and their behaviours contribute significantly to the efficiency of path-finding as well as the energy usage. In the original CPN model, SPs can discover routes efficiently but their identical search behaviour may incur unnecessary power when appropriate paths have been discovered and the network situation stays unchanged. Furthermore, towards network topologies where exits are distributed broadly, it takes time for heuristic algorithms such as CPN or Ant Colony Optimization (ACO) to adapt to an alternative exit when the desired exit is suddenly blocked by fire. Hence, in this section, we consider SPs as social insects and present two new search behaviours to improve the efficiency of energy utilisation and path-searching.

Inspired by the foraging behaviours of ant or bee colony in which most of the foraging force does not search for new food sources but rely on the scouts (communicated via pheromone or "dance language"), a portion of source routed SPs can be generated to measure the existed routes in the mailbox instead of exploring new paths. This mechanism can reduce the energy usage as the SPs do not explore the network. Furthermore, certain interested paths in the mailbox such as the expiring paths will be measured by source routed SPs before being deleted. Hence, the mailbox tends to maintain more routes and is easier to satisfy the specific requirement of diverse categories of evacuees. This policy can also accelerate the reconvergence process of RNN especially in highly dynamic environments.

On the other hand, the work in [49] reports a consistent novelty-seeking behaviour among honey bees to search new food sources and nest sites. A number of food scouts continue to search for new food sources independently even when abundant food has already been discovered. Similarly, nest scouts individually search for potential nesting cavities and cooperatively determine the best one as a backup. This behaviour can improve the efficiency of path-finding in built environments where the exits are distributed broadly (e.g., egresses are distributed on different floors or rescuers may generate temporary exits by using ladder trucks) because current CPN- or ACO-based algorithms may take time to swift the optimal solution in such topologies. For example, "ants" in ACO rely on artificial pheromone to solve problems and optimal solutions end up with more pheromone than mediocre links. However, when a current optimal path is suddenly blocked by fire, ants cannot change the search direction and discover the new optimal route connected to a distant exit immediately because pheromone evaporates at a fixed low rate. Similarly, it is difficult for a SP which is emitted by a node on the ground floor to find the exit on the second floor when its desired exit is on the ground floor (e.g., search as path from node A to node B) as shown in Figure 1. Hence, we propose a novel SP searching mechanism inspired by the novelty-seeking behaviour to search the alternative exits based on other SPs' experience. By searching routes to alternative exits and storing them in the mailbox beforehand, CPN can provide new suggested direction immediately when the former optimal exit is suddenly disconnected. To increase the efficiency of discovering distant exits, when a SP reaches a new node, it will first inquire the local mailbox to search paths to alternative exits, and then it will follow the best one to some extent as shown in Algorithm 1. If each sensor node has full knowledge of the building model, it can also perform a Dijkstra's algorithm to obtain the shortest path to each exit. Otherwise, if the sensor do not have full knowledge of the graph or an egress is newly generated (e.g., rescuers may generate temporary exits by using ladder trucks), the nodes in the vicinity of exits can initially broadcast a message to inform other nodes in the network when new exits are generated as shown in Algorithm 2. Please note that we do not need to ensure the broadcasted messages reach every node as a SP with novelty-seeking behaviour can get clues from the paths in the mailbox of nodes it traverses.

4.2. Optimise the Drift Parameter. The selection of drift parameter is a tradeoff between path-finding efficiency and quality. Actually, previous experiments indicate that low drift value leads to slow but steady discovery while high drift parameter resolves paths quickly but may stagnate with suboptimal solutions [50]. Hence, in this section, we propose a bioinspired mechanism to self-adapt the drift value based on the network topology as well as observation of SPs.

American white pelicans increase degree of coordination when the prey capture rates decrease and vice versa [51]. This foraging strategy that alerts with prey capture rates can benefit the exploration process of SPs. For instance, when ACKs tend to bring back new paths (high capture rates), the drift parameter will be set to a higher value adaptively to disseminate SPs more randomly to expand the searching area. In contrary, when ACKs are prone to bring back paths that have already been stored in the mailbox, the drift parameter will be adaptively assigned to a low value, and SPs will concentrate on monitoring the current paths. If the obtained paths have all been existed in the mailbox, then SPs will act as source-routed packets as mentioned in Section 4.1.

Noda et al. [52] investigate the prey searching behaviours of stout-body chromis (Ss). They usually search for zooplankton in a tortuous pattern within the conventional foraging regions, regardless of presence or absence of prey. In contrast, Ss will move rapidly between the foraging regions without searching. Similarly, each node in the CPN should be able to adapt its own drift parameter in accordance with its surrounding environment rather than setting an equivalent value. A node in a narrow corridor should have a smaller drift parameter for quick path-finding and rapid adjustment to a spreading hazard while a node in a hall which tends to be linked with more routes should have a larger drift parameter.

Based on the above two interested phenomena, we present a novel algorithm to adapt the drift parameter. The initial drift parameter is determined by the number of neighbours sensed by the current node:

$$D_i^{\text{initial}} = \frac{2}{\pi} \arctan\left(N_n\right) - b, \tag{1}$$

where D_i^{initial} is the initial drift parameter of node i, N_i is number of neighbour nodes of node i, and Term b is a constant parameter which is set to 0.4. This ensures when the number of neighbours of a node is 1, the node does not send source-routed SPs (because there is not any path in the mailbox during the initialisation process).

When a new path is discovered or the QoS of an existed path is significantly changed, the drift parameter will increase to explore broader areas. However, it can not exceed a maximum value $D_{\max}$. On the other hand, if ACKs have not brought back new routes for a certain time interval T_m, the drift parameter will decrease with rate R_d similar to the evaporation of pheromone. Hence, the drift parameter at any time instant t can be calculated as follows:

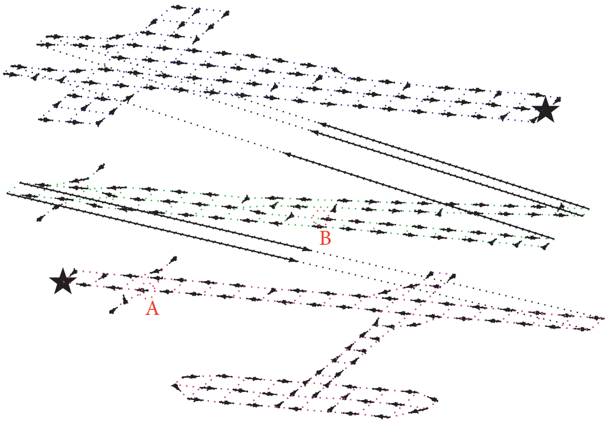

Figure 1: The initial directions given by the CPNSP algorithm before a fire breaks out. CPNSP, CPN based algorithms with distance metric.

```
While a SP with novelty-seeking characteristic reaches a node do
  if the node is not an alternative exit then
    Examine if any route to the alternative exists in the mailbox
    if there are routes to an alternative exist then
      Choose the top-ranked route $P$ and store it in the cognitive map of SP
      Generate a random value $D_u$ between 0 and 1
      if $D_u > D_n$ then
        Choose the next hop in the $P$ to follow
      else
        Choose a random neighbour node as the next hop
      end if
    else
      Choose a random neighbour node as the next hop
    end if
  else
    Stop searching and generate an ACK
  end if
end while
```

Algorithm 1: Search routes to alternative (distant) exits.

$$D(t) = \begin{cases} 0, & D(t) \leq 0, \\ D_{\text{initial}} + R_i * P_n - R_d \cdot 1\left[t^* < T_m\right] * \left(t^* - T_m\right), & 0 < D(t) < D_{\max}, \\ D_{\max}, & D(t) \geq D_{\max}, \end{cases} \tag{2}$$

where $D(t)$ is the current drift parameter, term t represents the current time instant, D_{initial} depicts initial drift parameter, R_i stands for the increased value when a new path is discovered, P_n is the number of discovered paths since the last time $0 < D(t) < D_{\max}$, R_d represents the decrease rate of the drift parameter, term t^* depicts the time cost since $0 < D(t) < D_{\max}$, T_m represents the time threshold after which $D(t)$ begins to reduce, and $1[X]$ is a function that takes the value 1 or 0 if X is false or true, respectively.

4.3. Optimise the Packet Rate. Based on the dynamic drift parameter of each node, we can also optimise the packet rate of SPs. When the SPs of a node have a relatively large drift parameter, more SPs are essentially required to explore the environment. In contrary, when the current drift parameter is small, CPN only demands a few SPs to remeasure and explore the surroundings. If drift parameter is zero, it means that SPs only need to measure the discovered paths. In this case, SPs will be sent to remeasure a route only when the path is nearly expired. The packet rate of SPs is proportional to the drift parameter $D(t)$ when $D(t)$ does not equal to 0:

$$T = k * D(t), \quad \text{if } 0 < D(t) \leq D_{\max}, \tag{3}$$

where k is a constant and $D(t)$ is the real-time drift parameter.

```
When the network is initialised:
for all the node $i$ in the network do
    Set the path to exits $\pi_j^{\text{exit}}$ at node $i$ to null
    where $j = 1, \ldots, N$, $N$ is the number of exits. Term $\pi_j^{\text{exit}}$ depicts the path to the $j$-th exit
end for
When a new exit is generated:
for all the exit $j$ in the network
    Broadcast SPs among the entire network
    When a SP arrives a node, the node will extract the route from the SP
    /* If the previous path is worse than the newly obtained one, then replace it */
    if $\text{QoS}^{\text{new}}_{\pi_j^{\text{exit}}} < \text{QoS}^{\text{old}}_{\pi_j^{\text{exit}}}$
        Then $\pi_{j_{\text{old}}}^{\text{exit}} \leftarrow \pi_{j_{\text{new}}}^{\text{exit}}$
      where $\pi_{j_{\text{new}}}^{\text{exit}}$ represents the route to exit $j$ in the newly reached SP. $\text{QoS}^{\text{old}}_{\pi_j^{\text{exit}}}$ and $\text{QoS}^{\text{new}}_{\pi_j^{\text{exit}}}$ depict the QoS of the previous and new
    routes to exit $j$, respectively. In our treatment, the QoS is number of hops to an exit
    end if
end for
```

Algorithm 2: Disperse messages when a new exit is generated.

4.4. Optimise the Time-to-Live. The number of hops that a SP can traverse before it is dropped has a remarkably influence on the performance of CPN-based systems. If time-to-live of SPs is too large, the system will be overburdened with packets that are in effect lost. On the other hand, if the live-time constraint is too small, some distant exits will not be reached. In this section, we make use of the optimal foraging theory [53] to present a mechanism for SPs to adaptively determine the time-to-live with respect to the surrounding environment. The optimal foraging theory is inspired by animal foraging behaviours in which predators maximise the average intake rate F_b:

$$F_b = \frac{E_a}{H_a + S_a}, \tag{4}$$

where E_a is the average energy obtained from preys, H_a represents the average time cost to capture a prey, and S_a depicts the average searching time to discovery a prey. When a predator finds a prey, if the current intake rate is large than or equals to F_b $((E_c/H_a) \geq (E_a/(H_a + S_a)))$, it will eat it. Otherwise, it will leave to search a new prey.

Based on the optimal foraging theory, we consider nodes as preys and SPs as predators to optimise the number of hops adaptively. SPs will be assigned an initial life constraint H_i with respect to the current average hops of paths in the mailbox. When a SP does not reach an exit after H_i hops during a searching process, it will be considered as a predator and use the animal foraging behaviours to decide whether to continue searching. To employ the optimal foraging theory, each node (prey) in the network is allocated with a value to describe the contained energy. The contained energy can be calculated by taking advantage of the minimal number of hops to the nearest exit which is brought back by ACKs. On top of this, E_a can be calculated by summing the energy obtained from the predecessor nodes. H_a and S_a are represented by the number of hops instead of time. The details of this algorithm are shown in Algorithm 3.

5. Assumptions and Simulation Model

We use the Distributed Building Evacuation Simulator (DBES) [54, 55] which is a multiagent discrete event simulator to evaluate the effectiveness of the bioinspired mechanisms in fire-related scenarios. In the simulator, physical areas are depicted by a graph model: vertices represent locations where evacuees may congregate; edges depict the physical routes between locations. Sensor nodes (SNs) are presumed to have been installed in the vicinity of vertices. We also assume that SNs collect sensory information periodically and provide advices to evacuees. To investigate the influence of building structures to navigation algorithms, two building models which separately represent the lower three floors of Imperial College's EEE building as shown in Figure 2 and a canary wharf shopping mall as shown in Figure 3 are employed.

To construct the energy usage model for SNs, we mimic Mica2 sensor nodes developed by UC Berkeley and make use of the measurements presented in [56]. Our energy model considers energy consumption during communications and computations. The energy consumption during sensing and communications processes is borrowed from [57], while the energy utilisation of CPU is predicted by an analogy of the code-transformation algorithm introduced in [56]. By using this algorithm, we can calculate the CPU active time with regard to the CPU frequency and the recorded total CPU cycles during the simulation. The current levels of CPU in active mode and idle mode is 8.0 mA and 3.2 mA, respectively. The CPU frequency is 16 MHz, and the power supply is 3.0 V. The instruction per cycle (IPC) of the CPU is set to 1. Detailed procedures of this algorithm are as follows:

(1) Convert the codes of the algorithms to simple CPU instructions

(2) Convert involved decimal numbers in the algorithms to binary numbers

```
Process 1: assign contained energy to each prey (node)
for all the node $i$ in the network do
    Set the minimal number of hops to the closest exit $N_{\min}^{\text{hop}}$ at node $i$ to an extremely large value
    When an ACK has arrived, extract the number of hops to the exit in the route which has been brought back by this ACK
    if $N_{\text{current}}^{\text{hop}} < N_{\min}^{\text{hop}}$ then
        Then $N_{\min}^{\text{hop}} \leftarrow N_{\text{current}}^{\text{hop}}$
        where $N_{\text{current}}^{\text{hop}}$ represents the number of hops to the exit in the route brought back by the latest ACK
        Assign the contained energy of node $i$: $E_c = 1/N_{\min}^{\text{hop}}$
    end if
end for
Process 2: optimise the time-to-live of a SP
for all the node $i$ in the network do
    Calculate the average number of hops $N_a^i$ from the obtained paths in the mailbox
    for all the SPs $S^i$ sent from node $i$ do
        Set the total obtained energy $E_t$ to 0 initially
        When a SP $S_j^i$ reaches a new node
        $E_t \leftarrow E_t + E_c$
        /* If the traversed number of hops of the current SP $N_t^S$ is larger than or equals to $N_a^i$ */
        if $N_t^S \geq N_a^i$ then
            if current node is not an exit node then
                /* $E_c$ is the contained energy of the current node, $H_a$ is the average handling time, which is set to 1 (one hop), $E_a$ is the average energy obtained from the current path, which can be calculated by $E_a = E_t/N_t^S$, $S_a$ is the average searching time, which is also 1 (one hop), and term $S_a$ which depicts the average searching time to discovery a prey is set to 0 */
                if $(E_c/H_a) \geq (E_a/(H_a + S_a))$ then
                    Continue to search exits
                else
                    Drop the SP $S_j^i$
                end if
            else
                Generate an ACK
            end if
        end if
    end for
end for
```

Algorithm 3: Determine the time-to-live of a SP.

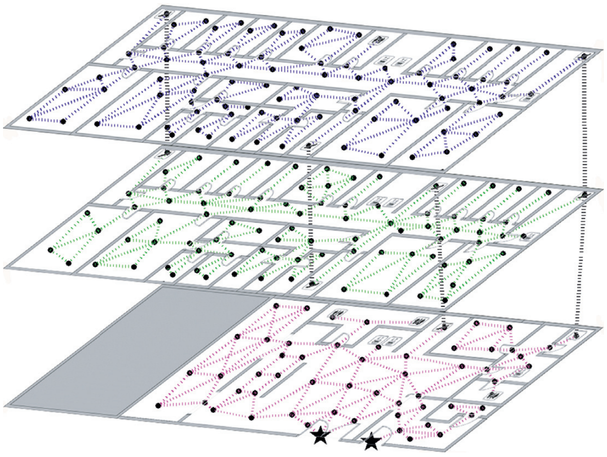

Figure 2: The graph-based layout of Imperial College's EEE building. The black stars represent the exits on the ground floor.

(3) Perform bitwise operations and sum the number of instructions used in computation

(4) Transform the number of instructions to CPU cycles counts in accordance with the Instructions per cycle;

(5) Convert CPU cycles counts to CPU active time with regard to CPU frequency

(6) Calculate energy consumption of CPU with respect to time cost on each power state

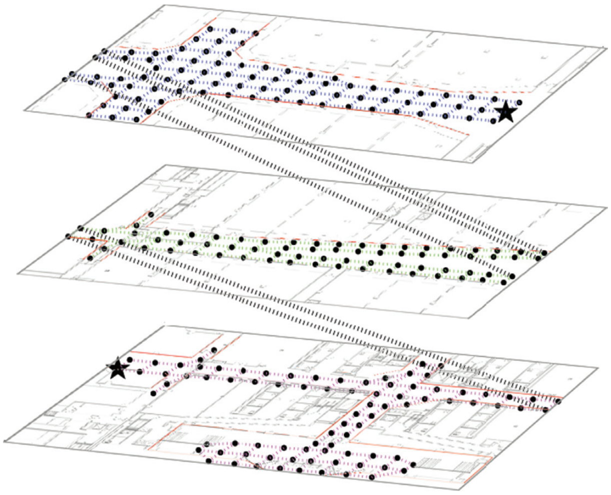

Figure 3: The graph representative of a Canary Wharf Shopping Mall. The black stars represent the egresses.

6. Experiments and Results

To evaluate the effectiveness of our proposed algorithm, we simulate several hazard environments as it is insufficient to test the resilience and robustness of the algorithm in normal conditions without introducing high dynamics of users and fast changes in network topology. Experiments are conducted on two featured building models with diverse population densities (30, 60, 90, and 120 civilians, respectively) when fire-related disasters breaking out. The bio-inspired mechanisms are combined with the CPN-based algorithm with distance metric. Dijkstra's shortest path algorithm (DSP) and the original CPN based algorithm with distance metric (CPNSP) are performed for comparison purpose.

Figure 4 summarises the percentage of simulated civilians that evacuate from the Imperial College's EEE building. It shows clearly that CPNSP with bioinspired mechanisms reaches the performance of DSP or CPNSP in lower occupancy rates (30 or 60 evacuees) although it actually sends fewer packets. While in higher population densities, both CPN-based algorithms surpass the performance of DSP. This is because DSP is a global optimal algorithm that always directs all the evacuees to the ideal path and therefore tends to cause congestion. On the other hand, as a heuristic algorithm, CPN sometimes provides near-optimal routes and naturally avoids jamming. Moreover, CPNSP with bio-inspired mechanisms outperforms the original CPNSP as it is more sensitive to changes in the environment. This is because (1) SPs with novelty-seeking behaviour can discover routes to alternative egresses in advance and redirect evacuees immediately when the desired exit is blocked by fire; (2) CPNSP with bioinspired mechanisms sends source-routed SPs to remeasure the QoS of paths in the mailbox and is more sensitive to the fire spreading while the top-ranked path in the original CPNSP may not be remeasured before being deleted due to timeout;

Figure 5 presents the average percentage of survivors of five experiments over a Canary Wharf shopping mall. Unlike the first building structure in which all the exits are on the ground floor, one of the exits of the shopping mall is located on the second floor. We assume a very fast-spreading fire disaster breaks out near the exit of the ground floor. The results indicate that CPNSP does not perform well in this scenario because it is difficult for normal SPs emitted from the ground floor to detect the distant exit on the second floor before the fire blocks the exit on the ground floor. In addition, constantly sending a large number of SPs when the network is stable may cause "overtraining" problem (the neuron which associates with the optimal path may be frequently reinforced and other neurons may be continuously punished; hence, it may take a short period for SPs to adapt to a new searching direction when the previous one suddenly becomes worse or gets disconnected). As a result, fast-spreading fire will reach evacuees before CPN adjusts suggested directions to the exit on the second floor and therefore cause a large number of injuries and causalities. On the other hand, DSP and CPNSP with bioinspired mechanisms achieve 100% survival rate because both of them can direct evacuees on the ground floor to the egress on the second floor promptly when their desired egress is blocked by the hazard. DSP can discover the optimal path quickly because it collects hazard information from each node in the network periodically and compute the global optimal path. CPNSP with bioinspired mechanisms can switch evacuation decisions immediately because it continuously monitors alternative exits by using novelty-seeking SPs as well as remeasures the current top-ranked path with source-routed SPs.

Figures 6 and 7 show the energy consumption and the total CPU cycles of 2 minutes long evacuation process when using the original CPNSP and CPNSP with bioinspired mechanisms, respectively. The results indicate that, by using the bioinspired mechanisms, CPN can reduce the energy utilisation and CPU cycles considerably. This is mainly because CPNSP with bioinspired mechanisms can adjust the number of SPs sent with respect to the interested events observed rather than using a predefined packet rate. When a CPN node detects spreading of a hazard, it will send more SPs with a larger drift parameter to measure a broader area

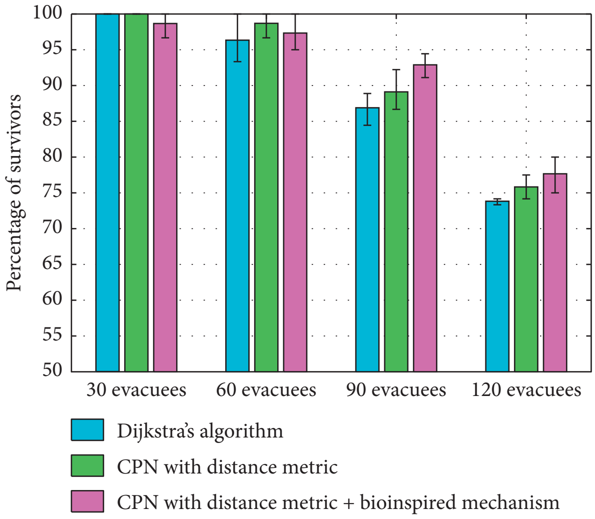

FIGURE 4: The percentage of survivors in the Imperial College's EEE building model for each scenario. The results are the average of five randomized simulation runs, and error bars show the min/max result in any of the five simulation runs.

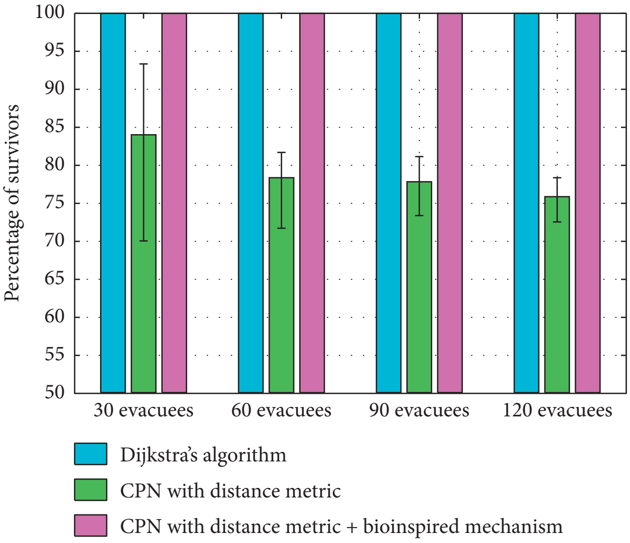

FIGURE 5: The percentage of survivors in the Canary Wharf building model for each scenario. The results are the average of five randomized simulation runs, and error bars show the min./max, result in any of the five simulation runs.

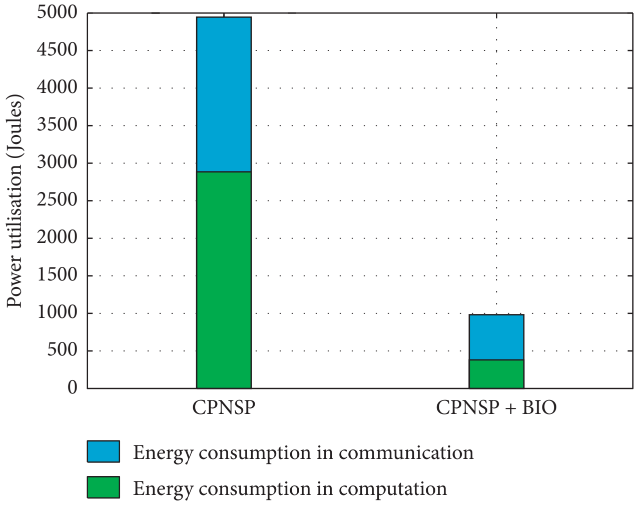

Figure 6: The total energy consumption of each algorithm for a two minute long evacuation process. The blue bar represents the overall energy consumption in communication process while the green bar shows the total energy consumption of microprocessors.

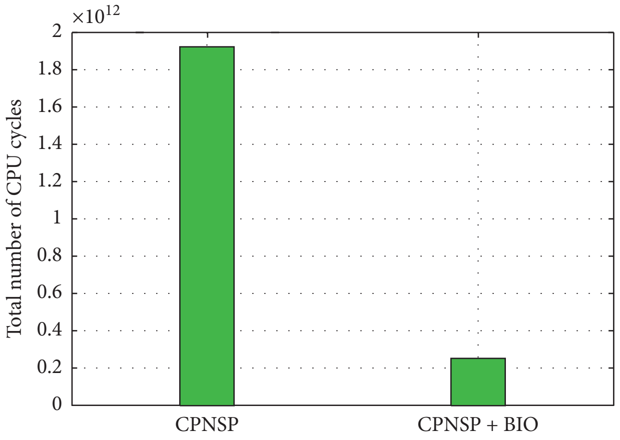

Figure 7: Total CPU cycles used in each algorithm.

for the sake of discovering the optimal route; otherwise, it will mainly send source-routed packets to measure the obtained optimal paths in the mailbox.

7. Conclusion

In this paper, we propose a bioinspired emergency navigation approach that can reconfigure key parameters based on the observed information brought back by SPs. Two novel packet searching motions inspired by social insect behaviours are also presented to improve the efficiency in path-finding and energy-saving. The experimental results indicate that the algorithm can achieve self-adaptive in a highly dynamic emergency environment and effectively increase survival rates.

Data Availability

Data and the software used in this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Key R&D Program for the 13th-Five-Year Plan of China (2018YFF0300305 in 2018YFF0300300). The first author would like to thank his PhD supervisor, Professor Erol Gelenbe, for his invaluable guidance and continuous help during the first author’s PhD program in Imperial College London.

References

[1] Z. He, W. Zhang, and N. Jia, “Estimating carbon dioxide emissions of freeway traffic: a spatiotemporal cell-based model,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 5, pp. 1976–1986, 2019.

[2] W. L. Shang, Y. Chen, and W. Y. Ochieng, “Resilience analysis of transport networks by combining variable message signs

with agent-based day-to-day dynamic learning," *IEEE Access*, vol. 8, pp. 104458–104468, 2020.

[3] Z. Yang, K. Li, Q. Niu, and Y. Xue, "A comprehensive study of economic unit commitment of power systems integrating various renewable generations and plug-in electric vehicles," *Energy Conversion and Management*, vol. 132, pp. 460–481, 2017.

[4] S. Gan, S. Liang, K. Li, J. Deng, and T. Cheng, "Long-term ship speed prediction for intelligent traffic signaling," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 1, pp. 82–91, 2016.

[5] S. Gan, S. Liang, K. Li, J. Deng, and T. Cheng, "Trajectory length prediction for intelligent traffic signaling: a data-driven approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 2, pp. 426–435, 2017.

[6] Y. Wang, K. Li, S. Gan, and C. Cameron, "Analysis of energy saving potentials in intelligent manufacturing: a case study of bakery plants," *Energy*, vol. 172, pp. 477–486, 2019.

[7] Y. Wang, K. Li, S. Gan, and C. Cameron, "Missing data imputation with ols-based autoencoder for intelligent manufacturing," *IEEE Transactions on Industry Applications*, vol. 55, no. 6, pp. 7219–7229, 2019.

[8] H. Bi and O. H. Abdelrahman, "Energy-aware navigation in large-scale evacuation using g-networks," *Probability in the Engineering and Informational Sciences*, vol. 32, no. 3, pp. 340–352, 2018.

[9] H. Bi and E. Gelenbe, "A survey of algorithms and systems for evacuating people in confined spaces," *Electronics*, vol. 8, no. 6, p. 711, 2019.

[10] K. Wang, Y. Chen, M.-S. Alouini, and F. Xu, "Ber and optimal power allocation for amplify-and-forward relaying using pilot-aided maximum likelihood estimation," *IEEE Transactions on Communications*, vol. 62, no. 10, pp. 3462–3475, 2014.

[11] K. Wang, K. Yang, and C. S. Magurawalage, "Joint energy minimization and resource allocation in c-ran with mobile cloud," *IEEE Transactions on Cloud Computing*, vol. 6, no. 3, pp. 760–770, 2016.

[12] T. W. Malone, K. R. Grant, F. A. Turbak, S. A. Brobst, and M. D. Cohen, "Intelligent information-sharing systems," *Communications of the ACM*, vol. 30, no. 5, pp. 390–402, 1987.

[13] E. Gelenbe, J. Domanska, P. Fröhlich, M. P. Nowak, and S. Nowak, "Self-aware networks that optimize security, qos, and energy," *Proceedings of the IEEE*, vol. 108, no. 7, pp. 1150–1167, 2020.

[14] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," *Ad Hoc Networks*, vol. 3, no. 3, pp. 325–349, 2005.

[15] S. Tilak, N. B. Abu-Ghazaleh, and W. Heinzelman, "A taxonomy of wireless micro-sensor network models," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 6, no. 2, pp. 28–36, 2002.

[16] J.-H. Chang and L. Tassiulas, "Maximum lifetime routing in wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 12, no. 4, pp. 609–619, 2004.

[17] R. Madan and S. Lall, "Distributed algorithms for maximum lifetime routing in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 5, no. 8, pp. 2185–2193, 2006.

[18] M. H. Yaghmaee and D. A. Adjeroh, "Priority-based rate control for service differentiation and congestion control in wireless multimedia sensor networks," *Computer Networks*, vol. 53, no. 11, pp. 1798–1811, 2009.

[19] S. K. Swain and P. K. Nanda, "Priority based adaptive rate control in wireless sensor networks: a difference of differential approach," *IEEE Access*, vol. 7, pp. 112435–112447, 2019.

[20] E. Felemban, C. G. Chang-Gun Lee, and E. Ekici, "Mmspeed: multipath multi-speed protocol for qos guarantee of reliability and. timeliness in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 6, pp. 738–754, 2006.

[21] S. Misra, S. Das, and M. Obaidat, "Context-aware quality of service in wireless sensor networks," *IEEE Communications Magazine*, vol. 52, no. 6, pp. 16–23, 2014.

[22] E. Gelenbe, "A diffusion model for packet travel time in a random multihop medium," *ACM Transactions on Sensor Networks*, vol. 3, no. 2, p. 10, 2007.

[23] E. Gelenbe, "On approximate computer system models," *Journal of the ACM*, vol. 22, no. 2, pp. 261–269, 1975.

[24] H. Elhammouti, E. Sabir, M. Benjillali, L. Echabbi, and H. Tembine, "Self-organized connected objects: rethinking qos provisioning for iot services," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 41–47, 2017.

[25] W. Osamy, A. M. Khedr, A. Salim, and Adsda, "Adaptive distributed service discovery algorithm for internet of things based mobile wireless sensor networks," *IEEE Sensors Journal*, vol. 19, no. 22, pp. 10,869–10,880, 2019.

[26] D. Duan, L. Yang, Y. Cao, J. Wei, and X. Cheng, "Self-organizing networks: from bio-inspired to social-driven," *IEEE Intelligent Systems*, vol. 29, no. 2, pp. 86–90, 2014.

[27] X. Deng, P. Jiang, X. Peng, and C. Mi, "An intelligent outlier detection method with one class support tucker machine and genetic algorithm toward big sensor data in internet of things," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 6, pp. 4672–4683, 2018.

[28] M. Nadji-Tehrani and A. Eslami, "A brain-inspired framework for evolutionary artificial general intelligence," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[29] A. Dore, A. F. Cattoni, and C. S. Regazzoni, "Interaction modeling and prediction in smart spaces: a bio-inspired approach based on autobiographical memory," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 40, no. 6, pp. 1191–1205, 2010.

[30] J. Park, H. Byun, and J.-R. Lee, "Bio-inspired load-balancing framework for loosely coupled heterogeneous server systems," *IEEE Transactions on Computers*, vol. 65, no. 11, pp. 3280–3292, 2016.

[31] S. Bitam, A. Mellouk, and S. Zeadally, "Bio-inspired routing algorithms survey for vehicular ad hoc networks," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 2, pp. 843–867, 2014.

[32] J. Li, H. Zhao, A. S. Hafid, J. Wei, H. Yin, and B. Ren, "A bio-inspired solution to cluster-based distributed spectrum allocation in high-density cognitive internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9294–9307, 2019.

[33] A. Jankowska, M. Schut, and N. Ferreira-Schut, "A wireless actuator-sensor neural network for evacuation routing," in *Proceedings of the Third International Conference on Sensor Technologies and Applications, SENSORCOMM'09*, pp. 139–144, IEEE, Athens, Greece, June 2009.

[34] Q. Li, Z. Fang, Q. Li, and X. Zong, "Multiobjective evacuation route assignment model based on genetic algorithm," in *Proceedings of the 2010 18th International Conference on Geoinformatics*, pp. 1–5, IEEE, Beijing, China, June 2010.

[35] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii," *Lecture Notes in Computer Science 1917*, pp. 849–858, Springer, Berlin, Germany, 2000.

[36] D. Eppstein, “Finding the k shortest paths,” *SIAM Journal on Computing*, vol. 28, no. 2, pp. 652–673, 1998.

[37] X. Pan, C. S. Han, and K. H. Law, “A multi-agent based simulation framework for the study of human and social behavior in egress analysis,” in *Proceedings of the ASCE International Conference on Computing in Civil Engineering*, vol. 92, Cancun, Mexico, July 2005.

[38] F. Samadzadegan and M. Yadegari, “A biologically-inspired optimization algorithm for urban evacuation planning in disaster management,” in *Proceedings of ACRS*, Hanoi, Vietnam, November 2010.

[39] E. Gelenbe, “Steps toward self-aware networks,” *Communications of the ACM*, vol. 52, no. 7, pp. 66–75, 2009.

[40] E. Gelenbe, R. Lent, and A. Nunez, “Self-aware networks and qos,” *Proceedings of the IEEE*, vol. 92, no. 9, pp. 1478–1489, 2004.

[41] E. Gelenbe, R. Lent, and Z. Xu, “Towards networks with cognitive packets,” in *Performance and QoS of Next Generation Networking*, pp. 3–17, Springer, Berlin, Germany, 2001.

[42] E. Gelenbe, “Random neural networks with negative and positive signals and product form solution,” *Neural Computation*, vol. 1, no. 4, pp. 502–510, 1989.

[43] E. Gelenbe, “Stability of the random neural network model,” *Neural Computation*, vol. 2, no. 2, pp. 239–247, 1990.

[44] E. Gelenbe, “Learning in the recurrent random neural network,” *Neural Computation*, vol. 5, no. 1, pp. 154–164, 1993.

[45] E. Gelenbe, “Natural computation,” *The Computer Journal*, vol. 55, no. 7, pp. 848–851, 2012.

[46] E. Gelenbe, E. Seref, and Z. Xu, “Simulation with learning agents,” *Proceedings of the IEEE*, vol. 89, no. 2, pp. 148–157, 2001.

[47] E. Gelenbe and S. Timotheou, “Random neural networks with synchronized interactions,” *Neural Computation*, vol. 20, no. 9, pp. 2308–2324, 2008.

[48] E. Gelenbe, N. Schmajuk, J. Staddon, and J. Reif, “Autonomous search by robots and animals: a survey,” *Robotics and Autonomous Systems*, vol. 22, no. 1, pp. 23–34, 1997.

[49] Z. S. Liang, T. Nguyen, H. R. Mattila, S. L. Rodriguez-Zas, T. D. Seeley, and G. E. Robinson, “Molecular determinants of scouting behavior in honey bees,” *Science*, vol. 335, no. 6073, pp. 1225–1228, 2012.

[50] A. Desmet, “A modelling approach to human navigation in constrained spaces,” Ph.D. thesis, Imperial College London, London, UK, 2014.

[51] B. F. McMahon and R. M. Evans, “Foraging strategies of American white pelicans,” *Behaviour*, vol. 120, no. 1–2, pp. 69–89, 1992.

[52] M. Noda, K. Gushima, and S. Kakuda, “Local prey search based on spatial memory and expectation in the planktivorous reef fish, Chromis chrysurus (Pomacentridae),” *Animal Behaviour*, vol. 47, no. 6, pp. 1413–1422, 1994.

[53] R. H. MacArthur and E. R. Pianka, “On optimal use of a patchy environment,” *The American Naturalist*, vol. 100, no. 916, pp. 603–609, 1966.

[54] N. Dimakis, A. Filippoupolitis, and E. Gelenbe, “Distributed building evacuation simulator for smart emergency management,” *The Computer Journal*, vol. 53, no. 9, pp. 1384–1400, 2010.

[55] G. Gorbil, A. Filippoupolitis, and E. Gelenbe, “Intelligent navigation systems for building evacuation,” in *Computer and Information Sciences II*, pp. 339–345, Springer, Berlin, Germany, 2011.

[56] V. Shnayder, M. Hempstead, B. . r. Chen, G. W. Allen, and M. Welsh, “Simulating the power consumption of large-scale sensor network applications,” in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, pp. 188–200, ACM, Baltimore MD USA, November 2004.

[57] H. Bi, W. Shang, and Y. Chen, “Cooperative and energy-efficient strategies in emergency navigation using edge computing,” *IEEE Access*, vol. 8, no. 54, pp. 441–455, 2020.