bioSmartSense: A Bio-inspired Data Collection Framework for Energy-efficient, QoI-aware Smart City Applications

Satyaki Roy*
Dept. of Computer Science
Missouri S&T
Rolla, MO 65409, USA
sr3k2@mst.edu

Nirnay Ghosh*
iTrust Centre for Research in Cyber Security
Singapore University of Technology and Design
Singapore 487372
nirnay_ghosh@sutd.edu.sg

Sajal K. Das
Dept. of Computer Science
Missouri S&T
Rolla, MO 65409, USA
sdas@mst.edu

***Abstract*—Recent years have seen a proliferation of intelligent (automated) decision support systems for various smart city applications such as energy management, transportation, healthcare, environment monitoring, and so on. A key enabler in the smart city paradigm is the *Internet-of-Things (IoT)* network of smart sensing and actuation devices assisting in real-time detection and monitoring of physical phenomena. The underlying IoT network must be energy-efficient for application sustainability and also quality of information (QoI)-aware for near-perfect device actuation. To this end, this paper proposes *bioSmartSense*, a novel bio-inspired distributed event sensing and data collection framework, based on the *gene regulatory networks (GRNs)* in living organisms. The idea is to make the sensing and reporting tasks energy-efficient through self-modulation of IoT device energy levels, analogous to the activation or repression of genes by the regulating proteins, called *Transcription Factors (TFs)*. To support energy-efficient and QoI-aware information dissemination, we first customize a heuristic designed for the *Maximum Weighted Independent Set* problem encompassing both 'quality' and 'quantity' of sensed data, where the former depends on the device energy levels while the latter on the number of events sensed. We utilize the heuristic to propose a sub-optimal device selection mechanism constrained on the IoT network's overall residual energy. Simulation experiments demonstrate that the *bioSmartSense* framework achieves better energy-efficiency while maximizing event reporting compared to a state-of-the-art data collection approach for smart city applications.**

***Index Terms*—Gene Regulatory Networks, Energy-efficiency, Smart city applications, Quality of Information**

I. INTRODUCTION

Urban population has undergone unprecedented growth over the last two decades. A recent survey[1] by the United Nations shows that 54% of the world's population live in urban areas, which is expected to increase to 66% by 2050. While only 2% of the world's surface is occupied by urban environments, the cities contribute to 80% of global gas emission, 75% of global energy consumption and 60% of residential water usage [1]. Thus, it is imperative to design efficient and smart solutions for various public services in urban areas, such as energy management, transportation, healthcare, environment monitoring, and so on. The objective is to provide support for intelligent decision making to improve quality of life and make our environment and hence the society more sustainable.

Among other things, smart cities rely on the information and communication technology (ICT) solutions to deliver services to the citizens [2]. *Internet-of-Things (IoT)* is envisioned as a candidate building block to develop sustainable ICT platforms [1]. An IoT network comprises smart devices with sensing, communication, computing, actuation, and storage capabilities that can communicate with other devices and users via smartphones and wearables, as well as through application platforms to enable ubiquitous computing environments. Unlike a traditional wireless sensor node, an IoT device is typically more powerful in terms of its capabilities. Also, it does not act as intermediate routing node to deliver messages to the application platform, but leverages the wireless communication and backbone network for data transfer.

Energy-efficient smart city solutions require the IoT devices to operate without compromising the quality of sensing, processing, and transferring the collected data [3], implying that the lifetime of IoT devices are critical. Often the devices may be deployed at locations remote to the application platform and their limited energy gets dissipated while sensing the environment and communicating the sampled data via wireless communication technologies, such as 3G/4G/LTE, WiFi, ZigBee, or Bluetooth/BLE [4]. Furthermore, it may not always be feasible to replenish their batteries nor replace them with fully-charged devices on-the-fly.

A possible way to achieve energy-efficiency is to use the available (or residual) energy judiciously, i.e., either refrain from unnecessary sensing and reporting operations, or schedule the devices such that the tasks are conducted intermittently (both periodic and event-driven sensing). For both methods, any drop in the frequency or intensity of collecting data samples results in inaccurate sensing which, in turn, directly affects the *quality of information* (QoI) [5], [6]. Thus, the actuation which the application platform triggers based on the information received infrequently or intermittently, will be

* Co-primary authors
[1]http://www.un.org/en/development/desa/news/population/world-urbanization-prospects-2014.html

erroneous and unproductive to the end users. Therefore, it is critical to design efficient data collection frameworks for IoT-based smart city applications [7] that incorporate both sensing and transferring/reporting of data samples. This serves as the motivation for our work.

A challenge in designing such frameworks is to minimize the cost of energy dissipation for sensing and reporting, by estimating the residual energy of the devices, yet simultaneously guaranteeing satisfactory quality of the contributed information to ensure the application's operational reliability. Realizing that IoT-based smart city applications will benefit from *energy-efficient* and *QoI-aware* data collection frameworks for sustainable delivery of public services, this paper proposes a novel bio-inspired data collection framework, called *bioSmartSense*, by exploiting the concept of *gene regulatory networks (GRN)* in living organisms. A GRN is a network of interactions between the DNA segments, called *genes*, regulating the concentration of protein synthesis in living cells. GRNs of unicellular organisms like *E. coli* and *Yeast* have been widely studied due to their biological robustness, which is a consequence of the underlying topological or graph properties (for details, refer to Section III).

The literature on robust, energy-efficient topology design includes works that attempt to utilize GRNs in wireless sensor networks (WSNs). Nazi *et al.* [8], [9] leveraged GRN topology to design efficient WSN topology by establishing a one-to-one mapping between the nodes in these two graphs. Roy *et al.* [10] introduced an edge rewiring mechanism to enhance GRN topology against random node failures. The bio-inspired WSNs exhibit high packet delivery rate and low network latency compared to Erdos-Renyi (ER) random topology and k-connected network topology. Markham *et al.* [11] proposed a GRN-based approach to configure WSNs in a target tracking application. Byun *et al.* [12] proposed a GRN-based self-organizing control of WSNs in order to guarantee energy-efficiency and communication latency. However, no existing work has yet exploited the regulatory information of GRNs to study energy-efficiency of resource-constrained networks.

Our choice of a biological network like GRN as the underlying principle of designing energy-efficient data collection framework in an IoT network of sensing devices is motivated by their functional similarity. Sensing devices can operate in different power (energy) states, such as *idle, sleep, active,* and *off* which determine its rate of power consumption [13]. Indeed, an IoT device adjusts to a power level following an event trigger, and starts to operate at a given degree of intensity specific to that power level. Analogously, the genes in a GRN function at different expression levels to synthesize proteins of variable quantities and concentration in order to meet dynamic cellular requirements and support sustainability of the living organism [14] [15]. The expression levels of a specific gene are controlled by itself as well as by other regulating (neighboring) genes. We intuit that if a sensing device in IoT network is capable of regulating its energy levels as well as that of its neighbors, then smart city applications will be more sustainable. The objective of proposed data collection framework, *bioSmartSense*, is to validate this very intuition.

To support both energy-efficiency and QoI requirements of our framework, we need to design a schedule that selects an optimal subset of devices to carry out sensing tasks and reporting at different times. To this end, we formulate an optimization problem, prove its *NP-completeness*, and derive a sub-optimal solution by customizing a heuristic for the *Maximum Weighted Independent Set (MWIS)* problem. To the best of our knowledge, this is the first work leveraging a biological network to design energy-efficient and QoI-aware data collection framework for smart city applications. Below we highlight the major contributions of the work.

1) A novel GRN-based data collection framework, *bioSmartSense*, is presented to facilitate energy-efficient and QoI-aware data sensing and reporting for sustainable IoT-based smart city applications.
2) Energy regulation of IoT devices is modeled exploiting self and neighbor regulation mechanisms in GRNs.
3) An optimization problem is formulated to select a subset of IoT devices for sensing and reporting tasks, by satisfying energy-efficiency and QoI requirements.
4) After proving NP-completeness of the optimization problem, a sub-optimal algorithm is proposed by customizing a heuristic for the *Maximum Weighted Independent Set (MWIS)* problem.
5) Performance evaluation through extensive simulation studies demonstrates the efficacy of the *bioSmartSense* framework and its superiority, in terms of energy-efficiency and event reporting capability, as compared with those of a state-of-the-art data collection approach for smart city applications.

The rest of the paper is organized as follows. Section II discusses the system model while Section III introduces GRNs and different types of regulations. Section IV describes in details the *bioSmartSense* framework. Section V presents the device selection mechanism and Section VI discusses the simulation results and compares the performance of proposed data collection framework. Section VII draws conclusions and identifies future work.

II. System Model

Fig. 1 depicts the system model that captures one of multiple sensing regions formed by partitioning an urban area. To realize the smart city setting, let us consider a set of IoT devices, $D = \{d_1, d_2, d_3, \cdots\}$ deployed over a region to sense the environment (e.g., vehicular traffic volume) and reporting the data to a remote application platform, termed as the *base station*. At deployment time, all devices are assumed to have the same initial energy, ε.

Let us now introduce the concept of *time epoch*, t, to delineate different processes taking place in each cycle of the application. It is a configurable parameter, whose duration depends on the frequency of data sample collected by the base station. Specifically, the time epoch is defined as a fixed temporal window $t = t_r + t_s + t_b + t_{tx}$, divided into the following four phases: (i) *Device energy regulation* (t_r): IoT

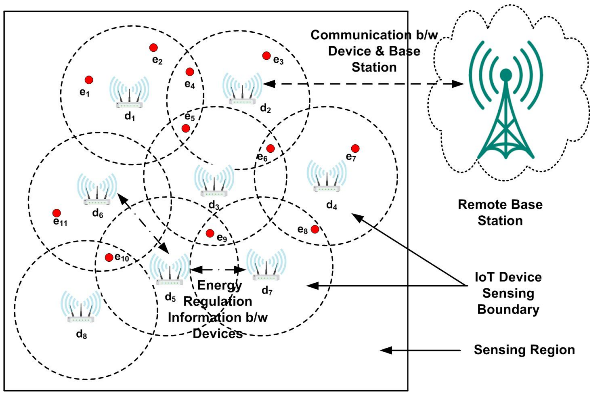


Fig. 1: System Model

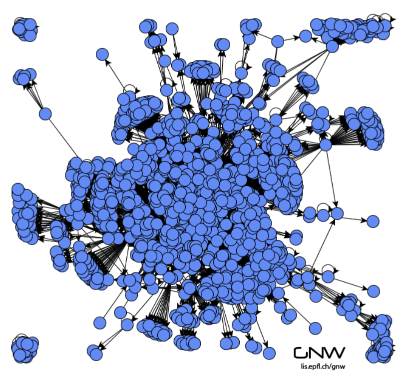
devices exchange regulation information among themselves and modulate energy levels to perform sensing and reporting; (ii) *Event sensing (t_s)*: Devices sense events occurred in the vicinity; (iii) *Device-base station beacon exchange (t_b)*: Devices send beacon messages to the base station informing the events they sensed along with their individual energy levels. The base station chooses a subset of these devices and informs them to initiate reporting of the sensed information; and (iv) *Information transmission (t_{tx})*: Each selected device processes and aggregates the samples collected during the sensing phase and transfers it as event information (reporting).

Three important components of this model are as follows:
1. IoT device: An IoT device $d_i \in D$ is a rechargeable battery-powered node with sensing, processing, storing, and reporting capabilities. It is placed in the sensing region to collect information about its vicinity (e.g., data related to traffic density). Each device d_i senses events in its sensing boundary defined by a fixed radius, r_d. The information is conditionally transmitted to the base station using WiFi, Bluetooth low energy (BLE) or ZigBee communication protocols depending on the device specifications. For sake of simplicity, we assume that the sensing region consists of only WiFi devices. All devices are equipped with GRN-based energy-level modulation, which enables them to control the intensity of operations, viz., sensing, reporting, and data processing. At any point, the device energy level belongs to the interval [0, 1]. Our system model supports two types of regulations:
a. Neighbor regulation: In this regulation, each sensor device sends regulatory messages to its successors and receives the same from its predecessors during every device energy regulation phase of the time epoch (details in Section IV-B).
b. Self regulation: Each device has a predefined lower and upper limits on its energy level. If the mean energy level drops below the lower limit, or exceeds the upper one, then it resets to 0.5. This action also takes place in the device energy regulation phase.
2. Event: Let a set of $E = \{e_1, e_2, e_3, \cdots\}$ events occur within the sensing region at a particular time epoch. Any event $e_k \in E$ (denoted by red dot in Fig. 1) is an alert which gets triggered at a location within the sensing region, if one or more device(s) sense(s) an anomalous deviation from a predefined normal condition (w.r.t. the traffic volume in our application). Each event is associated with a geographical location information and a quantitative metric, objectively measured by the IoT device. If multiple event alerts are generated at approximately the same location and during the same time epoch, we consider them to be the same event.
3. Base Station: A base station is the application platform which is remotely located from the sensing regions (refer Fig. 1). Its communication range (r_B) is much higher compared to that of IoT devices, and it uses a plethora of communication protocols and the backbone Wireless LAN network for bidirectional message exchange with the devices. The base station is responsible for making decisions regarding the selection of IoT devices to perform sensing and reporting tasks at different time epochs, such that the energy-efficiency and QoI requirements of the application are satisfied.

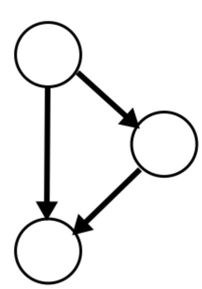
As explained above, Fig. 1 shows that a device has a fixed *sensing boundary* within which multiple events of interest may take place during a particular time epoch. The device can report all or a subset of events depending on its available residual energy. Following [16], we also assume that at a higher energy level, a device can generate data at a higher sampling rate, thus resulting in more accurate measurements. However, acceptance of a device's contribution by the base station at the current time epoch will depend on the number of events sensed ('quantity' of data) and the expected *QoI index* (refer to Section V) of data samples collected for the sensed events ('quality' of data).

III. Gene Regulatory Networks

The GRN topology represents interactions between genes and proteins, called *Transcription Factors (TFs)* or messenger RNAs (mRNAs). Let $\mathcal{G}_g(\mathcal{V}_g, \mathcal{E}_g)$ denote a GRN graph, where the vertex-set $\mathcal{V}_g$ represents the genes, proteins, mRNAs; and the edge-set $\mathcal{E}_g$ represents the interactions corresponding to molecular reactions or processes. In practice, $\mathcal{G}_g(\mathcal{V}_g, \mathcal{E}_g)$ is a directed, weighted graph. Fig. 2a shows the topology of *E. coli* GRN obtained from a software tool, called *GeneNetWeaver* [17].



(a) Topology of *E. coli*



(b) Feed Forward Loop Motif

Fig. 2: GRN: Topology and FFL Motif

The GRN topology has a very low graph density of 0.0015 on a scale of 0 to 1, where 0 corresponds to an empty graph and 1 corresponds to a complete graph. Despite its sparseness, the clustering coefficient[2] of GRN is over 80 times that of *Erdos-Renyi (ER) random graphs* [18] [3] of the same graph density, by virtue of the presence of acyclic triangular subgraphs, called *Feed Forward Loop (FFLs)* motifs [19] (Fig. 2b). Experiments show that the abundance of FFL motif makes the GRN topologically robust as compared to ER random graphs of same graph density [19]. We characterize *topological robustness* by the ability of a network to stay connected despite node failures [19]. The interactions between the GRN nodes (i.e. the edge weights) can be categorized into three classes:

Activation: Increase in the protein concentration of the regulating entity can lead to an increase of the same in the target entity. It is represented by $+$ sign.

Repression: Decrease in the concentration of the regulating entity can lead to the decreased concentration of the target entity. It is represented by $-$ sign.

Dual: Regulator can activate or inhibit the target node. It is represented by $+-$ sign. Additionally, GRN nodes are also capable of self-regulation. Due to this property, a single gene may activate or repress its own activity. A self-regulating gene is a very common phenomenon in *E. coli* GRN, where 40% of TFs negatively regulate themselves [20].

IV. Proposed bioSmartSense Framework

In this section, we present the different functional components of our proposed framework, *bioSmartSense*.

A. Mapping IoTNet to GRN Topology

As hinted in Section III, the inherent robustness, fault-tolerance and self-regulation in GRN topology makes it an ideal choice for designing energy-efficient data collection framework. First, we map the network of IoT devices, abbreviated *IoTNet*, into a corresponding GRN topology. We adopt the mapping algorithm proposed by one of our coauthors in *et al.* [8] [9], which takes the topology of *IoTNet* as input and generates a corresponding subgraph that preserve the topological robustness of GRNs. The mapping algorithm in our context is described as follows.

Let the *IoTNet* be represented as $\mathcal{G}_w(\mathcal{V}_w, \mathcal{E}_w)$, where $\mathcal{V}_w = d_1, d_2, ..., d_m$ represents the set of IoT devices and an edge $\{d_i, d_j\} \in \mathcal{E}_w$ exists if two devices d_i and d_j are within the (wireless) transmission range of one another. Thus, $\mathcal{V}_w$ is equivalent to the device-set D defined in Section II. We utilize the *page rank-based mapping algorithm* [9] to form a GRN overlay network over $\mathcal{G}_w$, by establishing a correspondence between the topology of GRNs and $\mathcal{G}_w$. It introduces the mapping function $M : \mathcal{G}_{mw} \rightarrow \mathcal{G}_g$ such that $\mathcal{G}_{mw}(\mathcal{V}_{mw}, \mathcal{E}_{mw})$ is a directed mapped IoTNet topology, where $\mathcal{V}_{mw} \subset \mathcal{V}_w$ and $\mathcal{E}_{mw} \subset \mathcal{E}_w$. An edge $\{d_i, d_j\} \in \mathcal{E}_{mw}$ exists if and only if there exists a path between $M(d_i)$ and $M(d_j)$ in $\mathcal{G}_g$. The time complexity of the algorithm is $O(m^2)$, where m is the number of sensing devices in *IoTNet*.

[2]Clustering coefficient of a graph is a measure of the degree to which nodes in a graph tend to cluster together.

[3]The Erdös-Rényi (ER) random graph is constructed by initializing a set of isolated nodes and introducing a directed edge between each distinct pair of nodes with probability p.

An Illustrative Example: As shown in Figs. 3a and 3b, the list of genes and IoT device labels are annotated by their respective normalized rank scores. The mapping algorithm processes the list of highest ranked genes and IoT devices, and maps device 1 into gene c. Since 0 is the next highest ranked device having an edge with mapped device 1 is 0, it gets mapped to the next highest ranked gene g (neighbor of gene c). Similarly, device 2 is mapped to gene b. The next highest ranking device 4 shares edges with mapped devices 0 and 1. It is mapped to gene e which has paths to corresponding mapped genes g and c. Note that the communication links inherit the direction of data forwarding from the direction of interactions between corresponding genes in the input GRN topology, resulting in a directed graph *mapped-IoTNet*. Finally, the last IoT device 3 is mapped to gene a, because the latter interacts directly with b. The *mapped-IoTNet* graph is shown in Fig. 3c.

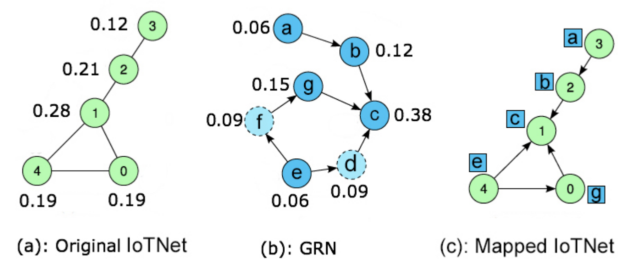


Fig. 3: Working of the Mapping Algorithm

In the next section, we describe the different regulation techniques that have been used to modulate the energy levels of the devices in *mapped-IoTNet*.

B. GRN-based Energy Regulation

For a directed edge $\{d_i, d_j\} \in \mathcal{E}_{mw}$, d_i is the *predecessor* of d_j while d_j is the *successor* of d_i. The list of predecessors and successors of a node $d_i \in \mathcal{V}_{mw}$ is denoted as $\phi(d_i)$ and $\eta(d_i)$, respectively. We consider two kinds of regulations:

Neighbor regulation: Each IoT device d_i sends regulatory messages to devices in $\eta(d_i)$ and receives regulatory messages $\langle senderID, sender\_energy\_level, regulation\_type, current\_time \rangle$ from devices in $\phi(d_i)$ during every regulation phase of the time epoch, t. For each device d_i with energy level $l^t_{d_i}$ at time instance t, its energy level at time $t+1$ is:

$$l^{t+1}_{d_i} = \sum_{v \in \phi(d_i)} \kappa \times \mathbf{W}(d_j, d_i) \times l^t_{d_i} \tag{1}$$

where $\mathbf{W}$ is a function that defines the nature of interactions among communicating devices, i.e., $\mathbf{W} : \mathbf{W}(d_j, d_i) \rightarrow \{+, -\}, \forall \{d_j, d_i\} \in \mathcal{G}_{mw}$, and κ is the rate constant which determines the degree of positive or negative regulation of a node by its predecessors.

Self regulation: This regulation technique is used by a device d_i to regulate its own energy level. Given two residual energy limits L and U $(0 \leq L < U \leq 1.0)$, if the mean energy level of the device d_i, $\frac{1}{rI} \sum_t l^t_{d_i}$, drops below L or exceeds U

during the regulation phase of duration rI, $l^t_{d_i}$ is re-initialized to energy level any baseline. In our experiments, we consider a baseline of 0.5.

C. Sub-optimal Selection of Sensing Devices

For real-time sensing-based applications, frequent replenishment of the battery of the sensing devices or replacement of inactive devices on-the-fly is often infeasible. Evidently, there is a significant constraint on the IoT network energy, and at any point of time it is roughly capped by the sum of residual energies of individual devices. Thus, efficient utilization of energy is paramount to maintain a fully operational network for longer periods of time. In general, the duration for which a system maintains the desired level of network performance and QoS, is denoted by *network lifetime*.

Likewise in our IoT-based smart city application, we gauge the network lifetime as the time until which it is able to disseminate event related data (occurring within a sensing region) with significant quantity (count) and quality (accuracy). Longer network lifetime will enable the base station (application platform) to support uninterrupted services without recharging the devices for an extended period of time. Thus, a smart city application ideally needs to maximize both quality and quantity of event information at every time epoch, given that the base station is constrained by overall residual energy of the *IoTNet*.

For a set of events $E = \{e_1, e_2, \cdots, e_n\}$, a set of devices $D = \{d_1, d_2, \cdots, d_m\}$ and an overall residual energy budget $\mathcal{E}^t = \sum_{i=1}^{m} \varepsilon^t_{d_i}$ at time epoch t, where $\varepsilon^t_{d_i}$ is the residual energy for device d_i, the task of *bioSmartSense* is to select an *optimal* set of devices whose reports (data samples) are to be accepted by the base station and used in the application for providing services. The selection mechanism will avoid receiving reports from all devices that have generated reports for the same event. Instead, it will make an optimal decision on selecting the best device, say d_b, based on the following two criteria:
(i) *Quality:* The self-regulated energy level of d_b is on the higher side to ensure superior QoI index (defined in Section V) of the data collected.
(ii) *Quantity:* Device d_b has sensed maximum events in its vicinity during the given time epoch to guarantee reporting of higher number of events.

We term this as the problem of *Device Selection for Quality and Uninterrupted Information Dissemination (DSQUID)*. It is formally defined as follows:

Definition 1: (DSQUID). It is the problem of selecting non-redundant event information from a set of devices, such that both the quality and quantity of the disseminated information are maximized, subject to constraint on the overall network residual energy budget.

1) NP-Completeness of DSQUID: As mentioned above, E is the set of events whose information will be used in the present time epoch. Let $\mathcal{D}$ be a collection of device contributions $\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_m$, where $\mathcal{D}_i$ comprises the events sensed by device d_i in the present time epoch.

Without loss of generality, we can simplify the $DSQUID$ problem by assuming $\mathcal{D}^* \subseteq \mathcal{D}$ such that (i) each event in E is contained in *at most* one subset in $\mathcal{D}^*$, and (ii) each event in E is contained in *at least* one subset in $\mathcal{D}^*$. Condition (i) ascertains that no two devices who have sensed the *same* event are selected; while condition (ii) ensures that information for *all* events sensed in the current time epoch are reported. Thus, the problem reduces to finding $\mathcal{D}^*$.

If conditions (i) and (ii) are not considered, the simplest way to solve this problem is to first select one device contribution from $\mathcal{D}$ arbitrarily (there are total m devices and assuming each device has contributed, $|\mathcal{D}| = m$), then select another device contribution from the rest of $\mathcal{D}$, and so on, until the selected set $\mathcal{D}^*$ satisfies $\mathcal{D}^* = E$. However, if the above mentioned conditions are imposed, the question becomes hard and cannot be solved deterministically.

Conversely, given a set of device contributions $\mathcal{D}^*$ as a certificate, we can quickly check to determine whether $\mathcal{D}^* = E$ or not. This verification of the certificate can be done in polynomial time. Hence, $DSQUID$ is in the *NP* class.

There exists no polynomial time solution to find the set of device contributions that exactly cover the set of events. In essence, the $DSQUID$ problem is *NP-complete*. To prove this, we reduce a classic *NP-complete* problem, known as the *Exact Cover* [21], to the $DSQUID$ problem.

Definition 2: (**Exact Cover Problem**). Given a collection $\mathcal{S}$ of subsets of a set X, an exact cover of X is a sub-collection $\mathcal{S}^*$ of $\mathcal{S}$ that satisfies two conditions:

- The intersection of any two distinct subsets in $\mathcal{S}^*$ is empty, i.e., the subsets in $\mathcal{S}^*$ are pairwise disjoint. Thus, if $\mathcal{S}_i, \mathcal{S}_j \in \mathcal{S}^*$, then $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset$.
- The union of the subsets in $\mathcal{S}^*$ is X, i.e., the subsets in $\mathcal{S}^*$ cover X. Thus, $\forall \mathcal{S}_i \in \mathcal{S}^*$, $\bigcup_i \mathcal{S}_i = X$.

We do the following construction. Let each $x_i \in X$ corresponds to an event $e_i \in E$. Then $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, ..., \mathcal{S}_m\}$ maps to $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_m\}$, such that $\mathcal{S}_i$ corresponds to $\mathcal{D}_i$. The mapping can be shown in polynomial time. Thus, the set $\mathcal{D}^*$ discussed earlier now corresponds $\mathcal{S}^*$. Hence, if we can find $\mathcal{D}^*$ in polynomial time, we can also solve the *Exact Cover* problem in polynomial time. As the DSQUID problem is *NP-complete*, at best we can find a sub-optimal solution.

2) Heuristic to Solve DSQUID Problem: The heuristic for finding a sub-optimal solution for the $DSQUID$ problem is motivated by the approximation algorithm [22] proposed for the *Maximum Weighted Independent Set (MWIS)* problem which has been used for efficient determination of a maximal weighted independent set in the topology graph of a wireless network [23].

Let $G = (V, A, \omega)$ be a simple weighted undirected graph, where V is the set of vertices, A is the set of arcs (directed edges), and ω is the vertex weighting function such that $\omega : V \mapsto \mathfrak{R}^+$, $\omega(u) \in \mathfrak{R}^+$ for all $u \in V$, $\omega(S) = \sum_{u \in S} \omega(u)$ for any nonempty set $S \subseteq V$ and the set of positive reals $\mathfrak{R}^+$. A subset $I \subseteq V$ is an independent set of G if for any two vertices $u, v \in I$, $\{u, v\} \notin A$. An independent set I of G

is maximum if there is no independent set I' of G such that $\omega(I) < \omega(I')$.

In the $DSQUID$ problem, our objective is to choose a minimal set of devices which has collected better quality information (QoI) for maximum number of events, provided that the data is not collected from two devices which have sensed the same event. Such choice enables dissemination of information of nearly all events occurred in the current time epoch with higher degree of accuracy, subject to the constraint that the energy dissipated does not exceed an overall residual energy budget. The weighing (fitness) function for device selection needs to incorporate both quantity (i.e. sensing of different events) and quality (i.e. accuracy in generated report) of contribution.

For the *MWIS* problem, Sakai *et al.* [22] proposed a generalized weighted greedy algorithm, called $GWMAX$. It is an extension of the existing $GMAX$ algorithm that selects a vertex of maximum degree, removes it and its neighbors from the graph, and iterates this process on the remaining graph until no vertex is remaining, implying the set of selected vertices is an independent set. $GWMAX$ has an approximation ratio of at least $\frac{1}{\Delta}$, where Δ denotes the maximum degree of the given graph $G(V, A, \omega)$. It generalizes the vertex selection rule as:

Select each v_i $(0 \leq i \leq |I| - 1)$ *that satisfies*

$$\sum_{u \in N_{G_i}(v_i)} \frac{\omega(u)}{deg(u)(deg(u)+1)} \geq \frac{\omega(v_i)}{deg(v_i)+1} \tag{2}$$

where, $N_{G_i}(v_i)$ is the set of vertices adjacent to v_i in the subgraph G_i and $deg(v_i)$ is the degree of vertex v_i. This implies that the vertex weight normalized by its degree plus 1 forms the selection criteria for all the vertices, and the one which has maximum normalized weight gets selected.

In the beginning of the $GWMAX$ algorithm, a set I is initialized to be an empty set, and the weight of each node in G is evaluated with Eqn. (2). Then, the $GWMAX$ iteratively selects a node v_i using maximum value in G and adds v_i to I, until no node can be selected. In each iteration, when a node v_i with maximum value in G is selected, G is updated as a subgraph of G_i induced by $V - N_{G_i}(v_i)$. In addition, reevaluation of the weights in the current subgraph is carried out using Eqn. (2). We will use a variant of the $GWMAX$ heuristic as the device selection criteria in the $DSQUID$ problem.

V. DSQUID Algorithm

Given an instance of the $DSQUID$ problem at a particular time epoch containing a set of events $E = \{e_1, e_2, ..., e_n\}$, a set of devices $D = \{d_1, d_2, ..., d_m\}$, and a residual energy budget $\mathcal{E}^t$ at time epoch t, our objective is to map this instance to an instance of $MWIS$ problem. Such mapping to the $MWIS$ problem will generate an undirected weighted graph $G(D, A, \omega')$, where the vertex set is given by the set of devices (i.e., D), ω' is the vertex weighing function (defined later), and the edge set A will have an edge (d_i, d_j) if and only if devices d_i and d_j have sensed and generated reports for one or more *common* events. The physical significance of such graphical model is to capture events which have occurred in the overlapped portions of the devices' sensing boundaries, and to choose the optimal reporting device. Thus, *if* k *is the degree of any node (device) in* $G(D, A, \omega')$*, it means that the event(s) reported by this node (device) has also been reported by* k *other devices.*

Fig. 4 depicts the graphical representation of our system model (refer to Fig. 1) which forms one of the instances of the $DSQUID$ problem captured at a given time epoch.

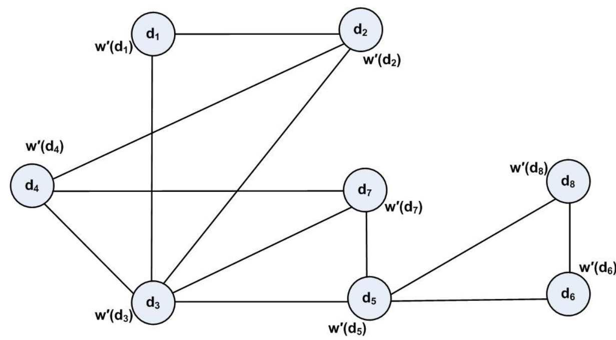


Fig. 4: Graphical Transformation of the $DSQUID$ Problem

The vertex weighing function ω' should take into account the number of events (quantity) sensed by device d_i, as well as the expected QoI index (quality) of the data samples collected. As mentioned earlier, the quality of data samples collected at any time epoch depends on the device's time-varying self or neighbor-regulated energy level $l^t_{d_i}$, which lies in the interval [0, 1]. It also gives a measure of expected accuracy of the data generated by the device. Thus, we customize the weighing function in the $DSQUID$ problem and model it as a weighted regression, where the *number of events sensed* and the *expected QoI index of the collected data samples* are the explanatory variables:

$$\omega'(d_i) = \gamma \cdot n^t_{d_i} + (1 - \gamma) \cdot q^t_{d_i} \tag{3}$$

where, $n^t_{d_i}$ and $q^t_{d_i}$ are the number of events sensed and the expected QoI index of the data collected by device d_i at time epoch t, respectively, and $0 \leq \gamma \leq 1$ is the preference factor which the base station assigns to the two variables. Such preferences are controlled by contextual information, such as the location, temporal biases, and so on. For example, if the event occurrence is frequent within the sensing region, a higher weight can be given to the number of events sensed (quantity). For regions with lower event occurrence rate, higher importance will be given to the QoI index (i.e., quality).

The expected QoI index of data collected by a device explicitly depends on its current energy level. We use the following non-linear relationship [16]:

$$q^t_{d_i} = \alpha \cdot (l^t_{d_i})^{\beta} \tag{4}$$

where, $0 < \alpha < 1$ is the highest QoI index that can be achieved if the device operates at maximum energy level of 1, and $0 < \beta < 1$ is the discounting factor. Both α and β are contextual parameters and vary with application as well as the spatial and temporal aspects of the sensing environment.

Fig. 5 shows the growth of the QoI index for different values of the parameters α and β. Evidently, for lower α and higher β, the QoI index is very low even for maximum energy level. In contrast, it undergoes a rapid growth for higher α and lower β, while it grows at moderate rate if both the parameters have comparable values. Fine tuning these parameters enables the application platform achieve varying levels of QoI and adapt to the spatio-temporal contextual requirements of the application.

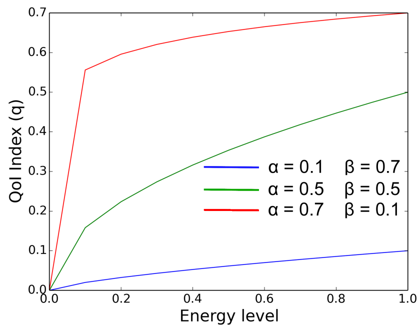


Fig. 5: QoI Index: Parameters

As mentioned earlier, the objective of the $DSQUID$ algorithm is to maximize the dissemination of event information, subject to constraint on the residual energy budget. Such objective is expected to remain valid for all time epochs. Thus, we need to select a set of devices $D^* \subset D$ that maximizes the sum of the weights given by Eqn. (3) in an iterative fashion, provided that the total energy dissipated for transfer of information is under the overall residual energy. In every time epoch, a selected device $d_i \in D^*$ will dissipate a fraction of its individual residual energy to sense the events and then transfer information to the base station. Let $\epsilon^t_{d_i}$ is the energy dissipated by the device d_i at time epoch t for transmission of information. Therefore, energy dissipated by device d_i at the end of time epoch t is given as follows:

$$\epsilon^t_{d_i \in D^*} = l^t_{d_i \in D^*} \cdot (\delta_1 + n^t_{d_i \in D^*} \cdot \delta_2) + \delta_3 \tag{5}$$

where δ_1 is a constant which combines the energy dissipated during the device in the idle mode and that during sensing the environment at a particular time epoch, δ_2 is a constant energy required to transmit information per event, and δ_3 is the fixed energy dissipated to activate the transmitter radio.

The residual energy of device $d_i \in D^*$ for the next time epoch is given as:

$$\varepsilon^{t+1}_{d_i \in D^*} = \varepsilon^t_{d_i \in D^*} - \epsilon^t_{d_i \in D^*} \tag{6}$$

The rest of the devices, which belong to the set $D \setminus D^*$, neither activate their transmitter radio nor transmit any event specific information to the base station. Thus, for any device $d_j \in D \setminus D^*$, the total dissipated energy will only constitute the energy spent for the sensing task and is given as follows:

$$\epsilon^t_{d_j \in D \setminus D^*} = l^t_{d_j \in D \setminus D^*} \cdot \delta_1 \tag{7}$$

Similar to Eqn.(6), the residual energy for the devices not selected for information transfer at the current time epoch is as follows:

$$\varepsilon^{t+1}_{d_j \in D \setminus D^*} = \varepsilon^t_{d_j \in D \setminus D^*} - \epsilon^t_{d_j \in D \setminus D^*} \tag{8}$$

Thus, if T is the total number of time epochs during which the smart city application can provide service, then the $DSQUID$ algorithm utilizing the $GWMAX$ heuristic, finds a solution to the following optimization problem:

$$\begin{aligned} &\text{maximize} && \sum_{i=1}^{|\mathcal{D}^*|} \omega'(d_i) \\ &\text{subject to} && \sum_{i=1}^{|D^*|} \epsilon^t_{d_i \in D^*} + \sum_{j=1}^{|D \setminus D^*|} \epsilon^t_{d_j \in D \setminus D^*} \le \mathcal{E}^t, \forall t \in T \end{aligned} \tag{9}$$

Essentially, the $DSQUID$ algorithm makes a schedule for selecting IoT devices to avoid processing of redundant event information in smart city applications, such that the number of devices chosen is minimum and the number of events reported by these devices is maximum. This is accomplished with the help of the following three steps:

1) Construct the graphical model $G(D, A, \omega')$ from an instance of the $DSQUID$ problem.
2) Establish the independent set I (schedule of devices) using the modified $GWMAX$ algorithm.
3) Disseminate the information obtained towards decision making if a feasible solution to the optimization problem (refer to Eqn.(9)) can be obtained.

A. Communication Protocol

In Section II, we discussed the device energy regulation and device-base station beacon exchange phases within a particular time epoch. During the device energy regulation, each device sends its message of format $\langle$r, $device\_ID, energy\_level$,t$\rangle$ to the successor devices and regulates its own energy level using Eqn. (1), where r denotes the type of message i.e. *regulatory* and t is the present time epoch.

In the device-base station beacon exchange phase, a sequence of communication among the devices and base station takes place, as discussed below.

Step-1: Each device sends beacon message, containing a list of event IDs sensed, to the base station. The message has the following format $\langle sb, device\_ID, energy\_level, [event\_IDs]\rangle$, where sb specifies the type of message which is of *sensing device-to-base station* type, and $[event\_IDs]$ is the list of events which the device has sensed.

Step-2: The base station on receiving the beacons, utilizes Eqns. (3) and (4) to calculate $\omega'(d_i)$ for each device d_i. It then invokes the $DSQUID$ algorithm to choose the optimal set of devices which have sensed disjoint sets of events. The base station sends beacons to the chosen devices asking them to send the information about the events sensed.

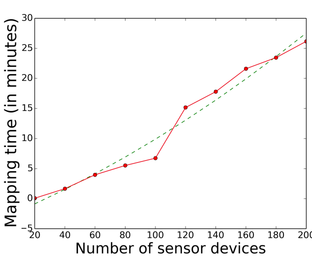
Step-3: The selected devices process and aggregate the collected data samples and transfer the event information to the base station.

VI. Experimental Results

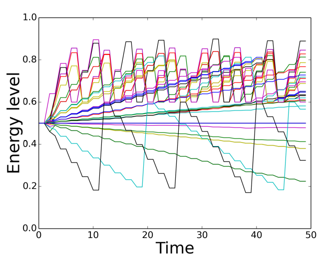
Recall that the goal of the proposed *bioSmartSense* framework is to ensure that the data collection mechanism in IoT-driven smart city applications is energy efficient as well as QoI-aware to address the issue of sustainability. The efficacy of *bioSmartSense* has been extensively studied through the development of a customized simulator (details explained in Section VI-A) using Python's *SimPy Discrete Event Simulation* library [24]. Due to the scarcity of real-data in the study of sustenance of smart city applications, we perform experiments using synthetic data (particularly deployment of devices and event occurrence) generated by our simulator. We also compare the results with a state-of-the-art data collection framework proposed from the perspective of smart city applications [4], which, to the best of our knowledge, is the only work that aligns with our proposed data collection framework.

A. Simulation Settings and Parameters

We consider a deployment area of 2×2 square kilometers comprising a single base station and 50 IoT devices deployed at random locations. The sensing devices are equipped with WiFi connectivity and the transmission range (defined as sensing radius in Section II) is assumed as 100 meters. The base station is also WiFi-enabled and has a router with range of 500 meters. The events occur at random locations within the deployment region and their frequency in each epoch is given by an exponential distribution with mean 20. For every event we construct a boundary with a fixed radius of 50 meters. If an event is sensed by two or more devices whose locations are within the former's boundary, then we assign the same event identifier to the latter.



(a) Mapping Time



(b) Energy Level of Devices

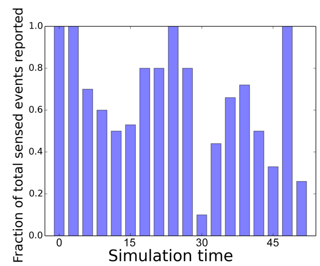
Fig. 6: Performance of GRN

B. Performance of GRN based regulation

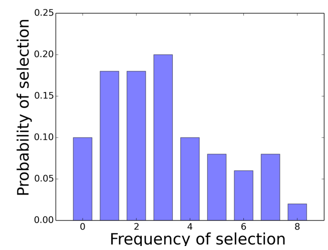
The duration for simulation is 50 time epochs, where the length of an epoch has been assumed to be 30 minutes. Also, each epoch is divided into four time intervals of equal duration to accommodate the following phases: *energy regulation, event sensing, beacon exchange, and information transmission*. These phases are repeated in a cyclic order in subsequent time epochs. Each IoT device is equipped with rechargeable battery energy of 15,000 Joules which makes the total residual energy of the network $15,000 \times 50 = 750,000$ Joules. However, we consider only 500,000 Joules as the initial energy budget for sensing and reporting tasks. We take the preference factor $\gamma = 0.5$ to assign equal weights to both 'quantity' and 'quality' in the device weighing function ω', while the values for the parameters of QoI index are given by $\alpha = \beta = 0.8$.

In this section, we evaluate the performance of our proposed mapping algorithm in terms of the time required for mapping the networked IoT devices to the GRN topology (refer to Section IV-A) and the energy levels of the devices due to self-regulatory property of the GRN.

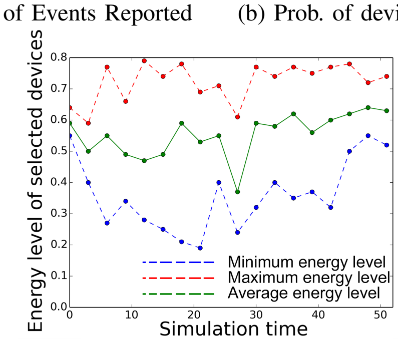
In Fig. 6a, we vary the number of IoT devices constituting the *IoTNet* from 20 to 200 nodes and observe that the mapping time ranges from a few seconds to 25 minutes. Assuming that the deployment of the IoT devices will not change frequently and that the mapping will be exercised only once, the time taken is reasonable. As discussed in Section IV-A, the mapping algorithm has a running complexity of $O(m^2)$. We verify this by applying nonlinear curve fitting to fit the mapping data to a curve of degree two. The resultant fit curve (shown in green dots) has the form $0.09x^2 + 2.31x - 0.87 = 0$.



(a) Frac. of Events Reported



(b) Prob. of device selection



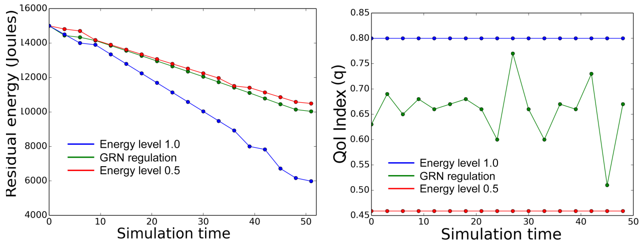
(c) Device Energy Level

Fig. 7: Performance of *bioSmartSense* Framework

We observe the evolution of energy levels of the sensing devices across the entire simulation time. The lower limit (L) and upper limit (U) for energy level of self regulation are 0.2 and 0.8 respectively, and the regulation rate is $\kappa = 0.01$. Fig. 6b depicts the energy levels of different sensors in different colors. The energy level of majority of the sensor nodes lie between 0.5 and 0.8 due to the collective effect of neighbor and self regulations.

C. Performance of bioSmartSense Framework

The objective of the *DSQUID* algorithm is to select a subset of nodes that will report maximum number of non-redundant information of the highest quality. Fig. 7a shows that the average fraction of events reported (defined as ratio of the number of events sensed to those events) range between 0.5 and 1.0. This implies that the proposed framework captures



(a) Residual Energy (b) QoI Index

Fig. 8: Effect of GRN-based Regulation

majority of the events occurring over the sensing regions, which is essential for the application to maintain its relevance.

Fig. 7b shows that likelihood of selection of devices by the proposed framework. It follows a positively skewed bell-shaped curve with maximum probability of selection of device corresponds to the frequency of 3. This implies that the device selection mechanism of *bioSmartSense* is not biased and facilitates judicious use of energy. Thus, our framework avoids choosing the same set of nodes in every time epoch and maximizes the duration of uninterrupted service delivery.

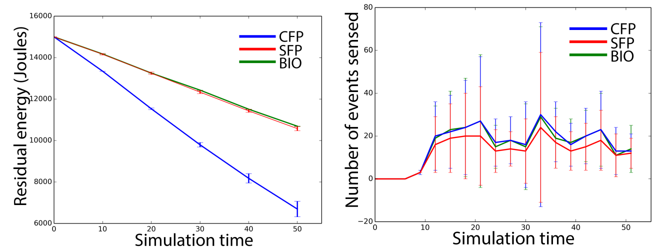
Fig. 7c shows that the average energy level of selected devices toggle between 0.5 and 0.6, while the minimum and maximum energy levels range from 0.35 to 0.75. Thus, GRN-based energy regulation enables our framework choose devices with good blends of high and low energy levels, which will implicitly allow the smart city application to sustain longer without recharging of devices.

D. Effect of GRN-based regulation

We evaluate the effect of GRN-based regulation in *bioSmartSense* by comparing it against systems, where GRN-based node regulation is turned off and energy level of all devices are fixed at 0.5 and 1. Fig. 8a shows that the GRN-based regulation incurs almost equal energy cost as fixed regulation 0.5 and significantly lesser if the energy level is fixed at 1. Thus, as far as saving residual energy is concerned, GRN regulation-based *bioSmartSense* is comparable to the optimal strategy. In contrast, the QoI index of the reports generated by devices selected by *bioSmartSense* lies in between that of fixed regulation 0.5 and 1.0. Here we observe a trade-off between energy-efficiency and QoI index. However, note that a smart application may not require higher quality reports at all the time and for all sensing regions. For example, during peak hours of the day, the application may need higher QoI, and for the remaining time it tolerates moderate data quality. Hence, *bioSmartSense* is overall the best choice for data collection mechanism for sustainability of smart city applications.

E. Comparison with Existing Approach

We compare our framework with a distributed data collection mechanism for smart city application, proposed by Capponi *et. al.* [4]. A key difference in their approach is that instead of fixed IoT devices as the data collecting unit, they considered the built-in sensors of smartphones to generate data for the application platform. The mechanism includes two data collection policies: (i) *Collector Friendly Policy (CFP)* which prioritizes data collection utility as specified by the application platform, and (ii) *Smartphone Friendly Policy (SFP)* which optimizes energy conservation of sensing devices and decides on whether or not to report in the current time epoch. In the interest of a fair comparison, we used the same values (in Watts) for the energy constants $\delta_1 = 3.68, \delta_2 = 0.37$ and $\delta_3 = 0.11 \times 10^{-3}$ from [4].



(a) Avg. Residual Energy (b) Avg. No. of Events Reported

Fig. 9: Comparison with [4]

We consider 50 different scenarios of randomly occurring events. Fig. 9a shows that the average residual energy of the *bioSmartSense* framework marginally outperforms SFP and significantly outperforms CFP. In addition, *bioSmartSense* exhibits the least variation in residual energy for various events scenarios. On the other hand, *bioSmartSense* exhibits similar number of events sensed as CFP and outperforms SFP (Fig. 9b). This shows that our proposed framework not only preserves device residual energy, but also judiciously selects the devices such that majority of the events get reported. Thus, as far as sustainability and relevance of smart city applications are concerned, the data collection strategy in *bioSmartSense* shows dual property of energy consumption and event sensing with quality when compared against the state-of-the-art approach.

VII. Conclusion

In this work we presented a bio-inspired distributed event sensing and data collection framework, called *bioSmartSense*, to address the energy-efficiency and QoI requirements of IoT-based smart city applications. The proposed framework utilizes the regulatory information of *Gene Regulatory Networks (GRNs)* as well as a heuristic for the *Maximum Weighted Independent Set (MWIS)* problem to select a subset of devices to sense and report event information of high quality and quantity, under constrained residual energy budget. We conducted extensive experiments on a customized discrete event simulator and demonstrated that *bioSmartSense* achieves higher event sensing and reporting capability, as well as better energy-efficiency in comparison with state-of-the-art data collection mechanism in smart city applications. As a part of future work, we will study the effect of dynamic sensing radius (w.r.t depleting residual energy) and mobility of the devices on the overall performance. We will also look for relevant real-data of smart city applications and then validate our framework more rigorously.

Acknowledgment

This work is supported by NSF grants under award numbers CNS-1818942, CCF-1725755, CNS-1545037, CNS-1545050.

REFERENCES

[1] C. Fiandrino, A. Capponi, G. Cacciatore, D. Kliazovich, U. Sorger, P. Bouvry, B. Kantarci, F. Granelli, and S. Giordano. Crowdsensim: a simulation platform for mobile crowdsensing in realistic urban environments. *IEEE Access*, 5:3490–3503, 2017.

[2] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi. Internet of things for smart cities. *IEEE Internet of Things Journal*, 1(1):22–32, 2014.

[3] W. Ejaz, M. Naeem, A. Shahid, A. Anpalagan, and M. Jo. Efficient energy management for the internet of things in smart cities. *IEEE Communications Magazine*, 55(1):84–91, 2017.

[4] A. Capponi, C. Fiandrino, D. Kliazovich, P. Bouvry, and S. Giordano. A cost-effective distributed framework for data collection in cloud-based mobile crowd sensing architectures. *IEEE Transactions on Sustainable Computing*, 2(1):3–16, 2017.

[5] T. Das, P. Mohan, V. N Padmanabhan, R. Ramjee, and A. Sharma. Prism: platform for remote sensing using smartphones. In *ACM MobiSys*, pages 63–76, 2010.

[6] J. Wang, J. Tang, G. Xue, and D. Yang. Towards energy-efficient task scheduling on smartphones in mobile crowd sensing systems. *Computer Networks*, 115:100–109, 2017.

[7] M. Tomasoni, A. Capponi, C. Fiandrino, D. Kliazovich, F. Granelli, and P. Bouvry. Profiling energy efficiency of mobile crowdsensing data collection frameworks for smart city applications. In *IEEE MobileCloud*, pages 1–8, 2018.

[8] A. Nazi, M. Raj, M. Di Francesco, P. Ghosh, and S. K. Das. Robust deployment of wireless sensor networks using gene regulatory networks. In *Springer ICDCN*, pages 192–207, 2013.

[9] A. Nazi, M. Raj, M. Di Francesco, P. Ghosh, and S. K. Das. Efficient communications in wireless sensor networks based on biological robustness. In *IEEE DCOSS*, pages 161–168, 2016.

[10] S. Roy, V. K. Shah, and S. K. Das. Characterization of e. coli gene regulatory network and its topological enhancement by edge rewiring. In *EIA BICT (formerly BIONETICS)*, pages 391–398, 2016.

[11] A. Markham and N. Trigoni. Discrete gene regulatory networks (dgrns): A novel approach to configuring sensor networks. In *IEEE INFOCOM*, pages 1–9, 2010.

[12] H. Byun and J. Park. A gene regulatory network-inspired self-organizing control for wireless sensor networks. *International Journal of Distributed Sensor Networks*, 11(8):789434, 2015.

[13] A. Pughat and V. Sharma. Performance analysis of an improved dynamic power management model in wireless sensor node. *Digital Communications and Networks*, 3(1):19–29, 2017.

[14] H. Hernández, C. Blum, and G. Francès. Ant colony optimization for energy-efficient broadcasting in ad-hoc networks. In *International Conference on Ant Colony Optimization and Swarm Intelligence*, pages 25–36. Springer, 2008.

[15] J. Nielsen. Principles of optimal metabolic network operation. *Molecular Systems Biology*, 3(1):126, 2007.

[16] C. H. Liu, B. Zhang, X. Su, J. Ma, W. Wang, and K. K Leung. Energy-aware participant selection for smartphone-enabled mobile crowd sensing. *IEEE Systems Journal*, 11(3):1435–1446, 2017.

[17] T. Schaffter, D. Marbach, and D. Floreano. Genenetweaver: in silico benchmark generation and performance profiling of network inference methods. *Bioinformatics*, 27(16):2263–2270, 2011.

[18] P. Erdős and A. Rényi. On random graphs. *Publicationes Mathematicae Debrecen*, 6:290–297, 1959.

[19] S. Roy, M. Raj, P. Ghosh, and S. K. Das. Role of motifs in topological robustness of gene regulatory networks. In *IEEE ICC*, pages 1–6, 2017.

[20] J. EM Hornos, D. Schultz, G. CP Innocentini, J. Wang, A. M Walczak, JN Onuchic, and PG Wolynes. Self-regulating gene: an exact solution. *Physical Review E*, 72(5):051907, 2005.

[21] R. M Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.

[22] S. Sakai, M. Togasaki, and K. Yamazaki. A note on greedy algorithms for the maximum weighted independent set problem. *Discrete Applied Mathematics*, 126(2-3):313–322, 2003.

[23] S. Basagni. Finding a maximal weighted independent set in wireless networks. *Telecommunication Systems*, 18(1-3):155–168, 2001.

[24] N. Matloff. Introduction to discrete-event simulation and the simpy language. *Davis, CA. Dept of Computer Science. University of California at Davis. Retrieved on August*, 2(2009), 2008.