Revisiting Data Collection in Robotic Sensor Networks: A Budget-Constrained Traveling Salesman Perspective

Soham Patil*, Yutian Chen†, Bin Tang*
*Department of Computer Science, California State University Dominguez Hills
spatil1@toromail.csudh.edu, btang@csudh.edu
†Economics Department, California State University, Long Beach, Yutian.Chen@csulb.edu

***Abstract*—We focus on *robotic sensor networks* (RSNs), wherein mobile data collectors or robots are dispatched into the sensor field to collect data from the sensor nodes, and study a new algorithmic problem called *battery-constrained data collection in RSNs* (BC-DCR). Given an RSN of sensor nodes with varying numbers of sensory data packets to be collected and a robot with limited battery power, the goal of the BC-DCR is to dispatch the robot into the sensor field to collect the maximum number of data packets before it runs out of battery power and returns to the depot for recharging. Although extensive research has been conducted to achieve various performance objectives of data collection in RSNs, not much work has focused on the robot's limited battery power. It is critical to consider the robot's limited battery power to optimize the data-collecting performance of a large-scale RSN. We show that at the core of the BC-DCR is a new variation of the classic traveling salesman problem called the Budget-Constrained Traveling Salesman Problem (BC-TSP), which has not been adequately solved. We design an Integer Linear Programming (ILP)–based optimal algorithm and a time-efficient iterative greedy algorithm to solve the BC-TSP. Via extensive simulations using real measurements of battery power and mobility models of robots, we show that a) our algorithms outperform the existing work by collecting 29.1% more packets with the same battery power of the robots and b) our BC-TSP-based approach achieves 32.02% more network lifetime of the RSN compared to the existing approach.**

Keywords – Data collection, robotic sensor networks, budget-constrained traveling salesman problem.

I. Introduction

Background and Motivation. Since its inception in the 1990s, wireless sensor networks, which help humans collect diverse information from the physical world, have evolved from research labs to real-world applications in military, health, and industrial environments [24], [17]. In recent years, with the strides made in robotic research and development [14], [9], [18], mobile robots have been introduced into wireless sensor networks to enhance their system performance and efficiency in environment monitoring, intrusion detection, and search and rescue [15], [10]. We refer to the sensor networks equipped with the mobile robots as *robotic sensor networks* (RSNs).

One of the main tasks that mobile robots help to perform in RSNs is data collection [8], [16]. In traditional sensor networks, sensory data is transmitted back to the base station in a multi-hop manner, which is energy-expensive and could deplete the limited battery power of sensor nodes quickly. By dispatching the battery-rechargeable robots into the sensor field to collect the data and bring it back to the base station, the energy bottleneck of the sensor network is migrated from the sensor nodes to the mobile robots, which can be recharged repeatedly. Consequently, the lifetime of the RSN is largely prolonged, and its functionality is significantly improved.

There is extensive research to achieve various goals of data collection in the RSNs, including minimizing the length of data collecting tour or energy consumption of the robots [21], [11], maximizing network lifetime or network utility [19], [11], [28], and retrieving as much sensed data as possible given a deadline [26], [29], [5]. However, almost all the existing research assumes that the robot has enough battery power to achieve any objective above by visiting any part of the sensor field before returning to the depot. However, a robot with finite (but rechargeable) battery power can only travel a limited distance and visit some parts of a sensing field before running out of battery and returning to the depot for recharging [22]. This is especially true for large-scale sensing applications, including environmental monitoring and underwater exploration, wherein robots are dispatched to collect data in a vast area. Therefore, it is essential to consider the limited battery power of a robot when it is operated in an RSN environment.

Our Contributions. In this paper, we focus on data collection in RSNs where a robot cannot collect all the sensory data in the RSN due to its battery constraint. We propose a new algorithmic problem called BC-DCR: *battery-constrained data collection in RSNs*. Given an RSN of sensor nodes generating varying numbers of data packets and a robot with limited battery power, the goal of the BC-DCR is to select a subset of sensor nodes and find a route for the robot to visit them to collect the maximum number of data packets before it returns to the depot for recharging. Here, we assume the more packets collected, the more information can be gathered from the environment, thus, better performance of the RSN. BC-DCR is a new problem that has not been studied in the RSN community to the extent of our knowledge.

We show that BC-DCR is equivalent to a new variation of the classic Traveling Salesman Problem [12], which we refer to as the *Budget-Constrained Traveling Salesman Problem* (BC-TSP). Given a weighted complete graph where each edge has a weight, each node has a prize to be collected, and a salesman has a given budget $\mathcal{B}$. The BC-TSP is to find a *prize-collecting cycle* $R = \{r = v_1, v_2, v_3, ..., v_x = r\}$ such that its total prize $P_R = \sum_{i \in R} p_i$ is maximized while its cost $C_R = \sum_{i=1}^{x-1} w(v_i, v_{i+1}) + w(v_x, v_1) \leq \mathcal{B}$. BC-TSP is NP-hard [27]. Although BC-TSP has been studied in the theory community, we are the first to apply it to model data collection in RSN using real parameters and measurements of robot battery power and mobility models.

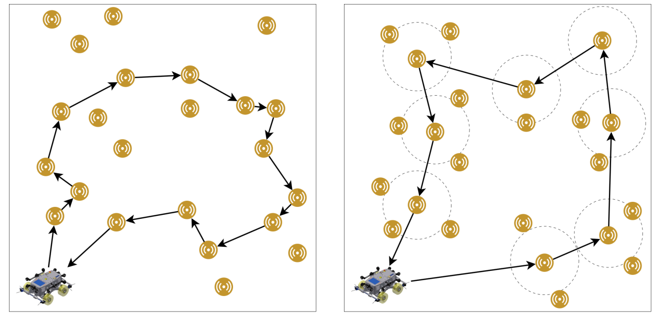
We design two algorithms to solve the BC-DCR: an Integer Linear Programming (ILP)–based optimal solution and an iterative greedy algorithm. Via extensive simulations using real measurements of battery power and mobility models of robots, we show that a) our algorithms outperform the existing work by collecting 29.1% more packets with the same battery power of the robot and b) our approach achieves 32.02% more network lifetime of the RSN compared to the existing literature using a different approach [21].

Paper Organization. Section II formulates the BC-DCR. Section III proposes two combinatorial algorithms, including an ILP optimal solution and an efficient greedy heuristic algorithm, to solve the BC-DCR. Section IV compares our algorithms with the existing research and discusses the results. Section V reviews all the related work. Section VI concludes the paper with future works.

II. Problem Formulation of BC-DCR

Network Model. We model the RSN, shown in Fig. 1(a), as a rectangular area of l meters by m meters. Let $V_s = \{1, 2, ..., |V_s|\}$ denote the set of $|V_s|$ sensor nodes randomly located in the RSN, with $i \in V_s$ located at the location (x_i, y_i), where $0 \leq x_i \leq l$ and $0 \leq y_i \leq m$. Each sensor node $i \in V_s$ has $d_i \geq 0$ number of data packets to be collected; each packet has a size of k bits. Let $r = (0, 0)$ denote the depot of the RSN, wherein both a base station and a charging station are installed. $d_r = 0$ as r is not a sensor node. The robot is dispatched from the depot to collect data in the RSN. Before its battery power runs out, it returns to the depot to upload the data to the base station and recharge its battery at the charging station. Let $V = V_s \cup \{r\}$ be the set of sensor nodes and the depot. Let $d(i,j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$, $\forall i, j \in V$, denote the distance between any pair of sensor nodes or a sensor node and the depot. Each sensor is powered by an unreplenishable battery, and the robot has a rechargeable battery with an initial full power of $\mathcal{E}$ joules.

Mobility Energy Model of the Robot. For the energy mode of the robot, as our goal is to find an energy-efficient route for the robot to collect data under battery power constraints, we focus on the energy consumption of a robot due to its mobility. *Mobility energy* of a robot includes all the energy needed to keep the robot in motion, such as the drive motor, steering motor, and related energy losses. Therefore, the mobility energy consumption of a robot is directly related to its traveled distance. Xiao et al. [30] show that the ideally achievable distance d (in meters) of a battery-powered wheeled mobile robot on one single charge is



(a) BC-TSP. (b) CSP.

Fig. 1. BC-TSP and CSP-based battery-constrained data collection.

$$d = \frac{\mathcal{E}}{w \times C_{crr}} \tag{1}$$

, where $\mathcal{E}$ (in joules) is the battery power of the robot, w (in Kg) is the weight of the robot, and C_{crr} represents the coefficient of rolling friction depending on the terrain type. We denote $\mu = w \times C_{crr}$ as the *mobility energy coefficient* (with a unit of joule/meter), indicating the amount of battery power consumed per unit traveled distance by the robot.

Data Collection Model. When the robot wirelessly collects the data packets from the sensor nodes, their energy consumption follows the first-order radio model [13]. We assume the robot and sensor nodes all have a transmission range of T_r meters; that is, the robot can collect data packets from a sensor node directly if their distance is within T_r. Assume the robot is currently located at node $j \in V$, when sensor node i sends a k-bit data packet to the robot over their distance $l_{i,j} \leq T_r$, the *transmission energy* spent by sensor node i is $E_i^t(j) = \epsilon_{elec} * k + \epsilon_{amp} * k * l_{i,j}^2$, the *receiving energy* spent by the robot is $E_j^{re} = \epsilon_{elec} * k$. Here $\epsilon_{elec} = 100nJ/bit$ is the energy consumption per bit on the transmitter and receiver circuit, and $\epsilon_{amp} = 100pJ/bit/m^2$ is the energy consumption per bit on the transmit amplifier. This energy model also applies to two sensors sending packets between them. We leave the more general case that the robot can adjust its transmission range for more energy-efficient data collection as future work.

In addition, as the battery power of the robots can be recharged while that of the sensor nodes cannot, we need to conserve the energy consumption of the sensor nodes as much as possible; otherwise, energy-depleted sensor nodes will render the RNS inoperable. We thus assume that when collecting data from sensor node i, the robot should arrive at a i's location to collect its data, as shown in Fig. 1(a). That is, the robot moves from its current location to i's location (x_i, y_i) following the straight line between them. It then directly collects the data packets from i. This way, the sensor node's transmission energy of sending one data packet to the robot is reduced to a minimal $\epsilon_{elec} * k$.

TABLE I
NOTATION SUMMARY

Notation	Description
V_s	The set of $\|V_s\|$ sensor nodes
r	The depot where the robot is located and recharged
V	$V = V_s \cup \{r\}$
T_r	Transmission range of sensor nodes and the robot
$\mathcal{E}$	The initial battery power of the robot
$d(i,j)$	Distance between two nodes (sensor nodes or robot) i and j
μ	Mobility energy coefficient of the robot
R	The data collecting route of the robot
E_R	The battery power consumption of the robot on route R
D_R	Total number of packets collected by the robot on route R
$E_u^t(v)$	Transmission energy spent by u to transmit one packet to v
E_v^{re}	Receiving energy spent by v to receive one packet
$x_{i,j}$	Decision variable if edge (i,j) on route R in ILP
u_i	Position variable for node i on route R in ILP

Problem Formulation of BC-DCR. Let $R = \{r, v_1, v_2, ..., v_x, r\}$ denote a *data-collecting cycle* of the robot, where the robot starts from depot r, visits a sequence of sensor nodes $v_i \in V_s$ to collect their data packets, and finally returns to depot r. Denote the battery power spent by the robot along R as E_R; $E_R = \mu \times \big(d(r, v_1) + \sum_{i=1}^{x-1} d(v_i, v_{i+1}) + d(v_x, r)\big)$. Denote the total number of data packets the robot collects along R as D_R; $D_R = \sum_{i=1}^{x} d_{v_i}$. Given the initial battery power $\mathcal{E}$ of the robot, the goal of the BC-DCR is to find a data-collecting cycle R for the robot to traverse to maximize D_R before running out of its battery power and return to r; that is, $E_R \leq \mathcal{E}$. Next, we introduce *budget-constrained traveling salesman problem* (BC-TSP) [27] and show that BC-DCR is the equivalent of BC-TSP.

BC-TSP. Given a weighted complete graph $G(V, E)$, edge $(u, v) \in E$ has a weight $w(u, v) \geq 0$, and each node $i \in V$ has a prize $p_i \geq 0$ to be collected. Let $r \in V$ be the node where the salesman starts and ends its route, and $\mathcal{B}$ his budget, the distance he can travel before returning to r. The goal of the BC-TSP is to find a *prize-collecting cycle* $R = \{r = v_1, v_2, v_3, ..., v_x = r\}$ such that its total prize $P_R = \sum_{i \in R} p_i$ is maximized while its cost $C_R = \sum_{i=1}^{x-1} w(v_i, v_{i+1}) \leq \mathcal{B}$. BC-TSP is NP-hard [27].

Theorem 1: BC-DCR is equivalent to BC-TSP.

Proof: In BC-DCR, the RSN can be represented as a complete graph $G(V, E)$ where $V = V_s \cup \{r\}$ includes the set of sensor nodes V_s and the depot r, and for any edge (i, j), its weight $w(i, j) = \mu \times d(i, j) = \mu \times \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$. Let the number of data packets d_i at node $i \in V_s$ in BC-DCR be the prize p_i available at node i in the BC-TSP, and let the initial battery power $\mathcal{E}$ of the robot in the BC-DCR be the initial budget $\mathcal{B}$ of the traveling salesman in the BC-TSP. Then, finding a data-collecting cycle for the robot to collect a maximum number of data packets while staying within its battery power of $\mathcal{E}$ in BC-DCR is the same as finding a prize-collecting cycle for the salesman to collect maximum prizes while staying within his budget of $\mathcal{B}$ in BC-TSP. Thus, BC-DCR is equivalent to BC-TSP.

Rationale of Modeling BC-DCR as BC-TSP. As the robot can be recharged while the sensor nodes can not (we leave the case of rechargeable sensor nodes as future work), a critical goal of the BC-DCR is to save sensor nodes' energy during data collection to prolong the network's lifetime. Following the above first-order radio model, the distance between the robot and sensor nodes should be zero to minimize a sensor node's transmission energy. That is, the robot should arrive at a sensor node's location to collect its data, as shown in Fig. 1(a). This way, the sensor node's transmission energy of sending one data packet to the robot is reduced to a minimal $\epsilon_{elec} * k$.

State-of-the-Art. Existing research of of data collections in the RSN [21], [26], [29] mainly take a *covering salesman problem* (CSP) approach, as shown in Fig. 1(b). The goal of the CSP is to select a set of *polling points* [21] or *rendezvous points* [26], [29], which are visited by the robot to collect the data packets and minimize the length of the data-gathering tour of such nodes. The sensor nodes transmit their data packets to the nearest polling point directly [21] or to the nearest rendezvous point via multi-hop manner [26], [29]. As Ma et al. [21] is the most representative work in this approach, we compare our work with theirs. They proposed a so-called *spanning tree covering algorithm*, which has three steps. First, it uses a minimum spanning tree-based greedy algorithm to find the polling nodes to be included in the data-gathering route. Second, it uses an ILP to compute the optimal TSP route among the polling nodes and depot. Finally, it uses the well-known 2-approximation algorithm for the TSP [6] to find the final route for the robot.

Although CSP can collect more data packets than BC-TSP does under the limited battery power of the robot, there are several disadvantages of the CSP approach. First, it is time-consuming to compute the polling points and their route, as it is ILP-based. Second, in the CSP model, sensor nodes must spend battery power transmitting data packets to the polling nodes, a significant energy drain to the RSN. Our simulations show that the BC-TSP-based approach achieves 32.02% more network lifetime than the CSP-based approach. Finally, like most existing work, CSP assumes the robot has enough battery to visit the entire RSN to collect data packets. This assumption is no longer valid, considering robots have limited battery power, and consequently, they cannot visit the entire sensor field in many large-scale applications.

III. ALGORITHMIC SOLUTIONS FOR BC-DCR

We first design an Integer Linear Program (ILP)-based solution to solve BC-DCR optimally. We then design a more time-efficient greedy algorithm to solve BC-DCR.

ILP Solution. We formulate below integer program ILP(A). *Decision variable* $x_{i,j}$ indicates if edge (i, j) is on the data-collecting route (i.e., node j is visited immediately after node i is visited); $x_{i,j} = 1$ if so and 0 otherwise. We introduce $|V_s|$ *order variables* u_i, $i \in V_s$, to indicate the order in which the nodes are visited. $u_r = 1$ as r is the starting node and $u_i < u_j$ indicates that node i is visited before node j (but not necessarily immediately). $u_i - 1$ equals the number of edges the robot has traversed after it has visited sensor node i. Note that $V = V_s \cup \{r\}$.

$$(A) \qquad \max \sum_{i \in V} \sum_{j \in V} d_i \cdot x_{i,j} \tag{2}$$

s.t.

$$x_{i,j} \in \{0, 1\}, \quad \forall i, j \in V \tag{3}$$

$$\sum_{j \in V_s} x_{r,j} = \sum_{i \in V_s} x_{i,r} = 1 \tag{4}$$

$$\sum_{i \in V} x_{i,k} = \sum_{j \in V} x_{k,j} \leq 1, \quad \forall k \in V \tag{5}$$

$$\sum_{i \in V} \sum_{j \in V} w_{i,j} \times x_{i,j} \leq \mathcal{E}, \tag{6}$$

$$2 \leq u_i \leq |V|, \quad \forall i \in V_s \tag{7}$$

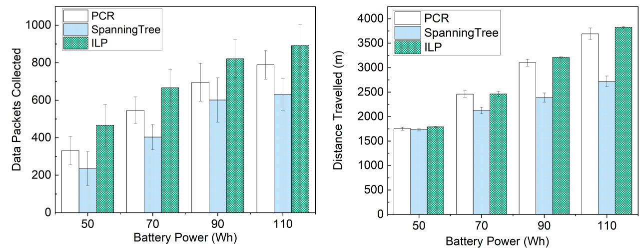
$$u_i - u_j + 1 \leq |V_s| \times (1 - x_{i,j}), \quad \forall i, j \in V_s \tag{8}$$

Objective function 2 is to maximize the total number of collected data packets. Constraint 3 is the integer constraint of $x_{i,j}$. Constraint 4 guarantees that the data-collecting route starts and ends at node r. Constraint 5 ensures the connectivity of the path and that each node is visited at most once. Constraint 6 guarantees that the total battery power spent by the robot on the data collecting path does not exceed its initial battery power of $\mathcal{E}$. Constraints 7 and 8 are *Miller–Tucker–Zemlin (MTZ) Subtour Elimination Constraints* [3], which guarantees that there is one global tour visiting all the selected vertices (otherwise, there could be multiple subtours each visiting only a subset of the selected vertices).

As ILP(A) is time-consuming to compute, we present a more time-efficient greedy algorithm to solve the BC-DCR next. We first give the below definition.

Definition 1: (**Battery-Feasible Sensor Nodes.**) Given the current sensor node s where the robot is located and the robot's current remaining battery power B, the *battery-feasible sensor nodes*, denoted as $\mathcal{F}(s, B)$, is a set of sensor nodes that the robot can visit to collect their data packets and then return to depot r without running out of its battery power. That is, $\mathcal{F}(s, B) = \{u | u \in U \ \wedge \mu \times \big(d(s, u) + d(u, r)\big) \leq B\}$, where U is the set of sensor nodes that have not been visited. □

Greedy Algorithm. Given two nodes $u, v \in V$, and the robot is at node u, we define the *prize cost ratio* of visiting v, denoted as $pcr(u, v)$, as the ratio between the data packets available at v and the distance $d(u, v)$ between u and v. That is, $pcr(u, v) = \frac{d_v}{d(u,v)}$. Algo. 1 works in rounds. In each round, located at the node s and with an available budget B, the robot checks if there are still unvisited battery-feasible nodes (line 2). If so, it visits the one with the largest prize cost ratio and updates all the data-collecting information accordingly (lines 3-7). The robot stops when all nodes have been visited, or none of the unvisited nodes are battery-feasible. At this point, the robot finishes the data-collecting process and returns to the depot r. It updates and returns the route with its total cost, total prizes collected, and the remaining battery power of the robot (lines 9 and 10). Its time complexity is $O(|V|^2)$.



(a) Collected Packets. (b) Total Distance Traveled.

Fig. 2. Comparing ILP, PCR, and SpanningTree.

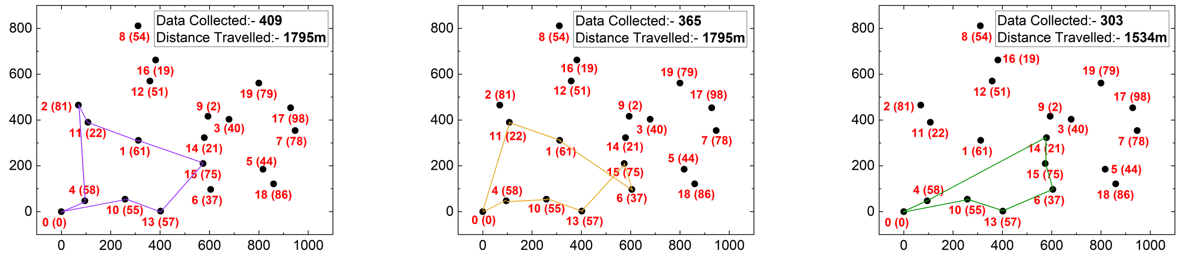
Algorithm 1 Greedy Algorithm for BC-DCR.

Input: A RSN graph $G(V, E)$, depot r, and initial battery $\mathcal{E}$;
Output: A data-collecting route R, its cost C_R and prize D_R.
Notations: R: the current route found, starts from r;
C_R: the distance of R, initially zero;
D_R: the packets collected on R, initially zero;
U: the set of unvisited nodes, initially $U = V_s$;
s: the node where the robot is located currently;
B: current remaining battery of the robot, initially $\mathcal{E}$;

1: $s = r$, $R = \{r\}$, $C_R = D_R = 0$, $B = \mathcal{E}$;
// if not all battery-feasible nodes are visited
2: **while** $(U \neq \phi \wedge \mathcal{F}(s, B) \neq \phi)$ **do**
3: Let $u = \text{argmax}_{v \in \mathcal{F}(s,B) \cap U} pcr(s, v)$;
4: $R = R \cup \{u\}$;
5: $C_R = C_R + d(s, u)$, $D_R = D_R + d_u$;
6: $B = B - d(s, u) \times \mu$, $U = U - \{u\}$;
7: $s = u$;
8: **end while**
9: $R = R \cup \{r\}$, $C_R = C_R + d(s, r)$, $B = B - d(s, r) \times \mu$;
10: **return** R, C_R, D_R, B.

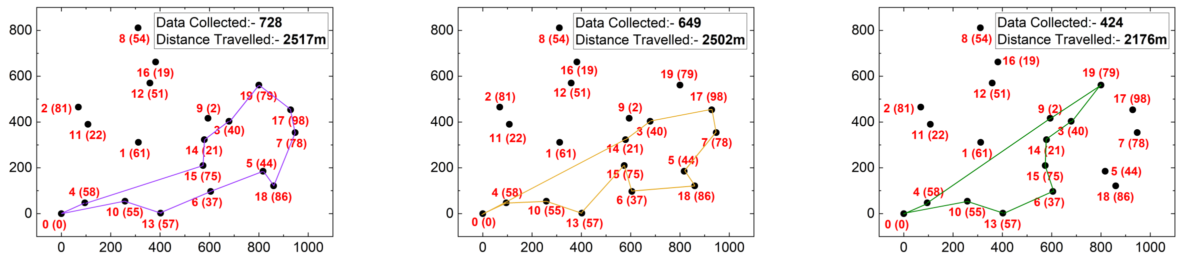
IV. Performance Evaluation

Experiment Setup. We write our own simulator in Java on Windows 11 with AMD Processor (AMD Ryzen 5 4000 Series 6-Core) and 24GB of DDR4 Memory. We refer to the ILP-based optimal solution as **ILP**, the prize cost ratio-based greedy algorithm (i.e., Algo. 1) as **PCR**, and the spanning-tree-based covering algorithm [21] as **SpanningTree**. We use CPLEX [2] for ILP computation in both ILP and SpanningTree. A depot is located at (0,0), where a robot is dispatched to the BSN to collect the data packets and returns before running out of battery power. Each sensor node has generated a random number of data packets in [0, 100], and each packet is 400B. In all plots, each data point is an average of ten runs, for each of which a different RSN instance is created. We compare all the algorithms using the same RSN instance for each run for a fair comparison. The error bars indicate 95% confidence intervals. For the *mobility energy coefficient* $\mu = w \times C_{crr}$, as [30] shows that the weight w of a



(a) ILP. (b) PCR. (c) SpanningTree.

Fig. 3. Visually comparison in an RSN of 1000m by 1000m with 20 nodes. $\mathcal{E} = 50$Wh with maximum distance of 1800m.



(a) ILP. (b) PCR. (c) SpanningTree.

Fig. 4. Visually comparison in an RSN of 1000m by 1000m with 20 nodes. $\mathcal{E} = 70$Wh with maximum distance of 2520m.

typical robot is around 600 Kg while terrain type $C_{crr} = 0.17$, we set μ as 100 J/m. Thus, a 1Wh (i.e., 3600J) amount of battery can power a robot to travel 36 meters.

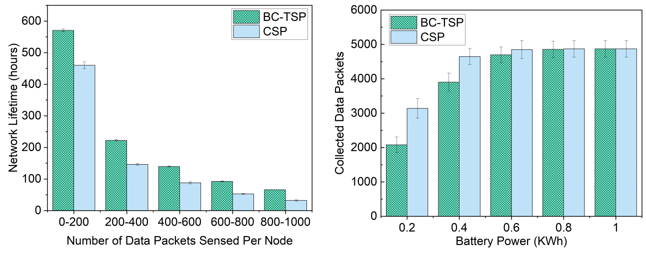
Compare ILP, PCR, and SpanningTree. We generate RSNs of 1000m by 1000m with 20 nodes randomly placed and vary the initial battery of the robot from 50Wh to 110Wh. We set T_r as 100m and assume no two sensor nodes can communicate with each other directly. Therefore, under SpanningTree, the robot must reach each sensor node to collect its packets. Fig. 2(a) shows the number of packets the robot collects before it runs out of its battery and safely returns to the depot. We observe that with the increase in battery power for each algorithm, the packet collected and distance traveled by the robot increase accordingly. ILP collects the maximum number of data packets at each battery level, demonstrating its optimality. PCR outperforms the existing work of SpanningTree by collecting 29.1% more data packets than SpanningTree. Fig. 2(b) shows the distance traveled by the robot for all the algorithms. As the battery power of the robot limits the traveled distance, they all give similar distances except for the SpanningTree, which is caused by its oblivion of data packets.

Fig. 3 and 4 visualize different algorithms with an initial battery power of 50Wh and 70Wh, respectively. It shows that in both cases, PCR can compute a data-collecting route similar to that of ILPs. This shows that PCR is a competitive data-collecting algorithm as it strikes a balance between packets collected and battery consumption. For SpanningTree, as it follows Prim's algorithm to grow a minimum spanning tree (MST) connecting all the polling nodes, it only focuses on the distance (i.e., battery cost) of the robot without paying attention to the number of packets available at the sensor nodes; thus it does not fare well compared to other data-collecting algorithms. Table II shows the execution time of different algorithms with different $\mathcal{E}$. We observe that for the SpanningTree, the execution time decreases with more battery power. This is due to how the MST is formed and the 2-approximation algorithm it uses, wherein the edges in MST do not grow exponentially when the battery power increases.

TABLE II
EXECUTION TIME (MS) OF DIFFERENT ALGORITHMS W.R.T. ROBOT BATTERY POWER.

Battery power $\mathcal{E}$ (Wh)	PCR	SpanningTree	ILP
50	4	38	54
70	5	34	55
90	6	27	57
110	8	26	58

Comparing BC-TSP- and CSP-based Data Collection Approaches. Finally, we compare our BC-TSP approach with the existing CSP approach [21] to study their pros and cons. We consider dense networks of 2000m by 2000m with 100 nodes and set the T_r as 200m. CSP works well when the network is dense enough that the robot has a few sensor nodes within its transmission range to collect data packets. The initial battery power of each sensor node is 6480 Joules, which is the amount of energy stored in a typical AAA battery [1]. We focus on a continuous data sensing and collection scenario that takes place in rounds. In each round of one hour, sensors generate



(a) Network Lifetime.

(b) Collected Packets.

Fig. 5. Comparing BC-TSP and CSP.

random numbers of packets in a pre-specified range, and then the robot is dispatched to collect the data packets. We aim to find the network lifetime achieved and packets collected by both approaches. Here, the network lifetime is defined as when the first sensor node in the RSN depletes its battery power.

Halgamuge et al. [20] studied different factors affecting the sensing energy and proposed a sensing energy model for generating b-bit packet as $E_s(b) = bV_{sup} \cdot I_{sens} \cdot T_{sens}$, where V_{sup} and I_{sens} are the supply voltage and current required for sensing activity and T_{sens} is the time duration for sensing one bit of information. As their typical values are 2.7 V, 25 mA, and 0.5 ms, respectively (Table 3, [20]), the sensing energy of generating a packet of 400B is calculated as 0.108 Joules.

Fig. 5(a) shows that the BC-TSP-based data collection yields up to 32.02% longer lifetime than CSP-based under different ranges of packets generated by each sensor in each round. This is because, in CSP, sensor nodes spend battery power not only in sensing but also in transmitting data packets to the robot, while in our BC-TSP model, sensor nodes only spend energy in sensing. Fig. 5(b) shows that the CSP approach collects more data packets than BC-TSP in each round, although the difference diminishes with increased battery power of the robot. The robot in BC-TSP can only collect data packets from the node it visits, while in CSP, it collects the data packets from all the sensor nodes within T_r. This demonstrates a tradeoff between data packets collected and network lifetime achieved in an RSN with the battery-constrained robot.

V. Related Work

In this section, we review the existing data-collecting techniques in RSN that inspire our research and the existing work in the theory community that solves the BC-TSP.

Data Collection in the RSN. Luo et al. [19] was one of the first to introduce robot mobility into wireless sensor networks. They proposed a data-gathering scheme to minimize the maximum average load of a sensor by jointly considering the problems of movement planning of robots and data-gathering routing. They assumed the sensor field was a circle and mainly used geometric calculations as the technique.

Ma et al. [21] instead took a graph-theoretical approach and modeled data-gathering in sensor networks as a *covering salesman problem* (CSP) [7]. The goal of the CSP is to minimize the length of the data-gathering tour of *polling points* visited by the robot, where the polling points cover all the sensor nodes in the network. That is, sensor nodes within the transmission range of a polling point will directly transmit their data packets to the polling point, which the robot will then collect. Guo et al. [11], [28] introduced wireless energy charging into mobile data collection and formulated and solved a network utility maximization problem considering energy balance and the bounded sojourn time of the mobile robot.

However, the above works did not address delay-sensitive applications, wherein all sensed data must be collected within a given time constraint. To address this problem, Salarian et al. [26] introduced *rendezvous points* (RPs), wherein a robot only visits RPs while sensor nodes that are not RPs forward their sensed data via multi-hop to the nearest RP. They designed a weighted rendezvous planning heuristic algorithm that enables a mobile sink to retrieve all sensed data within a given deadline while conserving the energy expenditure of sensor nodes. This approach is further improved by Wang et al. [29], which considered sensor nodes to have limited buffer sizes while producing data with different speeds and designed an efficient path-planning algorithm for reliable data-gathering.

All the above works assume the mobile robots have enough battery power to collect all the sensory data in the RSN. In a large-scale sensor field, it is possible that the robot does not have enough battery power to visit all the sensor nodes. In this case, a critical question is scheduling the robot to collect as many data packets as possible before recharging at the charging station. We formulate it as a graph-theoretical problem and show that it gives rise to a new variation of the well-known traveling salesman problem, which we refer to as a *budget-constrained traveling salesman problem*. Our work is the first to focus on data collection in the RSN, explicitly considering the battery constraint of the robot. Our other observation of the above covering salesman-based approach is that it still uses one or multi-hop wireless communication for data collection, which could quickly deplete sensor nodes' battery power in data-intensive sensing applications, rendering the entire RSN inoperable. In contrast, in our budget-constraint traveling salesman problem, the robot (i.e., the traveling salesman) must visit each sensor node directly to collect its data packet, dramatically alleviating the sensor node's energy depletion. Using realistic measurements of robot battery power and mobility characteristics, we show that our approach can increase the network lifetime by 32.02% compared to the existing approach [21].

Chen et al. [5] proposed to find an optimal data harvesting path to collect as much data as possible within a time duration and devised a constant-factor approximation algorithm. The time constraint they considered is equivalent to the robot battery power constraint in this paper. However, their data collection model is based on the CSP approach, which could deplete sensor nodes' energy and result in a short network lifetime. They assumed that each sensor node has one unit of data message, and the goal is to cover as many sensors as possible. In our model, however, different sensors have different numbers of data packets, and the goal is to collect as many data packets as possible with the battery constraint.

BC-TSP Research in Theory Community. BC-TSP has been studied in theory and operations research community [27], [23], [4]. Sokkappa et al. [27] was one of the first to study this problem and prove it is NP-hard. They found it important to consider a node's neighborhood when including it in the route. This is because a low-value node that brings the route closer to many other nodes may be more desirable than an isolated node of high value. This inspires our greedy algorithm that finds the node with the maximum prize cost ratio. Levin et al. [4] studied a related *budget prize collecting tree problem*, which finds a subtree with maximum prizes while the cost of the tree stays in a budget. They proposed a $(4+\epsilon)$-approximation algorithm. Paul et al. [23] considered constrained versions of the prize-collecting traveling salesman and the minimum spanning tree problems, wherein the goal is to maximize the number of vertices in the returned tour/tree subject to a bound on the tour/tree cost. They proposed a 2-approximation algorithm based on a primal-dual approach. Ruiz et al. [25] studied quota-driven TSP problem. The goal of the traveling salesman is to find a route from s to t of minimum distance such that the sum of the prizes on the route reaches a preset quota. This paper shows a real network application of the BC-TSP, which is the battery-constrained data collection in RSN. Considering that many network-related parameters (e.g., spatial correlation of data generation in the RSN) can be incorporated into the BC-DCR, many new variations of BC-TSP could exist to be further studied.

VI. Conclusion and Future Work

We focus on the robot's limited battery power and identify, formulate, and solve BC-DCR, a new algorithmic problem for data collection in RSNs. Limited battery power poses a severe challenge for data collection in large-scale RSNs. If not dealt with satisfactorily, it could negatively affect the data-collecting performance in the RSN and compromise its function. We design a suite of optimal and heuristic algorithms to solve the BC-DCR. BC-DCR is equivalent to a graph-theoretical problem called the budget-constrained traveling salesman problem. Via extensive simulations using real measurements of robot battery power and mobility models, we show that our algorithms outperform the existing work by collecting more data packets and achieving a more extended network lifetime of the RSN. In this paper, we focus on the robot's mobility energy during the data-collecting process and haven't considered its receiving energy consumption. Integrating this energy consumption with existing mobility energy to achieve a holistic energy-efficient data-collecting framework for the RSN is a challenging new problem.

References

[1] Aaa battery. https://en.wikipedia.org/wiki/AAA_battery.

[2] Ibm cplex optimizer. https://www.ibm.com/analytics/cplex-optimizer.

[3] Miller–tucker–zemlin (mtz) subtour elimination constraint. https://how-to.aimms.com/Articles/332/332-Miller-Tucker-Zemlin-formulation.html.

[4] A better approximation algorithm for the budget prize collecting tree problem. *Operations Research Letters*, 32(4):316–319, 2004.

[5] L. Chen, W. Wang, H. Huang, and S. Lin. On time-constrained data harvesting in wireless sensor networks: Approximation algorithm design. *IEEE/ACM Transactions on Networking*, 24(5):3123–3135, 2016.

[6] Thomas Corman, Charles Leiserson, Ronald Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT Press, 2009.

[7] J. R. Current and D. A. Schilling. The covering salesman problem. *Transportation science*, 23:208–213, 1989.

[8] M. D Francesco, S. K. Das, and G. Anastasi. Data collection in wireless sensor networks with mobile elements: A survey. 8(1), 2011.

[9] L. C. Garaffa, M. Basso, A. A. Konzen, and E. P. de Freitas. Reinforcement learning for mobile robotics exploration: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 34(8):3796–3810, 2023.

[10] P. Ghosh, A. Gasparri, J. Jin, and B. Krishnamachari. *Robotic Wireless Sensor Networks*, page 545–595. Springer International Publishing, 2019.

[11] S. Guo, C. Wang, and Y. Yang. Joint mobile data gathering and energy provisioning in wireless rechargeable sensor networks. *IEEE Transactions on Mobile Computing*, 13(12):2836–2852, 2014.

[12] G. Gutin and A. Punnen, editors. *The Traveling Salesman Problem and its Variation*. Springer Science+Business Media, 2007.

[13] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Proc. of HICSS 2000*.

[14] J. Hua, L. Zeng, G. Li, and Z. Ju. Learning for a robot: Deep reinforcement learning, imitation learning, transfer learning. *Sensors*, 21(4), 2021.

[15] H. Huang, A. V. Savkin, M. Ding, and C. Huang. Mobile robots in wireless sensor networks: A survey on tasks. *Computer Networks*, 148:1–19, 2019.

[16] H. Huang, A. V. Savkin, M. Ding, and C. Huang. Mobile robots in wireless sensor networks: A survey on tasks. *Computer Networks*, 148:1–19, 2019.

[17] D. Kandris, C. Nakas, D. Vomvas, and G. Koulouras. Applications of wireless sensor networks: An up-to-date survey. *Applied System Innovation*, 3(1), 2020.

[18] J. Kober, J. A. Bagnell, and J. Peters. Reinforcement learning in robotics: A survey. 32(11), 2013.

[19] Jun Luo and J.-P. Hubaux. Joint mobility and routing for lifetime elongation in wireless sensor networks. In *IEEE Infocom*, pages 1735–1746, 2005.

[20] M. Zukerman M. N. Halgamuge, K. Ramamohanarao, and H. L. Vu. An estimation of sensor energy consumption. *Progress In Electromagnetics Research B*, 12:259–295, 2009.

[21] M. Ma, Y. Yang, and M. Zhao. Tour planning for mobile data-gathering mechanisms in wireless sensor networks. *IEEE Transactions on Vehicular Technology*, 62(4):1472–1483, 2013.

[22] D. McNulty, A. Hennessy, M. Li, E. Armstrong, and K. M. Ryan. A review of li-ion batteries for autonomous mobile robots: Perspectives and outlook for the future. *Journal of Power Sources*, 545:231943, 2022.

[23] A. Paul, D. Freund, A. Ferber, D. B. Shmoys, and D. P. Williamson. Prize-collecting tsp with a budget constraint. In *25th Annual European Symposium on Algorithms (ESA 2017)*.

[24] P. Rawat, K. Singh, H. Chaouchi, and J. M. Bonnin. Wireless sensor networks: a survey on recent developments and potential synergies. *The Journal of Supercomputing*, 68:1–48, 2014.

[25] J. Ruiz, C. Gonzalez, Y. Chen, and B. Tang. Prize-collecting traveling salesman problem: a reinforcement learning approach. In *Proc. of IEEE ICC*, 2023.

[26] H. Salarian, K. W. Chin, and F. Naghdy. An energy-efficient mobile-sink path selection strategy for wireless sensor networks. *IEEE Transactions on Vehicular Technology*, 63(5):2407–2419, 2014.

[27] Padmini R. Sokkappa. The cost-constrained traveling salesman problem, 1991. Ph.D. Thesis, Stanford University.

[28] C. Wang, S. Guo, and Y. Yang. An optimization framework for mobile data collection in energy-harvesting wireless sensor networks. *IEEE Transactions on Mobile Computing*, 15(12):2969–2986, 2016.

[29] Y.-C. Wang and K-C. Chen. Efficient path planning for a mobile sink to reliably gather data from sensors with diverse sensing rates and limited buffers. *IEEE Transactions on Mobile Computing*, 18(7):1527–1540, 2019.

[30] X. Xiao and W. L. Whittaker. Energy considerations for wheeled mobile robots operating on a single battery discharge. Technical Report CMU-RI-TR-14-16, Pittsburgh, PA, August 2014.