Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Toward an ILP Approach of Temporal Coverage in Sensor Networks

Mohammed Salem MOUMNI

moumni.mohammed@edu.univ-oran1.dz

University of Ahmed Ben Bella Oran1

Lakhdar LOUKIL

University of Ahmed Ben Bella Oran1

Houssam-Eddine ZAHAF

Nantes Université, Ecole Centrale Nantes

Pierre-Emmanuel HLADIK

Nantes Université, Ecole Centrale Nantes

Research Article

Keywords: Sensor networks, Temporal coverage, Energy harvesting, Integer Linear Programming, Sensor scheduling

Posted Date: June 24th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9596709/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Toward an ILP Approach of Temporal Coverage in Sensor Networks

Mohammed Salem MOUMNI[1*], Lakhdar LOUKIL[1], Houssam-Eddine ZAHAF[2], Pierre-Emmanuel HLADIK[2]

[1*] Computer Science Department, LAPECI, Ahmed Ben Bella University Oran 1, Oran, Algeria .
[2] Nantes Université, École Centrale Nantes, CNRS, INRIA, LS2N, UMR 6004, F-44000, Nantes, France .

*Corresponding author(s). E-mail(s): moumni.mohammed@edu.univ-oran1.dz;
Contributing authors: loukil.lakhdar@univ-oran1.dz; houssameddine.zahaf@univ-nantes.fr; pierre-emmanuel.hladik@ls2n.fr;

Abstract

Energy has long been a critical concern in the deployment of outdoor monitoring systems based on sensor networks, particularly when sensors are difficult to access. Energy-neutral sensors offer a promising solution, but they introduce additional challenges, notably the need to maintain temporal coverage in the presence of energy shortages.

This work addresses the problem of ensuring temporal continuity in monitoring by orchestrating the activation and charging cycles of multiple sensors to guarantee continuous temporal and spatial coverage. We investigate two distinct approaches based on Integer Linear Programming (ILP). In the first approach, once a sensor is activated, it continuously monitors until its battery level reaches a predefined energy threshold. In the second approach, the monitoring and charging phases of a sensor may be interrupted. These approaches present different trade-offs in terms of sensor-side implementation complexity, solving time, and solution optimality. We evaluate their performance through an extensive set of synthetic experiments.

Keywords: Sensor networks, Temporal coverage, Energy harvesting, Integer Linear Programming, Sensor scheduling

1 Introduction

Sensor networks play a crucial role in a wide range of applications, including environmental monitoring, disaster response, military surveillance, and smart city deployments [1, 2]. These networks consist of sensors deployed to monitor specific Points of Interest (POIs) and collect data that support decision-making processes. However, ensuring uninterrupted sensor operation remains a major challenge, as most sensors rely on batteries with limited energy capacity [3]. Consequently, effective energy management is essential to sustain long-term network operation.

Recent advances have introduced energy harvesting techniques that exploit renewable energy sources such as solar radiation, mechanical vibrations, and radio-frequency signals [3, 4] to power low-end devices. While these techniques extend sensor lifetime and enable energy-neutral operation, they do not eliminate energy constraints entirely. Fluctuations in harvested energy and mismatches between energy availability and workload demand make it difficult to continuously satisfy application requirements. As a result, achieving an efficient balance between energy consumption and energy harvesting remains a significant challenge, motivating the need for optimized energy management strategies for sustainable sensor network operation [1].

To ensure the reliability of monitoring systems, two fundamental constraints must be satisfied: spatial and temporal coverage. Spatial constraints require that every POI is covered by at least one sensor, while temporal constraints ensure that each POI is continuously covered over time [5]. Satisfying both constraints simultaneously under energy limitations is a key challenge in the design and operation of energy-constrained and energy-neutral sensor networks, that we address in this paper.

Ensuring continuous coverage of Points of Interest (POIs) while addressing the energy and spatial (coverage) constraints of individual sensors requires particular attention. Each sensor operates in two distinct states: an *active state*, during which it monitors a subset of the POIs within its sensing range and consumes energy from its battery, and a *passive state*, during which it harvests energy to replenish its battery in preparation for future activation [2]. Even during the active state, a sensor may continue to harvest energy. Depending on the balance between harvested and consumed energy, two scenarios arise. In the first scenario, the sensor harvests more energy than it consumes per time unit. Under such conditions, the monitoring process is not constrained by energy availability, allowing the sensor to operate continuously [1]. In the second scenario, the sensor consumes more energy than it can harvest per time unit when active, resulting in a limited operating period. After this period, the sensor must enter a passive state to recharge. This work focuses on the second scenario, in which energy availability constrains sensor operation, as opposed to the first scenario, which is unconstrained by energy limitations and can be addressed by considering only spatial coverage constraints.

Balancing energy and coverage constraints presents significant challenges, particularly under limited energy availability. Passive sensors cannot contribute to coverage, while active sensors deplete energy faster than it can be replenished. The central problem lies in determining how to orchestrate the active and passive states of different sensors to satisfy both spatial and temporal constraints. This requires collaborative scheduling of the sensors' states, ensuring compliance with local sensor constraints

and global monitoring constraints. This work focuses on developing an effective sensor scheduling mechanism that guarantees uninterrupted coverage of all POIs while efficiently managing the sensors' energy.

Contributions

We address these challenges by proposing two distinct approaches. The first, called the non-preemptive approach (NPA), enforces strict switching rules based on battery levels. Sensors switch from the active to the passive state when required, but switch from the passive to the active state only when fully charged. The second approach, called the full-preemptive approach (FPA), introduces flexibility by allowing state transitions regardless of battery levels. However, if the battery level is below the minimum threshold, the sensor cannot be switched to the active state.

Our contributions include the development of an Integer Linear Programming (ILP) formulation to determine feasible schedules and an analysis of the solution feasibility of these two approaches. The ILP produces a coordinated schedule for transitions between active and passive states for each sensor. One of the key contributions of this work is demonstrating that the finite schedule produced is valid over the system's (potentially infinite) lifetime. We analyze the performance of the NPA and FPA approaches through a large set of synthetic experiments, evaluating key metrics: feasibility (the ability to find a schedule that meets all timing constraints), solving time (the time required to find a feasible schedule), and the number of switches from active to passive per time unit.

Paper organization

The remainder of this paper is organized as follows. In the next section, we present the design of energy-harvesting sensors. Section 3 formally defines the system model, including the constraints and assumptions underlying our models. Section 4 establishes the temporal and spatial feasibility properties that form the theoretical foundation of the approach. Section 5 presents the ILP formulation for the non-preemptive and fully-preemptive models. Section 6 discusses the experimental evaluation of the proposed approach, highlighting its effectiveness under various scenarios. In Section 7, we review related work on sensor scheduling, energy harvesting, and network coverage. Finally, Section 8 concludes the paper with a summary of findings and potential avenues for future research.

2 Energy-Harvesting Sensor Design

A typical design of our target sensors is shown in Figure 1. Each sensor consists of multiple components that facilitate energy harvesting and storage. The first component is the *Transducer*, which converts ambient environmental energy into usable electrical energy. These transducers can include piezoelectric devices (vibrations), RF (Radio Frequency), electromagnetic systems, or others [6]. The amount of energy harvested by the transducers depends on the energy source and may vary due to fluctuations. For instance, solar transducers efficiently harvest energy from sunlight but are prone

to significant fluctuations, whereas some energy sources, such as RF, are more stable [7]. This study focuses on these latter types of transducers and energy sources, enabling the assumption that a minimum amount of energy can always be extracted from the environment. The second component is the *Rectifier*, which optimizes energy extraction from the transducer. The *Regulator* ensures a stable voltage for the system, often employing DC-DC converters [8]. The harvested energy is then used to recharge a battery, which powers a compute unit connected to various IOs[1] and communication modules [9].

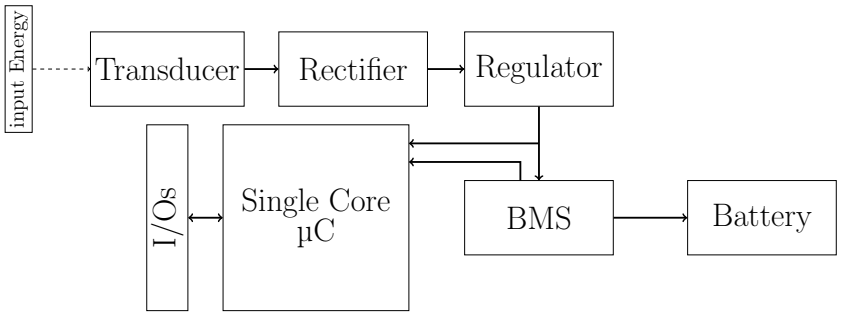


Fig. 1 Typical EH system design

The charging and discharging profiles of energy storage systems depend on the specific technology used. For batteries, the charging profile is mainly determined by the design of the Battery Management System (BMS) [9]. In devices such as those powered by lithium-ion batteries, a common BMS design utilizes a CC-CV (Constant Current–Constant Voltage) charging profile [8]. In this method, the BMS provides a constant current while monitoring the battery's voltage. Under this constant current, regulated by DC-DC converters and other components, the voltage rises *linearly* [7]. As the battery nears full charge, when the voltage reaches a predefined threshold, the current is gradually reduced to protect the battery, resulting in a slower charging rate [8].

A typical charging profile can be divided into two phases: a linear phase (up to nearly full charge) and a nonlinear phase as the battery reaches full capacity [9]. The nonlinear phase often requires substantial charging time for minimal additional energy gain. Consequently, the optimal operational point is typically found in the initial linear phase of charging.

In this paper, we define minimum $e_{\min}$ and maximum $e_{\max}$ energy levels that allow operation within the linear portion of the charging profile for a given battery (sensor). The nonlinear CC-CV phase (near full charge, where current tapers) is deliberately excluded from our operating range by setting $e_{\max}$ below the saturation threshold. Within the chosen $[e_{\min}, e_{\max}]$ window the charging profile is linear by the physics of CC-CV operation. The contributions of this paper, including the proposed Integer Linear Programs (ILPs), proofs of correctness, and extensions to infinite systems, are designed to accommodate this assumption. The charging profile is linear by the physics

[1] Input/Output

of CC-CV operation. However, these contributions could potentially be adapted to other charging profiles.

3 System Model

In this section, we present the different assumptions made in this paper, along with the symbols and notations used throughout the paper.

We consider a set of m Points of Interest (POIs), denoted as $\mathcal{P} = \{\mathsf{Poi_1}, \mathsf{Poi_2}, \cdots, \mathsf{Poi_m}\}$. Each Point of Interest $\mathsf{Poi_i}$ represents a subset of the monitored system. At each time instant t, $\mathsf{Poi_i}$ must be monitored by at least one sensor.

We also consider a set of sensors, $\mathcal{S} = \{\mathsf{S_1}, \mathsf{S_2}, \cdots, \mathsf{S_n}\}$. Each sensor $\mathsf{S_j}$ is associated with a coverage set $\mathsf{COV}(\mathsf{S_j}) = \{\mathsf{Poi_1}, \mathsf{Poi_2}, \cdots\}$, which denotes the set of POIs that the sensor can monitor.

Throughout this paper, we assume static sensor deployment and fixed coverage sets $\mathsf{COV}(\mathsf{S_j})$.

Each sensor $\mathsf{S_j}$ can operate in one of two states:

- **Active State**: The sensor monitors its POIs. In this state the sensor consumes and harvests energy.
- **Passive State**: The sensor is not monitoring and exclusively harvests energy to recharge its battery.

Each sensor $\mathsf{S_j}$ is further characterized by the following parameters:

- $\mathsf{B}(\mathsf{S_j})$: The battery capacity, representing the total energy the sensor can store.
- $c(\mathsf{S_j})$: The power dissipated by sensor $\mathsf{S_j}$ per unit of time when active.
- $h(\mathsf{S_j})$: The power harvested by sensor $\mathsf{S_j}$ per unit of time.

As stated in the introduction, we are interested only in systems that are constrained by energy availability, that is: $c(\mathsf{S_j}) > h(\mathsf{S_j})$.

The energy consumption (resp. harvested energy) of sensor $\mathsf{S_j}$ within the time interval $[t_1, t_2]$, denoted by ${}^{\mathsf{t1}}_{\mathsf{t2}}\mathsf{E}^{C}(\mathsf{S_j})$ (resp. ${}^{\mathsf{t1}}_{\mathsf{t2}}\mathsf{E}^{H}(\mathsf{S_j})$), is the integral of its power consumption (resp. harvested energy) over that period. This is expressed as:

$$ {}^{\mathsf{t1}}_{\mathsf{t2}}\mathsf{E}^{C}(\mathsf{S_j}) = \int_{t_1}^{t_2} c(\mathsf{S_j})\, dt $$

For the sake of simplicity, and without loss of generality, we assume that it can be computed as:

$$ {}^{\mathsf{t1}}_{\mathsf{t2}}\mathsf{E}^{C}(\mathsf{S_j}) = (t_2 - t_1)\ c(\mathsf{S_j}) $$

Similarly we define the energy harvested in the interval $[t_1, t_2]$ as:

$$ {}^{\mathsf{t1}}_{\mathsf{t2}}\mathsf{E}^{H}(\mathsf{S_j}) = (t_2 - t_1)\ h(\mathsf{S_j}) $$

We assume that the energy harvesting rate $h(\mathsf{S_j})$ is *constant* over time. This assumption is appropriate for stable energy sources such as RF harvesting, which this

work specifically targets (see Section 2). Extension to systems with time-varying or stochastic harvesting rates is identified as future work in Section 8.

Using these equations, we assume that the charging profile is linear, starting from 0 to the maximum capacity. This assumption does not result in any loss of generality, as the minimum and maximum battery levels that define the charging profile can be shifted by the minimum value.

We denote by $\mathsf{e}(\mathsf{S_i}, t)$ the energy level in the battery of sensor $\mathsf{S_i}$ at time instant t.

4 Temporal and Spatial Feasibility Properties

In this section, we introduce a set of fundamental properties required for formulating the ILP model. Consider a schedule that specifies, for each sensor, its activation and charging states over the interval $[0, H)$, where H denotes the hyperperiod (to be computed). This schedule is intended to be executed cyclically, such that the same pattern is repeated at every hyperperiod. Moreover, the schedule must satisfy the coverage constraints at every time instant.

To ensure that such a schedule is both feasible and sustainable, we establish two key requirements:

1. **(prop1)** *Validity*: the energy dynamics of each sensor must remain feasible throughout the hyperperiod, ensuring that no complete energy depletion occurs at any time, causing energy shortage.
2. **(prop2)** *Replayability*: the schedule must be energy-neutral over the hyperperiod, allowing it to be repeated indefinitely without causing cumulative energy deficits.

Definition 1 (Sensor Activation Interval) For each sensor i and each activation index j, we define an activation interval

$$I_{i,j} = [s_{i,j},\ s_{i,j} + p_{i,j} + a_{i,j}),$$

where $s_{i,j}$ denotes the start time of the j-th activation cycle, $p_{i,j} \geq 0$ is the duration of the passive (charging) state, and $a_{i,j} > 0$ is the duration of the active (monitoring) state. During $I_{i,j}$, sensor i remains passive on the interval $[s_{i,j},\ s_{i,j}+p_{i,j})$ and active on the interval $[s_{i,j} + p_{i,j},\ s_{i,j} + p_{i,j} + a_{i,j})$.

We denote by $\mathsf{e}(\mathsf{S_i}, s_{i,j})$ the energy level of sensor i at the beginning of the activation interval $I_{i,j}$.

Definition 2 (Complete Schedule) A complete schedule for sensor i over a hyperperiod of length H is a finite set of activation intervals

$$\mathcal{I}_i = \{I_{i,1}, I_{i,2}, \ldots, I_{i,n_i}\}$$

such that these intervals form a partition of the time domain $[0, H)$, that is,

$$[0, H) = \bigcup_{j=1}^{n_i} I_{i,j} \quad \text{and} \quad I_{i,j} \cap I_{i,k} = \emptyset \ \ \forall j \neq k.$$

Therefore, the activation intervals of sensor i are non-overlapping, contiguous, and ordered in time. From the system model, we derive several initial properties. In particular, we recall the assumption that $c(\mathsf{S_j}) > h(\mathsf{S_j})$, implying that the battery level decreases during the active state.

Lemma 1 (Maximum Active Duration) For a sensor $\mathsf{S_i}$, the duration of any active period a_i must satisfy

$$a_i \leq \frac{\mathsf{B}(\mathsf{S_i})}{c(\mathsf{S_i}) - h(\mathsf{S_i})}. \tag{1}$$

Proof The result follows directly from the assumed linear and constant energy consumption and harvesting rates. During an active period, sensor $\mathsf{S_i}$ consumes energy at rate $c(\mathsf{S_i})$ while harvesting energy at rate $h(\mathsf{S_i})$, with $c(\mathsf{S_i}) > h(\mathsf{S_i})$. Consequently, the battery is depleted at a constant net rate

$$c(\mathsf{S_i}) - h(\mathsf{S_i}).$$

The longest possible active period occurs when the sensor starts in the active state with a fully charged battery of capacity $\mathsf{B}(\mathsf{S_i})$. Dividing the available energy by the net depletion rate yields the maximum achievable active duration, leading to

$$a_i = \frac{\mathsf{B}(\mathsf{S_i})}{c(\mathsf{S_i}) - h(\mathsf{S_i})}.$$

Therefore, any feasible active period must satisfy $a_i \leq \frac{\mathsf{B}(\mathsf{S_i})}{c(\mathsf{S_i}) - h(\mathsf{S_i})}$. □

The maximum length of active intervals, as given by Lemma 1, is incorporated into the ILP formulation to over-constrain the solver, which can help reduce the solving time.

We now define what constitutes a valid complete schedule. To illustrate the notion of validity and replayability, we first provide an example.

Example 1 We consider a single sensor $\mathsf{S_1}$ with the following parameters:

$$\mathsf{B}(\mathsf{S_1}) = 100, \quad c(\mathsf{S_1}) = 5, \quad h(\mathsf{S_1}) = 1.$$

At time $t = 0$, let assume the battery level is $\mathsf{e}(\mathsf{S_1}, 0) = 80$.

(A) Valid sequence

Consider the following sequence of activation intervals:

- Interval 1: passive for 5 time units, then active for 10.
- Interval 2: passive for 4 time units, then active for 10.

The battery level at the beginning of interval 1 is $E^0 = 80$. We verify the validity property (Prop. 1) by checking that the battery level never becomes negative:

$$\begin{aligned}
\mathsf{e}(\mathsf{S_1}, 0) &= 80, \\
\mathsf{e}(\mathsf{S_1}, 15) &= \mathsf{e}(\mathsf{S_1}, 0) + 5 \cdot h(\mathsf{S_1}) - 10 \cdot (c(\mathsf{S_1}) - h(\mathsf{S_1})) = 45, \\
\mathsf{e}(\mathsf{S_1}, 29) &= \mathsf{e}(\mathsf{S_1}, 15) + 4 \cdot h(\mathsf{S_1}) - 10 \cdot (c(\mathsf{S_1}) - h(\mathsf{S_1})) = 9.
\end{aligned}$$

Since the energy evolution is linear, it is sufficient to check the battery level at the end of each activation interval. As $\mathsf{e}(\mathsf{S_1}, 15) \geq 0$ and $\mathsf{e}(\mathsf{S_1}, 29) \geq 0$, the battery never depletes, and the sequence satisfies the validity property.

(B) Invalid sequence
Consider instead the following sequence:

- Interval 1: passive for 2 time units, then active for 15.
- Interval 2: passive for 3 time units, then active for 12.

Again, assume $\mathsf{e}(\mathsf{S_1}, 0) = 80$.

$$\begin{aligned}\mathsf{e}(\mathsf{S_1}, 0) &= 80,\\ \mathsf{e}(\mathsf{S_1}, 17) &= \mathsf{e}(\mathsf{S_1}, 0) + 2 \cdot h(\mathsf{S_1}) - 15 \cdot (c(\mathsf{S_1}) - h(\mathsf{S_1})) = 22,\\ \mathsf{e}(\mathsf{S_1}, 32) &= \mathsf{e}(\mathsf{S_1}, 17) + 3 \cdot h(\mathsf{S_1}) - 12 \cdot (c(\mathsf{S_1}) - h(\mathsf{S_1})) = -23.\end{aligned}$$

This sequence leads to an energy shortage during the second activation interval and must therefore be discarded, as it violates the validity property.

Finally, note that the first sequence, although valid over a single hyperperiod, cannot be replayed indefinitely. At the end of the sequence, the battery level is 9, which is strictly lower than the initial level of 80. Repeating the same sequence would eventually violate the validity property. Consequently, the replayability property (Prop. 2), which requires energy neutrality over repeated executions, is not satisfied.

Definition 3 (Interval Completion Time) Let $f_{i,k}$ denote the end time of the execution of interval $I_{i,k}$ in the complete schedule

$$\mathcal{I}_i = \{I_{i,1}, I_{i,2}, \ldots, I_{i,k}, \ldots, I_{i,n_i}\}.$$

$f_{i,k}$ is given by

$$f_{i,k} = \sum_{j=1}^{k} \left(p_{i,j} + a_{i,j}\right).$$

Theorem 1 (Valid Complete Schedule) *Let $\mathcal{I}_i$ be a complete schedule for sensor $\mathsf{S_i}$,*

$$\mathcal{I}_i = \{I_{i,1}, I_{i,2}, \ldots, I_{i,n_i}\}.$$

The schedule is valid if and only if the following conditions hold:

$$\begin{cases} (1)\ \forall k,\ 1 \le k \le n_i, \\ \qquad \mathsf{e}(\mathsf{S_i}, 0) + \displaystyle\sum_{j=1}^{k} \Big(p_{i,j} \cdot h(\mathsf{S_i}) - a_{i,j} \cdot (c(\mathsf{S_i}) - h(\mathsf{S_i}))\Big) \ge 0, \\ (2)\ \mathsf{e}(\mathsf{S_i}, H) \ge \mathsf{e}(\mathsf{S_i}, 0) \ge 0. \end{cases} \tag{2}$$

Proof Let $\mathsf{e}(\mathsf{S_i}, 0)$ denote the battery level of sensor $\mathsf{S_i}$ at the beginning of the first interval. Under the assumed linear harvesting and consumption model, the battery level at the end of each activation interval satisfies the following recursion:

$$\mathsf{e}(\mathsf{S_i}, f_{i,k}) = \mathsf{e}(\mathsf{S_i}, f_{i,k-1}) + p_{i,k} \cdot h(\mathsf{S_i}) - a_{i,k} \cdot \big(c(\mathsf{S_i}) - h(\mathsf{S_i})\big), \qquad \forall k \ge 1,$$

with $f_{i,0} = 0$. Unrolling this recursion yields, for every $k = 1, \ldots, n_i$,

$$\mathsf{e}(\mathsf{S_i}, f_{i,k}) = \mathsf{e}(\mathsf{S_i}, 0) + \sum_{j=1}^{k} \Big(p_{i,j} \cdot h(\mathsf{S_i}) - a_{i,j} \cdot (c(\mathsf{S_i}) - h(\mathsf{S_i}))\Big).$$

Necessity: Assume that the complete schedule $\mathcal{I}_i$ is valid. By definition, the battery level of sensor $\mathsf{S_i}$ never becomes negative at any time during the hyperperiod. Consider any interval $I_{i,k}$. Each interval consists of a passive subinterval, during which energy is accumulated, followed by an active subinterval, during which energy is depleted. Consequently, the minimum battery level within $I_{i,k}$ is reached at the end of the active subinterval, that is, at time $f_{i,k}$. Since the battery never depletes, we must have

$$\mathsf{e}(\mathsf{S_i}, f_{i,k}) \geq 0 \qquad \forall k = 1, \ldots, n_i.$$

Substituting the closed-form expression of $\mathsf{e}(\mathsf{S_i}, f_{i,k})$ directly yields condition (1).

Moreover, validity over the entire hyperperiod implies that the battery level at time H (which coincides with $\mathsf{e}(\mathsf{S_i}, f_{i,n_i})$) is nonnegative. Since the schedule is intended to be executed cyclically, the battery level at the end of the hyperperiod must not be smaller than the initial level, otherwise repeated executions would eventually lead to depletion. Therefore, validity together with replayability implies $\mathsf{e}(\mathsf{S_i}, H) \geq \mathsf{e}(\mathsf{S_i}, 0) \geq 0$, which is condition (2).

Sufficiency: Conversely, assume that conditions (1) and (2) hold. From the expression above, condition (1) implies that $\mathsf{e}(\mathsf{S_i}, f_{i,k}) \geq 0$ for all $k = 1, \ldots, n_i$, that is, the battery level is nonnegative at the end of every activation interval. As previously argued, within any interval $I_{i,k}$ the battery level first increases during the passive subinterval and then decreases during the active subinterval, so its minimum value is attained at time $f_{i,k}$. Hence, the battery level never falls below zero at any time during the hyperperiod, which establishes the validity property.

Condition (2) ensures that the battery level at the end of the hyperperiod is not lower than at its beginning, guaranteeing energy neutrality (or non-degradation) across hyperperiods. As a result, the schedule can be repeated indefinitely without causing a cumulative energy deficit.

Conditions (1) and (2) are therefore necessary and sufficient for the complete schedule $\mathcal{I}_i$ to be valid. □

However, verifying the validity of a schedule requires knowing the required initial battery level. In particular, the last interval must harvest enough energy so that the battery level at time H is at least the initial level, i.e., $\mathsf{e}(\mathsf{S_i}, H) \geq \mathsf{e}(\mathsf{S_i}, 0)$, in order to enable replayability.

Lemma 2 (Sufficient Initial Energy Level) Let $\mathcal{I}_i = \{I_{i,1}, \ldots, I_{i,n_i}\}$ be a complete schedule for sensor $\mathsf{S_i}$. A sufficient condition on the initial battery level is

$$\mathsf{e}(\mathsf{S_i}, 0) \;\geq\; \max\Big\{0,\; a_{i,1} \cdot \big(c(\mathsf{S_i}) - h(\mathsf{S_i})\big) - p_{i,1} \cdot h(\mathsf{S_i})\Big\}.$$

Proof During the first interval $I_{i,1}$, the battery level evolves as

$$\mathsf{e}(\mathsf{S_i}, f_{i,1}) = \mathsf{e}(\mathsf{S_i}, 0) + p_{i,1} \cdot h(\mathsf{S_i}) - a_{i,1} \cdot (c(\mathsf{S_i}) - h(\mathsf{S_i})).$$

For the interval $I_{i,1}$ to be valid, the battery level must never become negative. As argued previously, the minimum battery level during $I_{i,1}$ is attained at its end. Therefore, we require

$$\mathsf{e}(\mathsf{S_i}, f_{i,1}) \geq 0,$$

which implies

$$\mathsf{e}(\mathsf{S_i}, 0) \geq a_{i,1} \cdot (c(\mathsf{S_i}) - h(\mathsf{S_i})) - p_{i,1} \cdot h(\mathsf{S_i}).$$

In addition, battery levels are nonnegative by definition, so $\mathsf{e}(\mathsf{S_i}, 0) \geq 0$. Combining both conditions yields the sufficient lower bound

$$\mathsf{e}(\mathsf{S_i}, 0) \geq \max\Big\{0,\, a_{i,1} \cdot (c(\mathsf{S_i}) - h(\mathsf{S_i})) - p_{i,1} \cdot h(\mathsf{S_i})\Big\},$$

which completes the proof. □

In other words, the energy required at time zero must be accumulated during the last interval of the schedule, so that the battery level at the end of the hyperperiod is sufficient to restore the initial energy level for the next execution.

Lemma 3 (Necessary condition for temporal coverage) Let Poi_j be a point of interest covered by a set of sensors $\mathcal{S}(\mathsf{Poi}_j)$.

A necessary condition for Poi_j to be temporally covered is:

$$\sum_{\mathsf{S_i} \in \mathcal{S}(\mathsf{Poi}_j)} \frac{h(\mathsf{S_i})}{c(\mathsf{S_i})} \geq 1. \tag{3}$$

Proof We prove this in two steps.

Step 1: Maximal active fraction of a single sensor.

When starting from energy level equal to 0, we must have:

$$p_i h(\mathsf{S_i}) \geq a_i (c(\mathsf{S_i}) - h(\mathsf{S_i})).$$

Dividing both sides by $p_i + a_i$ gives the maximal fraction of time that the sensor can be active:

$$\frac{a_i}{p_i + a_i} \leq \frac{h(\mathsf{S_i})}{c(\mathsf{S_i})}.$$

Hence, under steady operation, the maximal active fraction denoted as u_i of $\mathsf{S_i}$ is

$$u_i = \frac{h(\mathsf{S_i})}{c(\mathsf{S_i})}.$$

Step 2: Aggregation over all sensors. Temporal coverage requires that at least one sensor covering Poi_j is active at all times. Therefore, the sum of the maximal active fractions of all sensors covering Poi_j must be at least 1, we get:

$$\sum_{\mathsf{S_i} \in \mathcal{S}(\mathsf{Poi}_j)} \frac{h(\mathsf{S_i})}{c(\mathsf{S_i})} \geq 1,$$

which is exactly condition (3).

This proves that the inequality is a necessary condition for temporal coverage. □

Feasibility pre-filter. Before invoking the ILP, we apply the following lightweight checks:

1. **Per-sensor feasibility (battery constraint).** For each sensor $\mathsf{S_i}$:

$$\mathsf{B}(\mathsf{S_i}) \geq c(\mathsf{S_i}) - h(\mathsf{S_i}).$$

Sensors that do not satisfy this condition cannot be active even for a single time unit and are removed.

2. **Per-POI energy condition (Lemma 3).** For each Poi_j:

$$\sum_{\mathsf{S_i} \in \mathcal{S}(\mathsf{Poi}_j)} \frac{h(\mathsf{S_i})}{c(\mathsf{S_i})} \geq 1.$$

This necessary test allows us to avoid running the ILP described in the next section on scenarios that are clearly unsolvable.

Lemma 4 (FPA approach dominates the NPA approach) For any arbitrary scenario, if the NPA approach is able to solve it, then the FPA approach is also able to solve it.

Proof The proof relies on the explicit ILP constraints of both formulations, which are introduced in Section 5. It is therefore deferred to Lemma 5 in that section. □

5 Integer Linear Programming Models for Sensor Scheduling

In this section, we present our ILP formulation for solving the problem of temporal coverage for a set of sensors and points of interest.

The goal is to ensure continuous coverage of all POIs while adhering to the following constraints:

- **Spatial Constraint**: Every POI must be covered by at least one active sensor.
- **Temporal Constraint**: Coverage must be uninterrupted across the network, even as sensors transition between active and passive states.
- **Energy Constraints**: The energy consumed by each sensor during its active phase must not exceed its available battery capacity, and sufficient energy must be harvested during the passive phase to sustain future active periods.

To address these requirements, we propose two complementary models:

- **Non-Preemptive Model**: Sensors switch states strictly based on their battery levels, transitioning to the active state only when the battery is fully charged.
- **Preemptive Model**: Sensors are allowed to switch states irrespective of battery levels, providing greater flexibility in scheduling.

Although the theoretical analysis presented in the previous sections is formulated in continuous time, the ILP model operates over discrete time. The discretization granularity is not fixed by the model and is left as a design choice to the user. A finer time granularity allows the discrete formulation to more closely approximate the continuous-time behavior, but results in a larger ILP with increased computational complexity. Conversely, a coarser granularity reduces the size of the optimization problem and improves scalability, at the cost of abstracting short-term dynamics.

Regardless of the chosen granularity, the ILP captures the essential structure of the problem: sensor state transitions occur at discrete decision instants, and energy harvesting and consumption are accumulated over each time unit. The selected time

resolution thus determines a trade-off between modeling accuracy and solving time, allowing the formulation to be adapted to the computational resources and accuracy requirements of the target application.

5.1 Non-Preemptive ILP Model

The non-preemptive ILP model enforces energy constraints by reformulating battery dynamics into time-based constraints on sensor activation and charging phases. Instead of explicitly modeling energy harvesting and consumption through additional variables and constraints, the model captures their effect implicitly by constraining the duration and ordering of passive and active intervals. As a result, energy feasibility is guaranteed by construction: a sensor consumes energy while active and is allowed to return to the active state only after spending sufficient time in the passive state to restore its battery level.

By abstracting energy dynamics into temporal constraints, this approach simplifies both the model and its deployment. In particular, it avoids tracking instantaneous battery levels and eliminates the need for explicit energy balance equations in the ILP. From an implementation perspective, the resulting schedules are easy to enforce on the sensors, as they rely solely on time-based state transitions. Moreover, the non-preemptive nature of the model limits the number of state switches, which simplifies synchronization and reduces runtime overhead.

This simplicity, however, comes at the cost of optimality. By restricting sensors to complete phases without interruption, the non-preemptive model reduces scheduling flexibility and may exclude feasible schedules that would be allowed under preemptive formulations.

Remark (Discretization and Rounding in NPA): In the NPA formulation, the passive and active durations (p_i, a_i) are integer multiples of the discretization step Δt, derived from $c(\mathsf{S_i})$, $h(\mathsf{S_i})$, and $\mathsf{B}(\mathsf{S_i})$ via Theorem 1. To avoid rounding effects, Δt is chosen such that $a_i \left(c(\mathsf{S_i}) - h(\mathsf{S_i})\right)/h(\mathsf{S_i})$ is an integer multiple of Δt for all $\mathsf{S_i} \in \mathcal{S}$.

According to Theorem 1, several conditions must be satisfied to ensure that a valid schedule can be executed repeatedly. Among the required parameters is the length of the interval defining the hyperperiod. In the proposed ILP, this length is a decision variable, searched within the range $[0, T]$, where T denotes an upper bound on the time horizon considered for identifying a suitable hyperperiod $\mathbf{h}$.

For clarity, we introduce below a set of notations used throughout the paper. We denote by $\mathbb{N}_X^*$ the set of integers from 1 to X.

The sensing capability of each sensor is determined by the binary constant $c_{i,p}$, where $c_{i,p} = 1$ if sensor $\mathsf{S_i}$ can monitor POI $\mathsf{Poi_p}$, and $c_{i,p} = 0$ otherwise.

Furthermore, we define $\mathbf{x}_{i,t} \in \{0, 1\}$ as a binary variable indicating the state of sensor $\mathsf{S_i}$ at time instant t. The sensor is active when $\mathbf{x}_{i,t} = 1$ and inactive when $\mathbf{x}_{i,t} = 0$. Our ILP generates, for every sensor and every time instant, a variable $\mathbf{x}_{i,t}$ representing its activation state. Through this variable, we are able to express the various problem constraints stated above. These variables represent the complete design space, while the others are transformations or computations derived directly from exploring these decision variables.

We denote by $\mathbf{r}_{i,t} \in \{0,1\}$ a binary variable representing the rising transition of sensor i at time t (i.e., the sensor becomes active at t).

We denote by $\mathbf{fi_{i,t}} \in \{0,1\}$ a binary variable representing a falling edge (i.e., a transition from the active to the inactive state) for sensor $\mathsf{S_i}$ at time t.

We denote by $t_{\text{offset},i} \in \mathbb{N}$ an integer variable representing the offset time for sensor i. The offset indicates the activation time of the sensor in the first hyperperiod interval. For example, if $t_{\text{offset},i} = 0$, then the sensor is activated at the start of the hyperperiod.

We denote by $t_{\text{slack},i} \in \mathbb{N}$ an integer variable representing the slack time for sensor i. It corresponds to the duration of the last inactive state of sensor $\mathsf{S_i}$. It can be computed as the difference between the system's hyperperiod and the time of the last transition from the inactive to the active state.

In the rest of this section, we will show how the different variables are computed from $\mathbf{x}_{i,t}$ and $\mathbf{h}$.

Objective Function

In this work, we aim to minimize the system's hyperperiod, as this systematically reduces the number of sensor transitions, which can then be organized in an orchestrated manner at runtime. This, in turn, improves the implementability of the solution. Therefore, the objective function is defined as follows:

$$\text{Minimize} \quad \mathbf{h}$$

The objective function can be set to 0 to stop at the first feasible solution.

While the primary objective of this work is to minimize the hyperperiod h, the objective function can be extended to a bi-objective formulation that simultaneously minimizes the hyperperiod length and the total number of mode switches:

$$\min \quad \alpha \cdot h \; + \; \beta \cdot \sum_{i \in N_S^*} \sum_{t \in N_T^*} r_{i,t}, \tag{4}$$

where $\alpha, \beta \geq 0$ are user-defined weights. The second term $\sum_{i,t} r_{i,t}$ counts the total number of sensor activations over the hyperperiod using the already-defined rising-edge variable $r_{i,t}$, which equals 1 when sensor $\mathsf{S_i}$ transitions from inactive to active at time t. By varying α and β, the user can trade off between minimizing the schedule length and reducing synchronization overhead. This extension introduces no new variables, as $r_{i,t}$ is already part of both formulations. Setting $\alpha = 1$ and $\beta = 0$ recovers the original single-objective formulation used throughout this paper.

Model Constraints

Each POI must be covered at all times within the hyperperiod:

$$t < h \Rightarrow \sum_{i \in \mathbb{N}_S^*} c_{i,p} \cdot x_{i,t} \geq 1, \quad \forall p \in \mathcal{P}, \forall t \in \mathbb{N}_T^*$$

This variable is defined over the entire time horizon while it must indicate the coverage from time instant 0 to the system's hyperperiod. Therefore, for all time

instants beyond the system's hyperperiod, we enforce this variable to be set to 0.

$$t \geq h \Rightarrow x_{i,t} = 0, \quad \forall i \in \mathbb{N}_S^*,\ \forall t \in \mathbb{N}_T^*.$$

The binary variable $r_{i,t}$ indicates a transition from inactive to active state of sensor $\mathsf{S_i}$ at time t. Specifically:

- $r_{i,t} = 1$ if the sensor i transitions from an inactive state ($x_{i,t-1} = 0$) to an active state ($x_{i,t} = 1$) at time t.
- $r_{i,t} = 0$ otherwise.

Thus,

$$\forall i \in \mathbb{N}_S^*,\ \forall t \in \mathbb{N}_T^*, \quad r_{i,t} = \begin{cases} 1 & \text{if } x_{i,t} - x_{i,t-1} = 1, \\ 0 & \text{otherwise.} \\ & \textit{with } r_{i,0} = x_{i,0}, \forall i \in \mathbb{N}_S^*. \end{cases}$$

This constraint is not linear but can be easily linearized using a classical approach for condition linearization. Modern solvers support the direct expression of such conditional constructs. For the sake of space, we do not include the explicit linearization here.

The binary variable $\mathbf{fi_{i,t}}$ indicates the reverse transition from active state to inactive state of a sensor i at time t. Specifically:

$$\forall i \in \mathbb{N}_S^*,\ \forall t \in \mathbb{N}_T^*, \quad \mathbf{fi_{i,t}} = \begin{cases} 1 & \text{if } x_{i,t-1} - x_{i,t} = 1, \\ 0 & \text{otherwise} \\ & \textit{with } \mathbf{fi_{i,0}} = 0, \forall i \in \mathbb{N}_S^*. \end{cases}$$

This can also directly be expressed in linear solvers similarly to the previous one.

Sensors are required to remain active for their designated duration. Once this period elapses, they should transition to a passive state:

$$\forall i \in \mathbb{N}_S^*, r_{i,t} = 1 \Rightarrow \mathbf{fi_{i,t'}} = 0, \quad \forall t' \in [t, \min(t + a_i(\mathsf{S_i}) - 1, T)]$$

Sensors must remain inactive for their specified period p_i after deactivation:

$$\forall i \in \mathbb{N}_S^*, \mathbf{fi_{i,t}} = 1 \Rightarrow r_{i,t'} = 0, \quad \forall t' \in [t, \min(t + p_i - 1, T)]$$

The offset time $t_{\text{first},i}$ for each sensor i can be computed as:

$$\forall i \in \mathbb{N}_S^*, t_{\text{offset},i} = \min\{t \in \mathbb{N}_T^* : x_{i,t} = 1\},$$

The slack time can be computed as well as:

$$t_{\text{slack},i} = \max\{t \in \mathbb{N}_T^* : x_{i,t} = 1\}, \quad \forall i \in \mathbb{N}_S^*$$

These two variables, $t_{\text{offset},i}$ and $t_{\text{slack},i}$, are used to determine whether a given sensor has accumulated sufficient energy to execute the full duration of its first activation (Lemma 3). Therefore, the offset and slack activation time must satisfy the periodicity of the sensor:

$$t_{\text{offset},i} + \mathbf{h} - t_{\text{slack},i} \geq p_i, \quad \forall i \in \mathbb{N}_S^*, \forall t \in \mathbb{N}_T^*$$

This constraint ensures that the elapsed time is at least equal to the required recharging period p_i.

Complete ILP Formulation (Non-Preemptive Approach)

$$\text{Minimize} \quad \mathbf{h}$$

Subject to

$$t < v_h \Rightarrow \sum_i c_{i,p} \cdot x_{i,t} \geq 1, \qquad \forall p, t \qquad (1)$$

$$t \geq v_h \Rightarrow x_{i,t} = 0, \qquad \forall i, t \qquad (2)$$

$$r_{i,t} = x_{i,t} - x_{i,t-1}, \qquad \forall i, t \qquad (3)$$

$$\mathbf{fi_{i,t}} = x_{i,t-1} - x_{i,t}, \qquad \forall i, t \qquad (4)$$

$$r_{i,t} = 1 \Rightarrow \mathbf{fi_{i,t'}} = 0, \quad \forall t' \in \mathcal{T}_a(i,t), \qquad \forall i \qquad (5)$$

$$\mathbf{fi_{i,t}} = 1 \Rightarrow r_{i,t'} = 0, \quad \forall t' \in \mathcal{T}_p(i,t), \qquad \forall i \qquad (6)$$

$$t_{\text{offset},i} + v_h - t_{\text{slack},i} \geq T_p(i), \qquad \forall i \qquad (7)$$

$$x_{i,t}, r_{i,t}, \mathbf{fi_{i,t}} \in \{0,1\}, \qquad \forall i, t \qquad (8)$$

5.2 Fully Preemptive Approach ILP Model

In contrast to the previous formulation, in this model, a period of active and passive states does not necessarily need to be fully executed. Therefore, the harvested and consumed energy are explicitly tracked, making the problem formulation and the design space to explore larger. We keep the same objective function.

Model Constraints

We introduce a continuous decision variable $\mathsf{e}(\mathsf{S_i}, t)$ representing the energy level of sensor i at time t (as defined in System model). According to the system constraints, this quantity must always remain non-negative. The energy evolution is modeled as:

$$\begin{aligned} \forall i \in \mathbb{N}_S^*, \forall t \in \mathbb{N}_T^1, t < h \Rightarrow \\ \mathsf{e}(\mathsf{S_i}, t) = & \mathsf{e}(\mathsf{S_i}, t-1) \\ & + h(\mathsf{S_i}) \cdot (1 - x_{i,t}) \\ & - (c(\mathsf{S_i}) - h(\mathsf{S_i})) \cdot x_{i,t} \end{aligned}$$

Of course, we do not track energy levels outside the hyperperiod, so we enforce them to remain constant:

$$t \geq h \Rightarrow \mathsf{e}(\mathsf{S_i}, t) = \mathsf{e}(\mathsf{S_i}, t-1), \forall i \in \mathbb{N}_S^*, \forall t \in \mathbb{N}_T^*.$$

The initial battery level $\mathsf{e}(\mathsf{S_i}, 0)$ is treated as a free decision variable by the ILP. The solver determines the minimum value that satisfies all energy constraints simultaneously. Before executing the first hyperperiod, each sensor's battery must be charged to at least the level computed by the ILP; in the most conservative case, initializing all batteries to full charge is always sufficient.

We must ensure that energy remains non-negative:

$$\mathsf{e}(\mathsf{S_i}, t) \geq 0, \quad \forall i \in \mathbb{N}_S^*, \forall t \in \mathbb{N}_T^*.$$

We must ensure that the energy available at the end is greater or equal to the one computed as required for the first activation, that is:

$$\mathsf{e}(\mathsf{S_i}, T) \geq \mathsf{e}(\mathsf{S_i}, 0), \quad \forall i \in \mathbb{N}_S^*, \forall t \in \mathbb{N}_T^*.$$

The accumulated energy must be roofed, that is

$$\mathsf{e}(\mathsf{S_i}, t) \leq \mathsf{B}(\mathsf{S_i}), \quad \forall i \in \mathbb{N}_S^*, \ \forall t \in \mathbb{N}_T^*.$$

These constraints ensure that the battery level never drops below zero, remains below the maximum capacity, and is sufficient to sustain periodic reactivation at the beginning of each hyperperiod.

Hardware Switching Latency: The current FPA formulation assumes instantaneous switching and does not explicitly account for hardware-level transition latency. To preserve model simplicity while enhancing realism, we incorporate latency-aware guard constraints directly into the formulation. Specifically, a latency $\lambda_i > 0$ (representing the settling time required after a state transition) is enforced by $r_{i,t} = 1 \Rightarrow x_{i,t+k} = 0$ for $k = 0, \ldots, \lambda_i - 1$, ensuring that a sensor becomes operational only after a fixed warm-up period. Similarly, deactivation delay is modeled by $\mathbf{fi_{i,t}} = 1 \Rightarrow x_{i,t+k} = 1$ for $k = 0, \ldots, \lambda_i - 1$, capturing the fact that a sensor may remain active for a short cool-down period after a deactivation command. These constraints capture the essential temporal effects of hardware transitions while preserving the overall simplicity of the formulation.

Complete ILP Formulation (Full-Preemptive Approach)

In the following formulation, indices and symbols are used as follows: $i \in \mathbb{N}_S^*$ denotes the sensor index, $t \in \mathbb{N}_T^*$ the discrete time instants, and $p \in \mathcal{P}$ the points of interest (POIs) to be covered.

$$\text{Minimize} \quad h$$

Subject to:

$$t < v_h \Rightarrow \sum_i c_{i,p} x_{i,t} \geq 1, \forall p, t \tag{1}$$

$$r_{i,t} = x_{i,t} - x_{i,t-1}, \quad \mathbf{fi_{i,t}} = x_{i,t-1} - x_{i,t}, \forall i, t \tag{2}$$

$$t < h \Rightarrow \mathsf{e}(\mathsf{S_i}, t) = \mathsf{e}(\mathsf{S_i}, t-1) + h(\mathsf{S_i})(1 - x_{i,t}) - (c(\mathsf{S_i}) - h(\mathsf{S_i}))x_{i,t}, \forall i, t \tag{3}$$

$$t \geq h \Rightarrow \mathsf{e}(\mathsf{S_i}, t) = \mathsf{e}(\mathsf{S_i}, t-1), \forall i, t \tag{4}$$

$$\mathsf{e}(\mathsf{S_i}, t) \geq 0, \quad \mathsf{e}(\mathsf{S_i}, t) \leq \mathsf{B}(\mathsf{S_i}), \forall i, t \tag{5}$$

$$\mathsf{e}(\mathsf{S_i}, T) \geq \mathsf{e}(\mathsf{S_i}, 0), \forall i \tag{6}$$

$$x_{i,t}, r_{i,t}, \mathbf{fi_{i,t}} \in \{0, 1\}, \forall i, t \tag{7}$$

Lemma 5 Any feasible schedule produced by the NPA formulation is also a feasible schedule under the FPA formulation. This establishes the dominance claim of Lemma 4.

Proof Let σ be an arbitrary complete schedule $\{I_{i,1}, I_{i,2}, \ldots, I_{i,n_i}\}$ for all sensors $\mathsf{S_i} \in \mathcal{S}$, such that σ satisfies all NPA constraints (1)–(8). We show that σ also satisfies all FPA constraints.

Shared constraints. Both formulations share constraints (1)–(4) and (8): the coverage constraint ensuring every POI $\mathsf{Poi_i} \in \mathcal{P}$ is monitored at all times $t < h$, and the definitions of $x_{i,t}$, $r_{i,t}$, and $\mathbf{fi_{i,t}}$. Since σ satisfies all NPA constraints, it satisfies these in particular, and therefore satisfies all shared FPA constraints.

Energy feasibility. The NPA formulation enforces energy feasibility implicitly through the duration constraints on passive and active intervals: sensor $\mathsf{S_i}$ returns to the active state only after completing a full passive duration p_i (sufficient to fully recharge), and remains active for exactly a_i time units (constraints (5)–(7)). By Theorem 1, these duration constraints are sufficient to guarantee that the complete schedule $\mathcal{I}_i$ is valid, that is:

$$\forall k,\ 1 \leq k \leq n_i : \quad \mathsf{e}(\mathsf{S_i}, H) \geq \mathsf{e}(\mathsf{S_i}, 0) \geq 0 \tag{5}$$

which are precisely the energy constraints required by the FPA formulation (constraints (6) and (7) of the FPA model). Hence σ satisfies all FPA energy constraints.

NPA is strictly more constrained than FPA. The NPA formulation imposes two additional constraints absent from FPA:

1. Once sensor $\mathsf{S_i}$ becomes active ($r_{i,t} = 1$), the active interval a_i must be completed without interruption (constraint (5) of NPA).
2. Sensor $\mathsf{S_i}$ may only become active again after completing the full passive duration p_i (constraint (6) of NPA).

The FPA formulation imposes neither of these two constraints. It requires only $e(\mathsf{S_i}, t) \geq 0$ before any activation transition. Since every NPA constraint is either shared with FPA or is an additional restriction on top of FPA, the set of schedules accepted by NPA is a strict subset of those accepted by FPA.

Since σ was an arbitrary NPA-feasible schedule, every complete schedule satisfying the NPA formulation also satisfies the FPA formulation, which confirms the dominance claim of Lemma 4. □

The contiguous-phase structure of NPA can create situations in which sensor i completes its active period before another sensor covering the same POI has begun its own, leaving a temporal coverage gap. This is the primary reason NPA exhibits lower feasibility than FPA under high inflation factors: overlapping activation intervals cannot be decomposed or rearranged without violating the non-preemptive structure. The k-coverage parameter (requiring at least k sensors to cover each POI simultaneously) can be increased to provide redundancy as a mitigation strategy. The contiguous-phase constraint is a deliberate design choice that reduces implementation complexity, as sensors require only a simple timer with no mid-phase interruption logic.

The discrete energy evolution equation in constraint (3) approximates a continuous-time process using binary state variables. We now quantify the approximation error introduced by this discretization.

Lemma 6 (Per-Step Discretization Error Bound) Consider the discrete-time energy update given by constraint (3) of the FPA formulation, where each time step has unit duration. Assume that in the continuous-time model, sensor $\mathsf{S_i}$ may be active for a fraction $\delta_{i,t} \in [0,1]$ of time slot t. The maximum approximation error between the true continuous energy variation and the discrete ILP update during a single time step is bounded by:

$$\max_{i,t} \left| \Delta e_{i,t}^{\text{true}} - \Delta e_{i,t}^{\text{ILP}} \right| \;\leq\; \frac{c(\mathsf{S_i})}{2}, \tag{6}$$

where the true continuous energy variation over one time step is:

$$\Delta e_{i,t}^{\text{true}} = h(\mathsf{S_i})(1-\delta_{i,t}) - \big(c(\mathsf{S_i}) - h(\mathsf{S_i})\big)\delta_{i,t}, \tag{7}$$

and the discrete ILP approximation is:

$$\Delta e_{i,t}^{\text{ILP}} = h(\mathsf{S_i})(1-x_{i,t}) - \big(c(\mathsf{S_i}) - h(\mathsf{S_i})\big)x_{i,t}, \tag{8}$$

which corresponds exactly to constraint (3) of the FPA model with $x_{i,t} \in \{0,1\}$.

Proof Let $\delta_{i,t} \in [0,1]$ denote the true continuous active fraction of time slot t for sensor $\mathsf{S_i}$.

Case 1: $x_{i,t} = 1$**.**

$$\begin{aligned} \epsilon_{i,t} &= \left| h(\mathsf{S_i})(1-\delta_{i,t}) - \big(c(\mathsf{S_i}) - h(\mathsf{S_i})\big)\delta_{i,t} + \big(c(\mathsf{S_i}) - h(\mathsf{S_i})\big) \right| \\ &= \left| c(\mathsf{S_i})(1-\delta_{i,t}) \right| = c(\mathsf{S_i})(1-\delta_{i,t}). \end{aligned} \tag{9}$$

Case 2: $x_{i,t} = 0$**.**

$$\begin{aligned} \epsilon_{i,t} &= \left| h(\mathsf{S_i})(1-\delta_{i,t}) - \big(c(\mathsf{S_i}) - h(\mathsf{S_i})\big)\delta_{i,t} - h(\mathsf{S_i}) \right| \\ &= \left| c(\mathsf{S_i})\,\delta_{i,t} \right| = c(\mathsf{S_i})\,\delta_{i,t}. \end{aligned} \tag{10}$$

Worst-case bound. In both cases the error is bounded by:

$$\epsilon_{i,t} \leq c(\mathsf{S_i}) \cdot \max\big(\delta_{i,t},\; 1-\delta_{i,t}\big). \tag{11}$$

The function $\max(\delta, 1-\delta)$ over $\delta \in [0,1]$ is maximised at $\delta = \frac{1}{2}$, giving:

$$\epsilon_{i,t} \leq \frac{c(\mathsf{S_i})}{2}. \tag{12}$$

This bound is tight: it is attained when $\delta_{i,t} = \frac{1}{2}$, i.e., when the sensor switches state exactly halfway through a time step. □

6 Experimental Results

In this work, we evaluate the performance of both the NPA and FPA approaches using synthetic task sets. We compare NPA against FPA based on three metrics. The first metric is the *feasibility rate within a time bound*, which represents the number of feasible solutions found by the solver within a given time window. The second metric is the *solving time*. The third metric is the *mode-changing ratio*, defined as the average number of sensors that switch on or off per time unit. This last metric reflects the synchronization effort required to implement the ILP results: the higher the synchronization, the less desirable the solution.

The fourth metric is the mean solving time as a function of the discretization step Δt, which quantifies how the granularity of the time grid affects solver effort for each formulation. The fifth metric is the ILP size, measured by the number of binary decision variables, as a function of Δt; this characterizes the computational complexity growth induced by finer discretization and explains the asymmetry in solver behavior between FPA and NPA.

6.1 Scenario Generation

In this work, we generate multiple scenarios. Each scenario takes as input a set of n sensors, m points of interest, and a total charging rate U. It also takes as input the maximum duration and the required k-coverage level. For each sensor, the scenario generation process computes a set of charging and discharging parameters, from which we derive the corresponding charging and discharging durations. To avoid generating excessively large periods that cannot be feasibly computed within the given time bound, we define a maximum period as an input parameter. Additionally, the scenario specification includes the minimum and maximum battery capacities.

First we generate the set of charging rates, using a uniform distribution inspired from UUnifast Algorithm described in Algorithm 1.

The UUniFast algorithm has been modified to produce a set of charging rates for sensors that collectively match a predefined total input value. The algorithm takes as input a total charging, a maximum duration and maximum sensor battery capacity. It iterates until finding a set of charging constraints that fits with the constraint that the charging time should not exceed the maximum duration.

The algorithm iterates over each sensor and generates a single random charging rate, and retrieves the latter from the total remaining total charging rate. This rate is then checked to ensure it satisfies both the storage capacity and maximum duration constraints. For each sensor, the maximum charging must not exceed a predefined input value, denoted as max_duration. To verify this, the charging rate is multiplied by the maximum duration and compared with the sensor's storage capacity. If this value exceeds the capacity, it means that the sensor would require more charging time than allowed by the duration constraint. In such cases, the charging rate is capped at the maximum possible value that satisfies the constraints.

This adjustment contrasts with the original UUniFast algorithm, which accepts all generated rates without such checks. Additionally, since our modification may occasionally produce negative charging rates, we include a validation step: if a negative

Algorithm 1: UUniFast Utilization Generator

Input: n_bT: number of tasks, u: total utilization, $maxCapa$: maximum capacity, $max_duration$: maximum duration
Output: List of utilizations

```
repeat
    valid ← true ;
    sumU ← u ;
    nextSumU ← 0 ;
    utilizations ← [ ] ;
    for i ← 1 to n − 1 do
        t ← randomDouble(0, 1) ;
        nextSumU ← sumU × t^(1/(s−i)) ;
        u_i ← sumU − nextSumU ;
        nu ← u_i ;
        if u_i × (max_duration/2) < maxCapa then
            nu ← maxCapa/(max_duration/2) ;
            nextSumU ← sumU − nu + u_i ;
            if nextSumU < 0 then
                nextSumU ← 0 ;
                valid ← false ;
                break ;
            u_i ← nu ;
        append u_i to utilizations ;
        sumU ← nextSumU ;
    append sumU to utilizations ;
until valid = true;
```

rate is generated, the entire utilization generation process is repeated, as such values are considered invalid.

Furthermore, the remaining parameters are generated using Algorithm 2.

Once the charging rates are generated, our generator, described in Algorithm 2, begins by creating the sensors. Each sensor's charging rate is selected from the generated rates, while its discharging rate is obtained by inflating the charging rate by a predefined input ratio. Next, the points of interest are generated. Finally, these points of interest are randomly assigned to sensors, ensuring that each point of interest is covered by at least kcov sensors.

While the synthetic scenario generation provides controlled and reproducible benchmarks across a wide range of energy conditions, it is worth noting that purely synthetic evaluation is standard practice in ILP-based scheduling research. The primary goal of such evaluation is to characterize algorithm behavior across a controlled parameter space — in our case, the inflation factor, the number of sensors, and the coverage requirements — rather than to replicate a specific deployment environment.

Algorithm 2: Generator Execution Procedure

Input: n, U, maxCapa, max_duration, m, infl, kcov
Output: Initialized lists `sensors` and `pois` with assigned k-coverage
$chargingRate \leftarrow$ UUniFast$(n, U, maxCapa, max_duration)$;

for $i \leftarrow 0$ ***to*** $n-1$ **do**
  $chargeRate \leftarrow chargingRate[i]$;
  $dischRate \leftarrow infl * chargeRate$;
  $C \leftarrow$ random_between$(minCapa, maxCapa)$;
  $s \leftarrow$ new Sensor$(i, chargeRate,\ dischRate, C)$;
  append s to $sensors$;

// Point of Interest (POI) creation ;
for $i \leftarrow 0$ ***to*** $numPois - 1$ **do**
  $p \leftarrow$ new PointOfInterest$(i,$ "POI" $+\ i)$;
  append p to $pois$;

// Assign k-coverage between sensors and POIs ;
assignKCoverage$(id, info, pois, sensors, kcov)$;

This allows for a systematic and fair comparison between NPA and FPA across conditions that would be difficult to control in a real deployment. Nevertheless, validation against real energy-harvesting traces — such as solar irradiance measurements or RF energy harvesting logs — would further strengthen the practical applicability of the results and is identified as an important direction for future work.

6.2 Results and Discussions

In this section, we present a detailed comparison between the FPA and the NPA approaches. Our goal is to evaluate their respective abilities to maintain feasibility, minimize solving time, and reduce the number of synchronization events under varying energy constraints.

The primary experiment consists of varying the *inflation factor* from 1 to 6. This factor serves as a control parameter that increases the difficulty of the scheduling problem by amplifying the gap between energy consumption and energy harvesting. When the inflation factor is small, the system benefits from abundant energy and flexible scheduling opportunities. Conversely, as the inflation factor grows, the system becomes more energy-limited, making it increasingly difficult to find feasible activation schedules.

In addition to the main inflation-factor sweep, we include a dedicated discretization sensitivity experiment in which the same physical scenarios are solved at four different time resolutions $\Delta t \in \{0.5, 1.0, 1.5, 2.0\}$. This controlled comparison isolates the effect of the discretization step from all other parameters and directly measures the integrality gap: the fraction of scenarios for which the continuous-time Lemma 3 condition holds but the discrete ILP fails to find a solution.

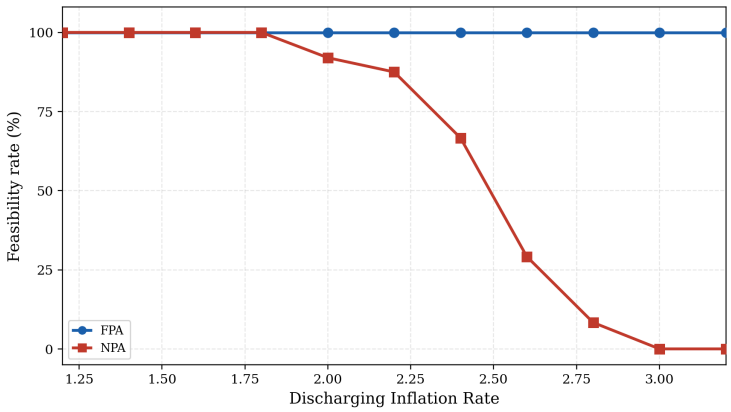


Fig. 2 Feasibility comparison between NPA and FPA.

Figure 2 shows the feasibility results obtained for both FPA and NPA. In this context, feasibility refers to the number of valid solutions found within the time limit of 8 minutes. All experiments were performed on a Dell Latitude 5490 laptop equipped with an Intel Core i7 processor and 16 GB of RAM, using the CPLEX optimization solver. We varied the inflation factor between 1.2 and 6. Each inflation rate has been tested 30 times.

As observed in Figure 2, FPA consistently finds feasible solutions for inflation factors ranging between 1.2 and 3.2. This robustness is primarily due to its ability to flexibly synchronize sensor activations by freely deciding when to activate or deactivate each node. This additional freedom allows FPA to continuously adjust to available energy conditions and avoid infeasibility, even when energy resources fluctuate.

In contrast, NPA exhibits a more constrained behavior. Because NPA enforces non-preemptive operation, each activation interval must be executed entirely once started, without interruption. As the inflation factor increases, charging phases become longer, leading to tighter temporal constraints and reduced flexibility. Consequently, NPA's feasible region shrinks faster than that of FPA. In many cases, overlapping activation intervals cannot be decomposed or rearranged, leading to infeasible configurations. This limitation explains the observed decline in schedulability as the inflation factor increases.

Figure 3 presents the same experiment but focuses exclusively on FPA to analyze its feasibility limits in detail. We can observe that FPA maintains a high level of feasibility across a wide range of inflation factors. Only when the inflation factor reaches values close to 5 does the number of feasible solutions begin to drop significantly. This behavior aligns with theoretical expectations, as this threshold corresponds to the point where energy consumption consistently exceeds the harvesting potential, making it impossible to meet all activation demands. Overall, these results highlight

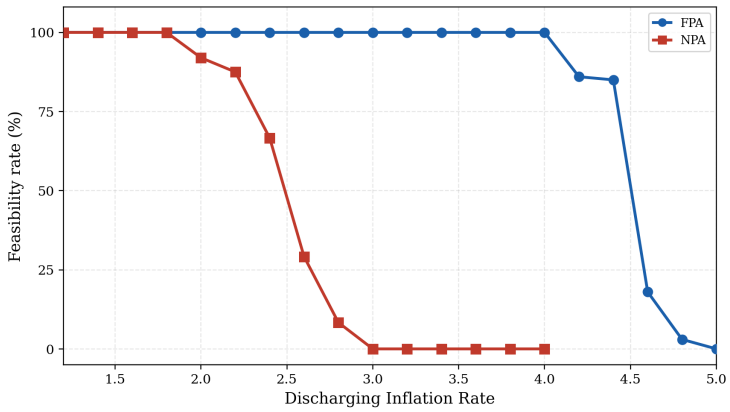


Fig. 3 Feasibility results for FPA.

the ability of FPA to sustain feasible operation in challenging energy environments, confirming its robustness compared to NPA.

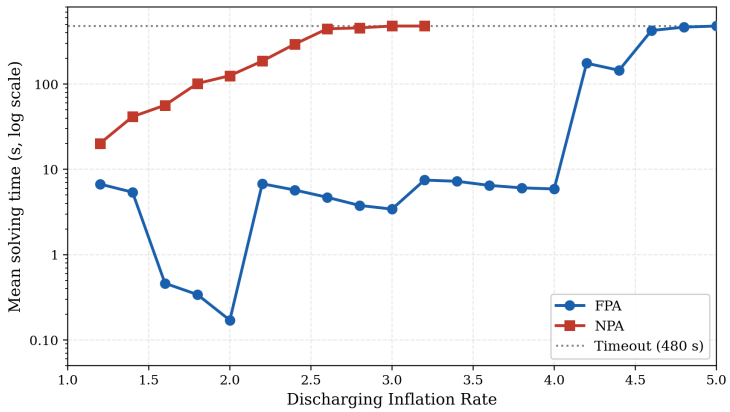


Fig. 4 Solving time comparison between NPA and FPA.

Figure 4 illustrates the average solving time on a log scale for both NPA and FPA, considering only instances where a feasible solution was found. As expected,

NPA requires significantly more computational effort. This is because NPA introduces additional scheduling constraints that increase the search space complexity, forcing the solver to explore a larger number of possibilities before concluding. Consequently, NPA's solving time increases rapidly with the inflation factor, reaching the upper time limit of 480 seconds in several cases.

In contrast, FPA exhibits much faster convergence. Its formulation allows a smoother search process since the problem remains more flexible and continuous. The solving time of FPA remains almost constant and low for most inflation factors, only increasing slightly as the problem becomes highly constrained. This makes FPA not only more robust in terms of feasibility but also far more practical for large-scale optimization scenarios where time is a limiting factor.

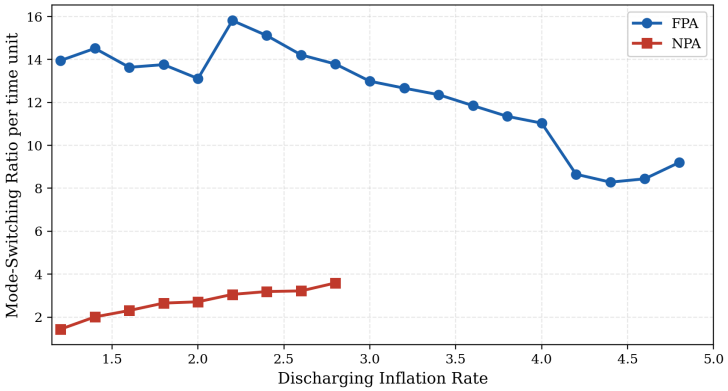


Fig. 5 Ratio of activations to minimum period per algorithm.

Finally, Figure 5 evaluates the quality of the generated schedules by analyzing the ratio of activations to the minimum period, which reflects the number of mode switches or synchronization events per time unit. This metric captures the practical cost of implementing a schedule: frequent mode switches increase system overhead and energy waste due to communication and control transitions.

The results clearly show that NPA maintains a very low number of synchronization events, producing smooth and stable activation patterns. In contrast, FPA exhibits a significantly higher number of mode changes per time unit, indicating more fragmented and discontinuous operation. This leads to higher implementation complexity in real systems.

Figure 6 shows the mean ILP solving time as a function of the discretization step Δt. For FPA, solving time increases steadily as Δt decreases, reflecting the growth of the ILP as the time grid becomes finer. The solver consistently finds a feasible solution

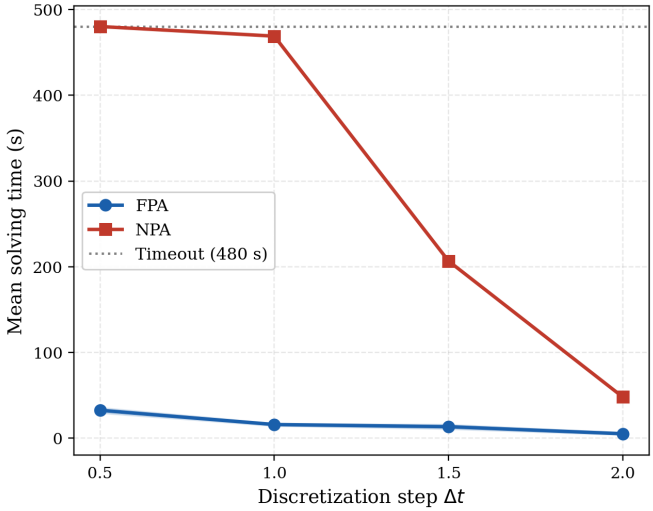


Fig. 6 Solving Time vs. Discretization Step Δt for NPA and FPA.

at all tested resolutions. For NPA, the behavior is fundamentally different: at coarse discretization, NPA solves within reasonable time, but as Δt decreases, solving time rises sharply until it reaches the timeout limit. The gap between the two curves widens as Δt decreases, confirming that fine discretization penalizes NPA far more severely than FPA.

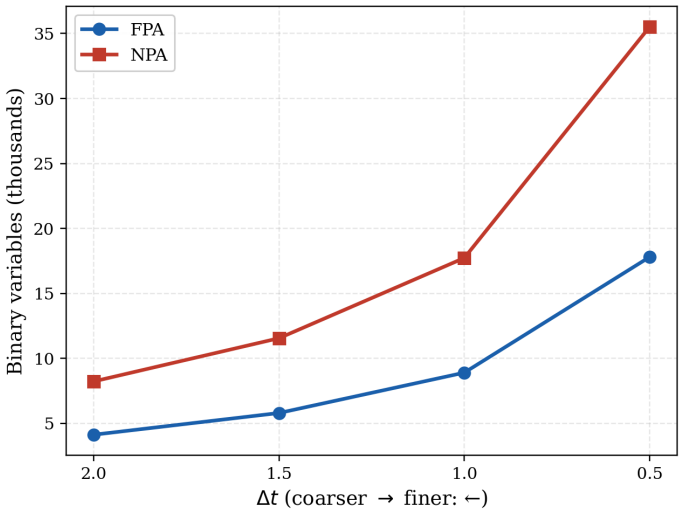


Fig. 7 ILP Size vs. Discretization Step Δt for NPA and FPA.

Figure 7 illustrates how the number of binary decision variables grows as Δt decreases for both formulations. Both curves follow the same inverse trend with Δt, since finer discretization produces more time steps and therefore more variables. NPA consistently requires approximately twice as many variables as FPA at every resolution, a structural consequence of its auxiliary start-of-block variables. Crucially, comparing this figure with the solving-time figure reveals that the two formulations experience the same proportional growth in ILP size, yet their solving times diverge drastically at fine resolutions. This confirms that the NPA runtime increase at small Δt is not caused by size growth alone, but by the difficulty of satisfying the contiguity constraint when the time grid is dense.

7 Related Work

The coverage problem has been widely investigated in the context of Wireless Sensor Networks (WSNs) and the Internet of Things (IoT). Existing studies can be broadly classified into two main categories: (i) spatial coverage, which concerns the placement and deployment of sensors to ensure adequate monitoring of a region or set of targets, and (ii) temporal coverage, which focuses on how sensors manage their active and passive states over time to ensure sustainable and continuous monitoring. Below we summarize and categorize the literature accordingly.

7.1 Spatial Coverage

Spatial coverage addresses the fundamental question of *where* to place sensors so that all points or regions of interest (RoIs) are adequately monitored. Classic formulations distinguish three canonical problem variants: point (target) coverage, which requires observing a finite set of discrete targets; barrier coverage, which detects intrusions across a boundary; and area coverage, which ensures that every location within a continuous region is within the sensing range of at least one node [10–12]. The present work belongs to the point coverage category, as sensors are assigned to a fixed set of discrete Points of Interest (POIs).

Deployment strategies. A central design choice is how sensors are placed in the field. Random deployment is common in hostile or inaccessible environments and is typically followed by local repositioning algorithms — virtual-force methods, gradient descent, or potential-field techniques — to eliminate coverage holes [13–15]. Deterministic placement strategies instead rely on regular tessellation patterns (triangular, square, or hexagonal grids) or optimisation-based point selection to minimise the number of deployed nodes while jointly guaranteeing coverage and connectivity [12, 16]. Throughout this paper, we assume that deployment has already been decided and that coverage sets are fixed; the spatial assignment of sensors to POIs is therefore treated as a given input to our scheduling problem.

Metaheuristic optimisation. For large-scale or realistic instances, exact placement methods are often computationally intractable. Genetic Algorithms (GA), Particle Swarm Optimisation (PSO), Ant Colony Optimisation (ACO), and more recent nature-inspired methods such as Grey Wolf Optimisation (GWO) have been widely applied to multi-objective sensor placement, balancing coverage quality against

energy consumption and deployment cost [17–22]. Recent surveys confirm that such multi-objective formulations are the dominant approach to spatial coverage optimisation at scale [23, 24]. A shared limitation of this body of work, however, is that it focuses exclusively on the static placement decision and does not address *how* sensors should alternate between active and passive states once deployed — the temporal dimension that is the focus of the present work.

Positioning of the present work. The spatial coverage literature provides the structural foundation on which our problem rests: we inherit its definitions of point coverage, coverage sets, and k-coverage redundancy. However, spatial optimisation alone is insufficient when sensors are energy-constrained and cannot remain active indefinitely. The question of how to schedule sensor activations over time to maintain continuous spatial coverage — while respecting energy harvesting and consumption dynamics — is precisely the problem we address in the following sections.

7.2 Temporal Coverage

Temporal coverage explicitly addresses the time dimension of sensing: since a sensor may not remain continuously active due to energy constraints, the design objective shifts from placement to *scheduling* — orchestrating activation and sleep phases so that coverage is maintained over time while respecting energy budgets.

Lifetime-oriented scheduling. Early studies formulated the scheduling problem as a sequence of sensor covers, where each cover guarantees complete coverage over a specific time interval. The primary objective was to maximise overall network lifetime while respecting battery capacity constraints [3, 25]. Static scheduling approaches determine these covers in advance, whereas dynamic strategies adapt activation decisions in real time according to residual energy, local sensing information, or estimated harvested energy [24, 26]. Harizan and Kuila [27] extend this direction by jointly optimising target coverage, connectivity, and energy level in a genetic-algorithm framework that also admits a linear-programming relaxation; the authors demonstrate significant lifetime gains over naive activation schemes. A common limitation of these lifetime-oriented works, however, is that they assume batteries are non-rechargeable and do not model energy replenishment, so the resulting schedules are inherently finite.

Energy-harvesting aware scheduling. Energy harvesting and wireless recharging technologies fundamentally broaden the scheduling design space, as replenishment from solar, RF, or vibration sources introduces time-varying energy constraints that must be balanced against coverage demands. When effectively exploited, these patterns can enable near-perpetual network operation [28, 29]. Recent research therefore emphasises the necessity of integrated spatio-temporal frameworks that reconcile coverage and duty-cycle scheduling under harvested energy [30]. In EH-WSNs and Wireless Rechargeable Sensor Networks (WRSNs), scheduling must balance immediate coverage against future energy availability. This challenge has inspired distributed, dynamic algorithms based on Lyapunov optimisation for hybrid solar/wind harvesting scenarios, explicitly addressing coverage utility under stochastic energy arrivals [31]. Sah et al. [32] propose a geometric relay-node placement and scheduling strategy for EH-WSNs that enforces coverage-connectivity constraints while exploiting harvested

energy; their approach achieves good energy efficiency but relies on a heuristic and does not address the temporal continuity of coverage across repeated activation cycles.

ILP-based sensor scheduling. Integer Linear Programming has been applied to sensor scheduling in several related settings. Bose et al. [33] present an ILP formulation for periodic sensor scheduling that minimises instantaneous aggregated traffic in a sensor-cloud infrastructure; while they share our use of exact optimisation over discrete activation variables, their model assumes sensors are always powered and therefore does not account for energy harvesting or battery dynamics. The work most similar in spirit to ours is thus a combination of these threads: exact ILP-based scheduling applied to sensors that both consume and harvest energy, with the goal of guaranteeing *perpetual* temporal coverage.

Positioning of the present work. The survey above reveals a clear gap: no existing work simultaneously addresses (i) exact, optimal scheduling via ILP, (ii) energy-harvesting sensors operating under a net energy deficit when active, and (iii) the derivation of a finite repeatable hyperperiod that provably guarantees continuous temporal coverage over an infinite horizon. Our work fills this gap by proposing and comparing two ILP formulations — a non-preemptive and a fully-preemptive variant — that are mathematically proven to produce valid, indefinitely replayable schedules, and by characterising the trade-offs between feasibility, optimality, and synchronisation overhead.

8 Conclusion

In this paper, we presented a comparative analysis of two scheduling strategies, the Fully-preemptive Approach (FPA) and the Non-Preemptive Approach (NPA), for managing sensor activation in energy-harvesting systems. The objective was to evaluate their ability to maintain feasibility, minimize solving time, and reduce synchronization overhead under varying energy constraints. Through extensive simulations, we demonstrated that FPA consistently outperforms NPA. Its flexibility in managing activation and deactivation decisions allows it to sustain feasible operation even under challenging energy conditions, while maintaining low computational overhead, but NPA provides solutions of a better quality.

Future work will focus on extending this study by combining the strengths of both approaches. The idea is to design a hybrid method that selects suitable activation periods to ease synchronization while maintaining limited preemption to reduce the number of mode changes. Such an approach could provide a better trade-off between flexibility and stability, improving energy efficiency while minimizing coordination overhead in realistic, dynamic environments.

Furthermore, the current framework assumes deterministic and time-invariant energy harvesting rates $h(S_i)$, which is appropriate for the stable energy sources targeted in this work. Relaxing this assumption to account for stochastic harvesting rates, arising for instance from solar irradiance or wind variability, is identified as an important direction for future investigation.

Additionally, validating the proposed scheduling framework against real energy-harvesting traces from actual sensor deployments — for instance, using publicly

available solar irradiance or RF harvesting datasets — is identified as an important step toward confirming the practical applicability of the results beyond synthetic benchmarks.

Acknowledgements. This work was supported by the French National Research Agency (ANR) under the project HeRITAGEs, grant number ANR-23-CES25-0001 and by PHC Tassili 2024 Program number 51319QD.

References

[1] Sandhu, M.M., Khalifa, S., Jurdak, R., Portmann, M.: Task scheduling for energy-harvesting-based IoT: A survey and critical analysis. IEEE Internet of Things Journal **8**(18), 13825–13848 (2021) https://doi.org/10.1109/JIOT.2021.3086186

[2] Edalat, N., Motani, M.: Energy-aware task allocation for energy harvesting sensor networks. EURASIP Journal on Wireless Communications and Networking **2016**, 28 (2016) https://doi.org/10.1186/s13638-015-0490-3

[3] Li, J., Jiang, C., Wang, J., Xu, T., Xiao, W.: Mobile charging sequence scheduling for optimal sensing coverage in wireless rechargeable sensor networks. Applied Sciences **13**(5), 2840 (2023) https://doi.org/10.3390/app13052840

[4] Zhang, R., Zhao, J., Li, T., Liu, H.: A joint optimization strategy of coverage planning and energy scheduling for wireless rechargeable sensor networks. Processes **8**(10), 1324 (2020) https://doi.org/10.3390/pr8101324

[5] Xiong, Y., Li, J., Lu, M.: Critical location spatial-temporal coverage optimization in visual sensor network. Sensors **19**(19), 4106 (2019) https://doi.org/10.3390/s19194106

[6] Akbari, S.: Energy harvesting for wireless sensor networks review. In: 2014 Federated Conference on Computer Science and Information Systems, pp. 987–992 (2014). https://doi.org/10.15439/2014F85

[7] Sudevalayam, S., Kulkarni, P.: Energy harvesting sensor nodes: Survey and implications. IEEE Communications Surveys & Tutorials **13**(3), 443–461 (2011) https://doi.org/10.1109/SURV.2011.060710.00094

[8] Shaikh, F.K., Zeadally, S.: Energy harvesting in wireless sensor networks: A comprehensive review. Renewable and Sustainable Energy Reviews **55**, 1041–1054 (2016) https://doi.org/10.1016/j.rser.2015.11.010

[9] Ijemaru, G.K., Ang, K.L.-M., Seng, J.K.: Wireless power transfer and energy harvesting in distributed sensor networks: Survey, opportunities, and challenges. International Journal of Distributed Sensor Networks **18**(3), 15501477211067740 (2022) https://doi.org/10.1177/15501477211067740

[10] Meguerdichian, S., Koushanfar, F., Potkonjak, M., Srivastava, M.B.: Coverage problems in wireless ad-hoc sensor networks. In: Proceedings IEEE INFOCOM 2001, pp. 1380–1387 (2001). https://doi.org/10.1109/INFCOM.2001.916633

[11] Cardei, M., Thai, M.T., Li, Y., Wu, W.: Energy-efficient target coverage in wireless sensor networks. In: Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies., vol. 3, pp. 1976–19843 (2005). https://doi.org/10.1109/INFCOM.2005.1498475

[12] Ghosh, A., Das, S.K.: Coverage and connectivity issues in wireless sensor networks: A survey. Pervasive and Mobile Computing **4**(3), 303–334 (2008) https://doi.org/10.1016/j.pmcj.2008.01.003

[13] Howard, A., Matarić, M.J., Sukhatme, G.S.: Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem. In: Asama, H., Arai, T., Fukuda, T., Hasegawa, T. (eds.) Distributed Autonomous Robotic Systems 5, pp. 299–308. Springer, Tokyo (2002). https://doi.org/10.1007/978-4-431-65941-9_30

[14] Deif, D.S., Gadallah, Y.: Classification of wireless sensor networks deployment techniques. IEEE Communications Surveys & Tutorials **16**(2), 834–855 (2014) https://doi.org/10.1109/SURV.2013.091213.00018

[15] Tirandazi, P., Rahiminasab, A., Ebadi, M.J.: An efficient coverage and connectivity algorithm based on mobile robots for wireless sensor networks. Journal of Ambient Intelligence and Humanized Computing **14**(7), 8291–8313 (2023) https://doi.org/10.1007/s12652-021-03597-9

[16] Tripathi, A., Gupta, H.P., Dutta, T., Mishra, R., Shukla, K.K., Jit, S.: Coverage and connectivity in wsns: A survey, research issues and challenges. IEEE Access **6**, 26971–26992 (2018) https://doi.org/10.1109/ACCESS.2018.2833632

[17] Hanh, N.T., Binh, H.T.T., Hoai, N.X., Palaniswami, M.S.: An efficient genetic algorithm for maximizing area coverage in wireless sensor networks. Information Sciences **488**, 58–75 (2019) https://doi.org/10.1016/j.ins.2019.02.059

[18] Siamantas, G., Kandris, D.: Particle swarm optimization for k-coverage and 1-connectivity in wireless sensor networks. Electronics **13**(23), 4841 (2024) https://doi.org/10.3390/electronics13234841

[19] Muhammad, H., Nam, H.: Sensor node deployment optimization for continuous coverage in WSNs. Sensors **25**(12), 3620 (2025) https://doi.org/10.3390/s25123620

[20] Qasim, T., Minhas, Q.u.A., Mujahid, A., Bhatti, N.A., Mushtaq, M., Saleem, K., Mahmood, H., Shujah Islam Sameem, M.: Aco-discreet: An efficient node deployment approach in wireless sensor networks. In: Latifi, S. (ed.) Information

Technology - New Generations, pp. 43–48. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-54978-1_6

[21] Elhoseny, M., Tharwat, A., Farouk, A., Hassanien, A.E.: K-coverage model based on genetic algorithm to extend WSN lifetime. IEEE Sensors Letters **1**(4), 1–4 (2017) https://doi.org/10.1109/LSENS.2017.2720648

[22] Ou, Y., Qin, F., Zhou, K.-Q., Yin, P.-F., Mo, L.-P., Mohd Zain, A.: An improved grey wolf optimizer with multi-strategies coverage in wireless sensor networks. Symmetry **16**(3) (2024) https://doi.org/10.3390/sym16030286

[23] Anusuya, P., Vanitha, C.N., Cho, J., Easwaramoorthy, S.V.: A comprehensive review of sensor node deployment strategies for maximized coverage and energy efficiency in wireless sensor networks. PeerJ Computer Science **10**, 2407 (2024) https://doi.org/10.7717/peerj-cs.2407

[24] He, S., Shi, K., Liu, C., Guo, B., Chen, J., Shi, Z.: Collaborative sensing in internet of things: A comprehensive survey. Commun. Surveys Tuts. **24**(3), 1435–1474 (2022) https://doi.org/10.1109/COMST.2022.3187138

[25] Slijepcevic, S., Potkonjak, M.: Power efficient organization of wireless sensor networks. In: ICC 2001. IEEE International Conference on Communications. Conference Record (Cat. No.01CH37240), vol. 2, pp. 472–4762 (2001). https://doi.org/10.1109/ICC.2001.936985

[26] Ye, W., Heidemann, J., Estrin, D.: An energy-efficient MAC protocol for wireless sensor networks. In: Proceedings IEEE INFOCOM 2002, vol. 3, pp. 1567–1576 (2002). https://doi.org/10.1109/INFCOM.2002.1019408

[27] Harizan, S., Kuila, P.: Coverage and connectivity aware energy efficient scheduling in target based wireless sensor networks: an improved genetic algorithm based approach. Wireless Networks **25**, 1995–2011 (2019) https://doi.org/10.1007/s11276-018-1792-2

[28] Niyato, D., Hossain, E., Fallahi, A.: Sleep and wakeup strategies in solar-powered wireless sensor/mesh networks: Performance analysis and optimization. IEEE Transactions on Mobile Computing **6**(2), 221–236 (2007) https://doi.org/10.1109/TMC.2007.30

[29] Lu, X., Wang, P., Niyato, D., Kim, D.I., Han, Z.: Wireless charging technologies: Fundamentals, standards, and network applications. IEEE Communications Surveys & Tutorials **18**(2), 1413–1452 (2016) https://doi.org/10.1109/COMST.2015.2499783

[30] Ojha, A., Gupta, B.: Evolving landscape of wireless sensor networks: a survey of trends, timelines, and future perspectives. Discover Applied Sciences **7**(8), 825 (2025) https://doi.org/10.1007/s42452-025-07070-6

[31] Bao, X., Jiang, Y., Han, L., Xu, X., Zhu, H.: Distributed dynamic scheduling algorithm of target coverage for wireless sensor networks with hybrid energy harvesting system. Scientific Reports **14**(1), 27931 (2024) https://doi.org/10.1038/s41598-024-78671-1

[32] Sah, D.K., Srivastava, S., Kumar, R., Amgoth, T.: An energy efficient coverage aware algorithm in energy harvesting wireless sensor networks. Wireless Networks **29**, 1175–1195 (2023) https://doi.org/10.1007/s11276-022-03125-3

[33] Bose, S., Chowdhury, A., Mukherjee, N.: Scheduling periodic sensors for instantaneous aggregated traffic minimization. Wireless Networks **30**, 3257–3268 (2024) https://doi.org/10.1007/s11276-024-03722-4