Abstract

We consider the problem of maximizing the lifetime of a wireless sensor network with a mobile sink. The sink travels at finite speed among a subset of possible sink locations to collect data from a stationary set of sensor nodes. The problem we consider chooses a subset of sink locations for the sink to visit, finds a tour for the sink among the selected sink locations, and prescribes an optimal data routing scheme from the sensor nodes to each location visited by the sink. The sink’s tour is constrained by the time it spends at each location collecting data. We examine two variations of this problem based on assumptions regarding delay tolerance. We provide exact mixed-integer programming formulations to model this problem, along with cutting planes, preprocessing techniques, and a Benders decomposition algorithm to improve its solvability. Computational results demonstrate the effectiveness of the proposed methods.

Keywords: Wireless Sensor Networks; Lifetime Maximization; Sink Trajectory; Mixed-Integer Programming; Mobile Sink.

1 Introduction

We examine a wireless sensor network (WSN) consisting of a set of limited-power sensors that are deployed in an area for data collection purposes. Sensors may for instance measure a physical attribute of their surrounding area (such as temperature) or sense the movement of objects in their vicinity. The collected information needs to be transmitted for further processing to a data-collecting node called the sink. Sensors are often deployed randomly, but in high quantities to
prevent coverage breach. This dense deployment of the sensor nodes allows multihop delivery of the collected information: Each sensor can remotely communicate with other nearby sensors via wireless links and use them to relay its collected information to the sink. The need for energy conservation and frequency reuse make multihop delivery an essential component of any routing protocol for WSNs [19, 20]. See [1] for a complete survey on WSNs and their applications.

Because sensors require energy (often provided by battery power) to transmit and receive data, sensor energy is a scarce resource in WSNs. However, if sensors are deployed in not-easily-accessible or hostile environments, the task of replacing their batteries becomes impractical. An important optimization problem is to devise balanced communication schemes between the sensors and the sink to prolong WSN lifetime. Various lifetime maximization problems for WSNs have been studied recently [8, 9, 17, 21, 27], especially with respect to the task of exploiting sink mobility [4, 5, 11, 12, 16, 18, 22, 23, 24, 25, 27, 28]. When the sink is static (i.e., located at one fixed position), the sensor nodes closer to the sink are burdened with traffic aggregation because they need to relay other nodes’ traffic. Therefore, these nodes exhaust their battery energy sooner than the rest of the sensor nodes, disconnecting the rest of the network from the sink. This phenomenon is known as the energy hole problem [14, 15, 26]. Moving the sink in the sensor field can mitigate the energy hole problem, which results in an extended network lifetime. In this paper, we focus on situations in which there exists one sink that moves over a subset of pre-specified sink locations in the sensor field.

In WSNs having a static sink, the main decision affecting network lifetime is how to route the data from each sensor node to the sink. On the other hand, when the sink is mobile, one must also determine how long the sink should stay at each sink location. We will refer to these decisions as scheduling decisions. When the sink can move arbitrarily fast between its possible locations, scheduling only refers to the amount of time that the sink stays at each location. (Later in this paper, we will also include the order in which those locations are visited.) The problem is then to find a set of routing and scheduling decisions that maximize the network lifetime, where the lifetime is defined as the time until the first sensor node expends all of its battery power [8, 9].

At higher discharge rates, batteries are typically less efficient at converting their chemically stored energy into available electrical energy (see, for example, [7]). Therefore, it is energy-efficient to use routing and scheduling schemes that require frequent changes in the sensors’ energy consumption rates. In this paper, we consider a cyclic model in which the sink is required to finish some (positive integer) C cycles during the network lifetime (see [13, 27] for similar cyclic models). Our lifetime maximization problem is then focused on finding an optimal trajectory for the sink as
well as optimal routing schemes for the sensor nodes during one cycle.

Papadimitriou and Georgiadis [18] propose a linear programming formulation for the problem of maximizing the lifetime of a wireless sensor network with one mobile sink. Their formulation integrates both routing and scheduling decisions. (See also [5] for related work on this problem that employs column-generation techniques.) A major assumption in these papers is that the sink’s travel time from one location to another is negligible, or equivalently, that there are many stationary sinks, one at each possible location. However, when the sink’s travel times are significant compared to its dwelling times at each sink location, the order in which the sink visits these locations becomes important.

Keskin et al. [13] formulate a lifetime maximization problem that assumes nonzero sink travel times, and propose heuristic algorithms to solve it. Their formulation requires a shortest path routing of the collected data from sensor nodes to the sink, which may result in a suboptimal network lifetime as discussed in [18]. Unlike [13], the models we propose in this paper incorporate the routing of data from sensor nodes to the sink.

We formulate the WSN lifetime maximization problem as a mixed-integer program (MIP) to capture the order in which the sink locations are visited during each cycle. We assume that the sink cannot gather information from the sensor nodes while traveling between different sink locations. As a result, the underlying application must be delay-tolerant with a maximum tolerable delay $D$, i.e., data can be delivered to the sink with a maximum delay of $D$ time units. In some delay-tolerant WSN applications, sensor nodes are capable of postponing data transmission to the sink by storing the data locally. Alternatively, if the sensor nodes’ computational power and storage capabilities are limited, the sensors may instead continue data transmission to their target sink locations, where the transmitted data is stored until a subsequent visit by the sink. While our models and solution methods in this paper can be applied to both cases, we take the perspective of the latter case, where sink locations store data. In addition to an optimal set of routing schemes from the sensor nodes to each sink location, our formulations find an optimal trajectory for the sink that satisfies the maximum tolerable delay restriction for the underlying application.

The rest of this paper is organized as follows. In Section 2 we present MIP formulations for the lifetime maximization problems examined in this paper. In Section 3, we describe a cutting-plane approach for more effectively solving one variant of the problem. Next, we recast the problem as a two-stage optimization problem in Section 4, and prescribe a Benders decomposition algorithm for its solution. Computational results are presented in Section 5, and we conclude the paper with a summary and future research directions in Section 6.
2 Problem Statement

We begin in Section 2.1 by introducing notation and discussing several key characteristics of the lifetime maximization problem. We then formulate the lifetime maximization problem in Section 2.2 and present an alternative problem variation in Section 2.3. We provide a comparison of the presented models in Section 2.4.

2.1 Preliminaries

Let $N$ be the set of sensor nodes (“nodes” for short) and $L$ be the set of sink locations of a WSN. The sink periodically visits sink locations in a subset of $L$ and collects data from the nodes. When the sink is at location $l \in L$, node $i$ can send its data (or relay other nodes’ data) only to a set of downstream neighbors $S^l_i$. This set is often given as $S^l_i = \{ j \in N \cup \{l \} : \mu_{ij} \leq \bar{\mu} \}$, where $\mu_{ij}$ is the Euclidean distance between $i$ and $j$ and $\bar{\mu}$ is the maximum transmission range of any node. Denote by $E_i$ the initial energy level of node $i$, and suppose that the sink is required to finish $C$ cycles during the network lifetime. Therefore, maximizing the network lifetime is equivalent to maximizing the cycle duration, while ensuring that each sensor node’s energy expenditure in a cycle does not exceed $\bar{E}_i = E_i / C$. The required energy for transmitting one unit of data from node $i$ to a point $j \in S^l_i$ while the sink is at location $l \in L$ is denoted by $e_{ij}^l > 0$. Similarly, $\gamma > 0$ is the required energy for receiving one unit of data at every node. Finally, $d_i > 0$ represents the data generation rate of node $i$.

Under the commonly-used assumption of zero sink travel times [5, 18], the lifetime maximization problem in WSNs with a mobile sink can be formulated as a linear program. Let variable $z_l$ determine how long the sink stays at location $l \in L$ during each cycle, and $y_{ij}^l$ represent the volume of data transmitted from sensor node $i$ to $j \in S^l_i$ while the sink is at location $l$. The lifetime maximization problem can be formulated as follows [18].

\begin{align*}
\text{Max} & \quad \sum_{l \in L} z_l \\
\text{s.t.} & \quad \sum_{l \in L} \left( \sum_{j \in S^l_i} e_{ij}^l y_{ij}^l + \sum_{j : i \in S^l_j} \gamma y_{ji}^l \right) \leq \bar{E}_i, \quad \forall i \in N \\
& \quad \sum_{j \in S^l_i} y_{ij}^l - \sum_{j : i \in S^l_j} y_{ji}^l = d_i z_l, \quad \forall i \in N, \ l \in L \\
& \quad z_l \geq 0, \quad \forall l \in L
\end{align*}
Figure 1: A WSN with five sink locations

\[ y_{ij}^l \geq 0, \quad \forall i \in N, \ j \in S_i^l, \ l \in L. \]  

The objective function optimizes the WSN lifetime by maximizing the sum of time that the sink spends staying over all possible sink locations. (The actual lifetime would equal this value multiplied by \( C \).) Constraints (1b) ensure that the total energy expenditure of each sensor node \( i \) during each cycle does not exceed \( E_i \). Constraints (1c) are flow conservation constraints and ensure that for each sink location \( l \), every sensor node’s out-flow equals its in-flow plus the volume of data generated at the sensor node during the sink’s stay at \( l \). Constraints (1d) and (1e) state the nonnegativity requirements for \( z \)- and \( y \)-variables. We refer to this model as the Mobile Sink Model (MSM).

Observe that this model implicitly assumes that data is routed to the sink locations in the same manner each time the sink traverses a cycle. (Here, routing includes both path selection and rate allocation.) This assumption is made for the sake of simplifying routing and data collection, and all models we present here employ this cyclic routing assumption.

When the sink moves with a finite speed, Formulation (1) may not necessarily yield a feasible trajectory for the sink. Consider the WSN in Figure 1(a) where the sink visits sink locations \( \{l_1, l_2, l_3, l_4\} \). Figure 1(b) contains the distances between each pair of sink locations. Suppose that in some solution to (1), the sink stays at location \( l_1 \) for \( z_1 = 5 \) time units, during which time...
the sensor nodes route data to the sink according to a communication pattern specified by vector \( y^1 \). Also, suppose that the solution requires a \( z_2 = 7 \), \( z_3 = 8 \), and \( z_4 = 6 \) time-unit stay at sink locations \( l_2 \), \( l_3 \), and \( l_4 \) with communication schemes \( y^2 \), \( y^3 \), and \( y^4 \), respectively. If the sink moves arbitrarily fast, as assumed in MSM, then this solution need not prescribe a specific order of visit to the sink locations.

Suppose instead that the sink travels with a speed of 2 distance units per time unit. We can build a feasible trajectory for the sink based on the foregoing optimal \( z \)- and \( y \)-values as follows:

- The sink leaves \( l_1 \) toward \( l_2 \) at time \( t = 0 \). During the sink’s travel from \( l_1 \) to \( l_2 \), the sensor nodes transmit their collected data to \( l_2 \) according to the communication pattern specified by \( y^2 \). The transmitted data is stored at \( l_2 \) until the sink arrives there at time \( t = 2 \).

- The sink stays at \( l_2 \) until \( t = 7 \) and collects data from the sensor nodes (according to \( y^2 \)), and then heads to \( l_3 \).

- The sink reaches \( l_3 \) at time \( t = 11.5 \), collects all stored data immediately, continues collecting data until \( t = 15 \) (according to \( y^3 \)), and then leaves \( l_3 \) to go to \( l_4 \).

- The sink then reaches \( l_4 \) at \( t = 18.5 \), and leaves there at \( t = 21 \).

- Finally, the sink reaches \( l_1 \) at \( t = 25 \), and stays there until the end of the cycle at \( t = 26 \).

Clearly, the sink can only travel on an arc if its travel time is no more than the dwelling time at the destination. Now, suppose that the sink only moves 1 distance unit per time unit. There exists no feasible trajectory for the sink that spans \( \{l_1, l_2, l_3, l_4\} \), given \( z_1, \ldots, z_4 \): The fact that \( z_1 + \cdots + z_4 = 26 \) implies that the sink’s tour cannot take longer than 26 time units, but the shortest tour spanning these four nodes is 28 time units.

### 2.2 Problem Formulation

In this section, we consider the WSN lifetime maximization problem with a finite sink speed. Suppose that it takes \( t_{lm} > 0 \) time units for the sink to go from location \( l \in L \) to location \( m \in L \). Data generated while the sink travels from \( l \) to \( m \) is transmitted and stored at location \( m \) and is collected by the sink once it reaches \( m \). The maximum tolerable delay is assumed to be \( D \), so the sink can only travel on arcs that satisfy \( t_{lm} \leq D \). We will refer to the directed graph \( G = (L, A) \) with arc set \( A = \{(l, m) : t_{lm} \leq D\} \) as the sink graph. In the sink graph define \( N_l^+ = \{m \in L : (l, m) \in A\} \) and \( N_l^- = \{m \in L : (m, l) \in A\} \). Our goal is to maximize the WSN
lifetime by finding a tour for the sink over a subset of \( L \), as well as routing patterns from each sensor node to each visited sink location. We refer to this problem as \( \text{MSM-FS1} \).

To formulate the problem as a MIP, we introduce binary variables \( u_{lm} \) that equal 1 if and only if the sink travels along arc \((l, m)\) in the underlying sink graph. Recall that the main constraint in this model requires the sink’s dwelling time at \( l \) to be at least as large as \( t_{ml} \) if \( u_{ml} = 1 \). Our formulation satisfies this requirement because the transmission time to each sink location is now given by \( z_l + \sum_{m \in N^-_l} t_{ml} u_{ml} \), where \( z_l \) is defined as the sink’s stop length at \( l \). We also introduce auxiliary binary variables \( v_l \) that equal 1 if and only if location \( l \) is visited by the sink. Problem \( \text{MSM-FS1} \) can be formulated as follows, where constants \( M_l, \forall l \in L \), are large values that we specify later in this paper.

Max \( \sum_{l \in L} \left( z_l + \sum_{m \in N^-_l} t_{ml} u_{ml} \right) \)  

s.t. \( \sum_{l \in L} \left( \sum_{j \in S^l_i} e^l_j y^l_j + \sum_{j : i \in S^l_j} \gamma y^l_j \right) \leq \bar{E}_i, \quad \forall i \in N \)  

\( \sum_{j \in S^l_i} y^l_{ij} - \sum_{j : i \in S^l_j} y^l_{ji} = d_i \left( z_l + \sum_{m \in N^-_l} t_{ml} u_{ml} \right), \quad \forall i \in N, \ l \in L \)  

\( z_l \leq M_l v_l, \quad \forall l \in L \)  

\( \sum_{m \in N^+_l} u_{lm} = v_l, \quad \forall l \in L \)  

\( \sum_{m \in N^-_l} u_{ml} = \sum_{m \in N^+_l} u_{lm}, \quad \forall l \in L \)  

\( \sum_{l \in S} \sum_{m \in S} u_{lm} \geq v_k + v_r - 1, \quad \forall S \subset L : 2 \leq |S| \leq |L| - 2, \ k \in S, \ r \in S \)  

\( v_l \in \{0,1\}, \quad \forall l \in L \)  

\( u_{lm} \in \{0,1\}, \quad \forall l \in L, \ m \in N^+_l \)  

\( z_l \geq 0, \quad \forall l \in L \)  

\( y^l_{ij} \geq 0, \quad \forall i \in N, \ j \in S^l_i, \ l \in L. \)

The objective maximizes the time that the sink dwells at sink locations, plus the time it spends traveling between these locations. Constraints (2b) and (2c) are the energy and flow conservation
constraints, respectively, similar to (1b) and (1c). Constraints (2d) force \( v_l = 1 \) whenever \( z_l > 0 \), \( \forall l \in L \). Constraints (2e) define the relationship between \( u- \) and \( v- \) variables and ensure that at most one arc enters each sink location, while constraints (2f) ensure that for each \( l \in L \), the number of incoming and outgoing arcs in a feasible tour of sink locations are equal. Finally, constraints (2g) are the generalized subtour elimination constraints [3]. These constraints ensure that for any subset \( S \subset L \) such that \( 2 \leq |S| \leq |L| - 2 \), there exists an arc going from \( S \) to \( \overline{S} \) if both \( S \) and \( \overline{S} \) contain a sink location that is on the tour.

**Remark 1.** Note that when \( C = 1 \), the sink does not need to return to its initial location and therefore, one seeks a feasible path for the sink instead of a cycle. In this case, let binary variable \( \delta_l (\tau_l) \) indicate whether the sink’s path starts (ends) at sink location \( l \). We would then substitute constraints (2e) and (2f) with the following constraints.

\[
\begin{align*}
\delta_l + \sum_{m \in N_l^+} u_{lm} &= v_l, \quad \forall l \in L \\
\sum_{m \in N_l^-} u_{ml} - \sum_{m \in N_l^+} u_{lm} &= \tau_l - \delta_l, \quad \forall l \in L \\
\sum_{l \in L} \delta_l &= 1 \\
\delta_l &\in \{0, 1\}, \quad \forall l \in L \\
\tau_l &\in \{0, 1\}, \quad \forall l \in L.
\end{align*}
\]

Moreover, because we seek a directed path, the left-hand-side of subtour elimination constraints (2g) becomes \( \sum_{l \in S} \sum_{m \in \overline{S}} (u_{lm} + u_{ml}) \). Note that the equality \( \sum_{l \in L} \tau_l = 1 \) is implied by the aggregation of (3b) and (3c). \( \square \)

For simplicity, we preprocess the cases in which the sink visits only a single sink location \( l \in L \). To accomplish this, we solve the MSM in which \( L \) is restricted to the singleton \( \{l\} \). Let \( M'_l \) be the optimal MSM objective to this problem. From this point forward, we assume that the sink visits at least two locations in \( L \). Our “ultimate” solution is one having the largest objective among the single-sink-location solutions and the multiple-sink-location solution we obtain.

We next address the derivation of the \( M_l \)-values used in (2d). Noting that \( M_l \) must be at least as large as the maximum dwelling time at any location \( l \in L \), we consider the scenario in which the sink spends (almost) the entire cycle duration at location \( l \). Recall that \( M'_l \) is the maximum amount of time the sink can spend at \( l \in L \), and the sink must spend some time in transit to \( l \),
due to the assumption that the sink visits at least two locations. Hence, $M'_l$ is an upper bound on $z_l + t_{ml}$, where $m$ is whichever sink location visited immediately before $l$. Therefore,

$$M_l = M'_l - \min_{m \in N_l} t_{ml}$$

(4)

can be used in (2d). Alternatively, we can determine the minimum amount of energy that a node must expend while the sink is at $l$. Dividing the node’s energy by this minimum expenditure yields a valid bound for $M'_l$, and enables us to use

$$M_l = \min_{i \in N} \left\{ \frac{E_i}{d_i \min_{j \in S_i} \{ e_{ij} \}} \right\} - \min_{m \in N_l} t_{ml}$$

(5)

as a valid upper bound on $z_l$. Although calculating $M_l$ via (4) is more computationally expensive than calculating (5), it generally results in a stronger bound, and hence we compute $M_l$ via (4) in the remainder of this paper.

2.3 Alternative Delay Model

Problem MSM-FS1 assumes real-time transmission of data whenever possible, i.e., a sink location stores data only when the sink is en route to the location, and data transmission happens in real time while the sink dwells at a sink location. This model is appropriate for applications such as surveillance and real-time monitoring where unnecessary delay must be avoided, except, e.g., when the sink is traveling between the locations. In other more delay-tolerant WSN applications, it may be more appropriate to allow sensors to transmit data to some location $l$, even when the sink is not currently present at, or en route to, location $l$. However, we still honor the maximum delay restriction stating that once information begins transmission to location $l$, the sink must arrive at node $l$ no more than $D$ time units later.

To illustrate this alternative delay model, consider again the wireless sensor network in Figure 1(a) and suppose that the sink’s trajectory is $l_1 \to l_2 \to l_3 \to l_1$. The cycle time equals $\sum_{m=1}^{3} r_m$, where $r_m$ is the length of the period during which the sensor nodes send their data to sink location $l_m$ according to the data transmission scheme given by $y^m$. Data is transmitted to these sink locations in the same order in which they are later visited by the sink. Unlike (2), it is now possible, for instance, that while the sink is moving from $l_1$ to $l_2$, the sensor nodes start transmitting data to $l_3$.

First, note that the the sink’s tour length cannot exceed the cycle time. Therefore, there must
exist at least one sink location \( l \) on the sink’s tour, such that sensors are still sending data to \( l \) when the sink arrives at \( l \). (This characterization includes the case in which sensors stop transmitting to \( l \) immediately when the sink arrives at \( l \).) We will refer to one such sink location as the tour’s \textit{origin}. This assumption is necessary to ensure the uniformity of data routing and collection during each cycle performed by the sink.

In our example, if \( l_1 \) serves as the origin, delay restrictions impose the following constraints.

\[
\begin{align*}
t_{12} &\leq D, \\
\max\{t_{12} - r_2, 0\} + t_{23} &\leq D, \\
\max\{\max\{t_{12} - r_2, 0\} + t_{23} - r_3, 0\} + t_{31} &\leq D. 
\end{align*}
\]

Here inequality (6a) ensures that transmission delay along the path \( l_1 \to l_2 \) does not exceed \( D \), while (6b) and (6c) ensure that this requirement is met along \( l_1 \to l_2 \to l_3 \) and \( l_1 \to l_2 \to l_3 \to l_1 \), respectively. Moreover, because \( l_1 \) serves as the origin, the sensor nodes must still be sending data to \( l_1 \) when the sink arrives there, and so the following constraint must be satisfied:

\[
r_1 \geq \max\{\max\{t_{12} - r_2, 0\} + t_{23} - r_3, 0\} + t_{31}. 
\]

To formulate these constraints when the tour’s origin is not determined \textit{a priori}, let variables \( u_{lm} \) and \( v_l \) determine the sink’s trajectory (as modeled by (2e)–(2i)). Also, define binary variable \( s_l \), for \( l \in L \), which equals 1 if location \( l \) serves as the tour’s origin. We impose constraints analogous to (6) for every combination of origin node \( l \in L \) and sink location \( m \in L \). These constraints are made inactive (by weakening the right-hand-side with a big-\( M \) term) whenever \( s_l = 0 \) or \( v_m = 0 \), and are otherwise active. To model these delay constraints, let variable \( q_{lm}, \forall l, m \in L \), represent the amount of time that sensors transmit data to \( m \) before the sink begins moving toward \( m \) from its preceding sink location, given that \( l \) serves as the tour’s origin. Therefore, data is transmitted to \( m \) starting from \( q_{lm} + \sum_{k \in N_m^-} t_{km}u_{km} \) time units before the sink arrives at \( m \), and the following set of constraints must hold:

\[
q_{lm} + \sum_{k \in N_m^-} t_{km}u_{km} \leq D, \quad \forall l \in L, m \in L. 
\]

Constraints (8a) and (8b) below ensure the existence of a designated origin from among those nodes
visited by the tour.

\[ \sum_{l \in L} s_l = 1 \]  
(8a)

\[ s_l \leq v_l, \quad \forall l \in L. \]  
(8b)

Without loss of generality, if the sink’s tour traverses sink locations 1, \ldots, p in order with origin node 1, then \( q_{1,w+1} = \max\{q_{1w} + t_{w-1,w} - r_w, 0\} \), where all index addition and subtraction is performed modulo \( p \), for each \( w = 1, \ldots, p \). Moreover, because location 1 is the origin (and location 2 should not collect data before the sink leaves 1), \( q_{12} = 0 \). All other \( q \)-variables should equal 0. The following constraints properly define the \( q \)-variables as such in terms of the \( r \)-, \( s \)-, and \( u \)-variables.

\[ q_{lm} \geq \sum_{k \in N_m} q_{lk} + \sum_{h \in N_k} (t_{hk} - r_k)u_{hk} \begin{bmatrix} u_{km} - D(1 - s_l) \end{bmatrix}, \quad \forall l \in L, m \in L \]  
(9a)

\[ D(1 - s_l) + r_l \geq q_{ll} + \sum_{m \in N_l} t_{ml}u_{ml}, \quad \forall l \in L. \]  
(9b)

To see that (9a) and (9b), along with (7), correctly define the \( q \)-variables, we first claim that there exists a solution to these constraints in which each \( q \)-variable takes the smallest value allowed by (9a) and (9b) (due to \( q_{lm} \) being on the left-hand-side of (7), which is of the \( \leq \) sense, and because \( q \)-variables are not present elsewhere in the model). If \( s_l = 0 \), then note that setting \( q_{lk} = 0, \forall k \in L \), is feasible: The right-hand-side of (9a) would be no more than \( t_{hk} - D \leq 0 \) (for any \( h \in N_k^- \)), and (9b) permits \( q_{ll} = 0 \) due to the fact that \( t_{ml} \leq D \) for all \( m \in N_l^- \). Now consider the \( l \in L \) such that \( s_l = 1 \). If \( v_m = 0 \), then \( \sum_{k \in N_m} u_{km} = \sum_{k \in N_m^+} u_{mk} = 0 \), and thus setting \( q_{lm} = 0 \) is feasible to (9a). Finally, consider \( m \in L : v_m = 1 \). Reindex the locations so that the tour visits locations 1, \ldots, p in order, with origin \( l = 1 \) (where \( p \geq 2 \)). To begin, note that (9b) restricts \( r_1 \geq q_{11} + t_{p1} \), and so (9a) for \( m = 2 \) reduces to \( q_{12} \geq q_{11} + t_{p1} - r_1 \), which is thus dominated by \( q_{12} \geq 0 \). Hence, \( q_{12} = 0 \) as desired. By induction, assume that \( q_{12}, \ldots, q_{1w} \) are correctly defined for some \( 2 \leq w \leq p \). Constraint (9a) defines \( q_{l,w+1} \geq q_{lw} + t_{w-1,w} - r_w \) (where \( w+1 \equiv 1 \) if \( w = p \), and \( w-1 \equiv p \) if \( w = 1 \)), which along with the nonnegativity of the \( q \)-variables forces \( q_{l,w+1} \geq \max\{q_{lw} + t_{w-1,w} - r_w, 0\} \) as desired. Repeating this argument shows that \( q_{lw} \) is correctly defined for all \( w = 1, \ldots, p \).

To linearize (9a), we introduce nonnegative variables \( \alpha_{lk} = q_{lk}u_{km}, \beta_{hk} = u_{hk}u_{km} \), and
\[ \lambda_{hk} = r_k u_{hk} u_{km}, \] and add the following constraints:

\[
D(1 - u_{km}) + \alpha_{lk} \geq q_{lk}, \quad \forall l \in L, \; m \in L, \; k \in N_m^-
\] (10)

\[
\beta_{hk} \geq u_{hk} + u_{km} - 1, \quad \forall h \in L, \; k \in N_h^+, \; m \in N_k^+
\] (11)

\[
\lambda_{hk} \leq r_k, \quad \forall h \in L, \; k \in N_h^+, \; m \in N_k^+
\] (12)

\[
\lambda_{hk} \leq D u_{hk}, \quad \forall h \in L, \; k \in N_h^+, \; m \in N_k^+
\] (13)

\[
\lambda_{hk} \leq D u_{km}, \quad \forall h \in L, \; k \in N_h^+, \; m \in N_k^+
\] (14)

where (9a) is now revised as

\[
q_{lm} \geq \sum_{k \in N_m^-} \left[ \alpha_{lk} + \sum_{h \in N_k^-} (t_{hk} \beta_{hk} - \lambda_{hk}) \right] - D(1 - s_l).
\]

Again, because \(\alpha\)- and \(\beta\)-variables appear on the right-hand-side of (9a) (a \(\geq\)-constraint), there always exists an optimal solution in which these variables take on their smallest possible value permitted by (10) and (11), and so the upper-bounding constraints required to linearize \(\alpha\)- and \(\beta\)-variables are unnecessary in this formulation. By a similar argument, the lower-bounding constraints for linearizing \(\lambda\)-variables are not required in the formulation, because there always exists an optimal solution in which these variables assume their largest possible values. Hence, problem MSM-FS2 can be formulated as follows.

Max \( \sum_{l \in L} r_l \) (15a)

s.t. \( \sum_{l \in L} \left( \sum_{j \in S_i^l} e_{ij}^l y_{ij}^l + \sum_{j \in S_j^l} \gamma_{ji}^l y_{ji}^l \right) \leq \overline{E}_i, \quad \forall i \in N \) (15b)

\( \sum_{j \in S_i^l} y_{ij}^l - \sum_{j \in S_j^l} y_{ji}^l = d_i r_l, \quad \forall i \in N, \; l \in L \) (15c)

\( r_l \leq M_{l} v_l, \quad \forall l \in L \) (15d)

Constraints (2e)–(2i) (15e)

Constraints (7)–(14) (15f)

\( q_{lm} \geq 0, \quad \forall l \in L, \; m \in L \) (15g)

\( r_l \geq 0, \quad \forall l \in L \) (15h)
\[
\alpha_{lkm}, \beta_{hkm}, \lambda_{hkm} \geq 0, \quad \forall l \in L, k \in N_l^+, \; m \in N_k^+. \quad (15i)
\]

Formulation (15) contains an exponential number of subtour elimination constraints. In practice, one can relax the subtour elimination constraints, and add those inequalities that are violated at each node of the branch-and-bound tree by a polynomially solvable separation routine [10]. We next formulate an alternative MIP model for MSM-FS2 that is polynomially sized. To that end, we decompose each dwelling time \( r_l \) into variables \( r_{1l} \) and \( r_{2l} \), where \( r_{1l} \) represents the amount of time that sensors send data to \( l \) before the sink arrives at \( l \), and \( r_{2l} \) represents the amount of time that the sink stays at \( l \). Next, define the start of a sink’s cycle as the time the sink departs the origin. We also introduce variables \( \pi_l \) as the duration from the start of each cycle until the sensors start transmitting data to \( l \), and \( \rho_l \) as the duration from the start of each cycle until the sink reaches \( l \).

Given these new variable definitions, consider the following alternative formulation for MSM-FS2, where the \( \rho \)- and \( \pi \)-variables serve the dual roles of enforcing delay limits and preventing subtours.

\[
\text{Max} \sum_{l \in L} r_{1l} + \sum_{l \in L} r_{2l} \quad (16a)
\]

\[
\text{s.t.} \sum_{i \in L} \left( \sum_{j \in S_i^l} e_{ij} y_{ij}^l + \gamma y_{ji}^l \right) \leq E_i, \quad \forall i \in N \quad (16b)
\]

\[
\sum_{j \in S_i^l} y_{ij}^l - \sum_{j : i \in S_j^l} y_{ji}^l = d_i (r_{1l} + r_{2l}), \quad \forall i \in N, \; l \in L \quad (16c)
\]

\[
r_{1l} + r_{2l} \leq M'l, \quad \forall l \in L \quad (16d)
\]

Constraints (2e), (2f), (2h), (2i), (8a), and (8b) (16e)

\[
\rho_m \geq \sum_{l \in N_m^-} t_{lm} u_{lm}, \quad \forall m \in L \quad (16f)
\]

\[
\pi_m \leq M(2 - s_l - u_{lm}), \quad \forall l \in L, m \in L \quad (16g)
\]

\[
\rho_m \geq \rho_l + r_{2l} + t_{lm} u_{lm} - M(1 - u_{lm} + s_l), \quad \forall l \in L, m \in L \quad (16h)
\]

\[
r_{1l} + r_{2l} \geq \pi_m - \pi_l - M(1 - u_{lm} + s_l), \quad \forall l \in L, m \in L \quad (16i)
\]

\[
r_{1l} + r_{2l} \geq \rho_l - \pi_l - D(1 - s_l), \quad \forall l \in L \quad (16j)
\]

\[
r_{1l} + r_{2l} \geq t_{ml}(u_{ml} + s_l - 1), \quad \forall l \in L, m \in L \quad (16k)
\]

\[
r_{1l} \leq \rho_l - \pi_l \leq D, \quad \forall l \in L \quad (16l)
\]
y_{ij} \geq 0, \quad \forall i \in N, \ j \in S_i^l, \ l \in L \quad (16m)

r_1^l, r_2^l \geq 0, \quad \forall l \in L. \quad (16n)

The big-M values in constraints (16g), (16h), and (16i) can be set to the optimal objective function value of the corresponding MSM instance. The next two propositions formally establish the equivalence of formulations (15) and (16).

**Proposition 1**  
A feasible solution to (16) does not contain any subtours.

**Proof** Let \((r^1, r^2, y, u, s, v, \rho, \pi)\) denote a feasible solution to (16). Suppose that there exists a subtour, \(T\), that does not contain the tour’s origin. Without loss of generality, assume that \(T\) contains arcs \(\{(1, 2), (2, 3), \ldots, (p, 1)\}\), and \(s_l = 0, \forall l = 1, \ldots, p\). Aggregating constraints (16h) for all arcs in \(T\) yields \(t_{12} + t_{23} + \cdots + t_{p1} + \sum_{l=1}^{p} r_2^l \leq 0\), which is a contradiction because all \(t\)-values are positive and all \(r^2\)-variables are nonnegative. This completes the proof. \(\square\)

**Proposition 2**  
Formulations (15) and (16) are equivalent.

**Proof** We prove that each solution to formulation (15) corresponds to a solution to formulation (16) having the same objective function value, and vice versa. First, let \(a = (r^1, r^2, y, u, s, v, \rho, \pi)\) denote a feasible solution to (16). Define \(r_l = r_1^l + r_2^l\), for all \(l \in L\). For \(l, m \in L\), let \(q_{lm} = \max\{0, \sum_{k \in N \setminus m} (\rho_k - \pi_m) u_{km}\}\) if \(s_l v_m (1 - u_{lm}) = 1\), and \(q_{lm} = 0\) otherwise. Solution \(b = (r, y, q, u, s, v)\) has the same objective function value for (15) as \(a\) does for (16). We next show that \(b\) is feasible to (15). Using the foregoing definitions and Proposition 1, \(b\) satisfies constraints (15b)–(15e), (15g), (15h), (8a), and (8b). Also, we compute the \(\alpha\)-, \(\beta\)-, and \(\lambda\)-values according to their desired linearizations, and hence the solution satisfies (10)–(14) and (15i). Hence, we must show that \(b\) satisfies (7), (9a), and (9b).

If \(q_{lm} = 0\) for \(l, m \in L\), then \(b\) satisfies (7) because \(t_{km} \leq D\) for all \(k \in N_m^-\). Now suppose that \(q_{lm} > 0\) for \(l, m \in L\). Then, by definition, we must have \(s_l = v_m = 1\) and \(u_{lm} = 0\). Let \(k \in N_m^- \setminus l\) be the location for which \(u_{km} = 1\). Note that

\[
D \geq \rho_k - \pi_m \quad \text{(by (16l))}
\]
\[
\geq \rho_k + r_2^k + t_{km} - \pi_m \quad \text{(by (16h), written for \((k,m)\))}
\]
\[
\geq \rho_k + t_{km} - \pi_m.
\]
Because $D \geq t_{km}$ as well, we conclude that $b$ satisfies (7) because

$$D \geq \max\{\rho_k - \pi_m, 0\} + t_{km} = q_{lm} + t_{km}.$$

We next prove that $b$ satisfies (9a) for $l, m \in L$. Note that if $v_m = 0$, then (9a) holds because $u_{km} = 0$ for all $k \in N_m$. Also, if $v_m = 1$ and $s_l = 0$, then because $l$ does not serve as the origin, we have that $q_{lk} = 0$, $\forall k \in L$, and (9a) holds because $r_k \geq 0$ for $k \in N_m$ and $t_{hk} \leq D$ for $k \in L$ and $h \in N_k$. Hence, we focus on the case in which $v_m = s_l = 1$. Suppose that $q_{lm} = 0$ and (9a) reduces to $0 \geq q_{ll} + t_{hl} - r_l$. If $q_{ll} = 0$, the inequality is satisfied because of (16k) written for $(l, h)$, and otherwise if $q_{ll} = \rho_l - \pi_l > 0$, then the inequality is satisfied because

$$r_l = r_l^1 + r_l^2 \geq \rho_l - \pi_l \geq (\rho_h + r_h^2 + t_{hl}) - \pi_l = q_{ll} + r_h^2 + t_{hl} \geq q_{ll} + t_{hl},$$

where the first two inequalities hold because of (16j) and (16h) (written for $(h, l)$), respectively. In case (ii), assume that $h = l$. Hence, $q_{lk} = 0$ and (9a) reduces to $q_{lm} \geq t_{lk} - r_k$. Using the definition of $q_{lm}$, (16i) (for $(k, m)$), and (16f), we have that

$$q_{lm} \geq \rho_k - \pi_m \geq \rho_k - r_k - \pi_k \geq t_{lk} - \pi_k - r_k.$$

Also, (16g) implies that $\pi_k = 0$, and hence (9a) holds. Finally, in case (iii), assume that $h \neq l$ and $k \neq l$. The following inequalities hold:

$$q_{lm} \geq \rho_k - \pi_m \geq \rho_k + r_h^2 + t_{hk} - \pi_m \geq q_{ll} + r_h^2 + t_{hl} \geq q_{ll} + t_{hl},$$

where the first two inequalities hold because of (16j) and (16h) (written for $(h, l)$), respectively. In case (ii), assume that $h = l$. Hence, $q_{lk} = 0$ and (9a) reduces to $q_{lm} \geq t_{lk} - r_k$. Using the definition of $q_{lm}$, (16i) (for $(k, m)$), and (16f), we have that

$$q_{lm} \geq \rho_k - \pi_m \geq \rho_k - r_k - \pi_k \geq t_{lk} - \pi_k - r_k.$$
\[
\begin{align*}
\geq \pi_h + r_h^1 + r_h^2 + t_{hk} - \pi_m \\
\geq \pi_k + t_{hk} - \pi_m \\
\geq \pi_m - r_k + t_{hk} - \pi_m \\
= t_{hk} - r_k.
\end{align*}
\] (18)

Using (17) and (18), we obtain

\[
q_{lm} \geq \max\{0, \rho_h - \pi_k\} + t_{hk} - r_k = q_{lk} + t_{hk} - r_k.
\]

Therefore \(b\) satisfies (9a).

We next prove that \(b\) satisfies (9b). If \(s_l = 0\), then (9b) holds because \(q_{ll} = 0, t_{ml} \leq D\), for all \(m \in N_l^-\), and \(r_l \geq 0\). Now suppose that \(l\) and \(k \in N_l^-\) are chosen such that \(s_l = u_{kl} = 1\). If \(q_{ll} = \rho_k - \pi_l\), then using (16j) and (16h) for \((k,l)\) we have

\[
r_l = r_l^1 + r_l^2 \geq \rho_l - \pi_l \geq \rho_k + r_k^2 + t_{kl} - \pi_l \geq \rho_k + t_{kl} - \pi_l.
\]

Hence, \(r_l \geq q_{ll} + t_{kl}\) and (9b) is satisfied. Otherwise we must have \(q_{ll} = 0\), and (16k) implies that \(r_k \geq t_{kl}\), which again implies that (9b) is satisfied.

Conversely, let \(b = (r, y, q, u, s, v, \rho, \pi)\) denote a feasible solution to (15). Let \(l\) and \(m\) be two sink locations for which \(s_l = u_{lm} = 1\). Define:

- \(r_k^1 = q_{lk} + \sum_{h \in N_k^-} t_{hk}u_{hk}\) and \(r_k^2 = r_k - r_k^1\), for \(k \in L\).
- \(\pi_m = 0\) and \(\rho_m = t_{lm}\).
- \(\pi_k = \pi_h + r_h\) if \(u_{hk} = 1\) and \(h \neq l\).
- \(\rho_k = \rho_h + r_h^2 + t_{hk}\) if \(u_{hk} = 1\) and \(h \neq l\).

Then \(a = (r^1, r^2, y, u, s, v, \rho, \pi)\) has the same objective function value as does \(b\) in (15), and satisfies (16b)–(16e) and (16n) due to the fact that \(a\) is feasible to (15). Solution \(b\) satisfies all remaining constraints to (16) by construction (the details of this analysis are straightforward and are omitted for brevity). \(\square\)
2.4 Comparison of MSM, MSM-FS1, and MSM-FS2

We conclude this section by stating the relationship between the optimal objective function values of the three lifetime maximization problems.

Let \((z, y)\) denote an optimal solution to (1). From this solution, we build a directed graph \(G' = (L', A')\) with \(L' = \{l \in L : z_l > 0\}\) as the node set and \(A' = \{(l, m) \in A : z_m \geq t_{lm}\}\) as the arc set. Figure 2 illustrates \(G'\) for the examples discussed in Section 2.1. Note that when the sink moves slower, \(G'\) becomes sparser (Figure 2(a)) and as we showed earlier, there exists no feasible trajectory for the sink. However, with a faster sink, \(G'\) is Hamiltonian and as we observed earlier, a feasible trajectory for the sink exists.

![Figure 2: Illustration of \(G'\) for the example in Section 2.1](image)

**Proposition 3** Let \(Z_{MSM}\), \(Z_{MSM-FS1}\), and \(Z_{MSM-FS2}\) denote the optimal objective function value of problems MSM, MSM-FS1, and MSM-FS2, respectively. Then

\[
Z_{MSM-FS1} \leq Z_{MSM-FS2} \leq Z_{MSM}. 
\] (19)

**Proof** Suppose that \((z, y, u, v)\) represents an optimal solution to MSM-FS1 and let \(r_l = z_l + \sum_{m \in N_l} t_{ml} u_{ml}\) and \(q_{lm} = 0\) for all \(l \in L\) and \(m \in L\). Also, suppose that \(s_l = 1\) for an arbitrary \(l\) on the tour that is represented by \((u, v)\), and \(s_m = 0, \forall m \in L \setminus \{l\}\). We prove that \(Z_{MSM-FS1} \leq Z_{MSM-FS2}\) by showing that \((r, y, q, u, v, s)\) is feasible to MSM-FS2. (The two solutions clearly provide the same objective function value.) Note that if \(u_{ml} = 1\), then \(t_{ml} \leq r_l\) and therefore (9b) holds for all \(l \in L\). The sink can only travel on an arc whose travel time is not more than \(D\), and so (7) is satisfied. Finally, constraints (8a), (8b), and (9a) hold by definition. Also, because MSM-FS2 is a restriction of MSM, we have that \(Z_{MSM-FS2} \leq Z_{MSM}\), which completes the proof. \(\square\)
Proposition 4 establishes a sufficient condition under which \( Z_{MSM-FS1} = Z_{MSM-FS2} = Z_{MSM} \).

**Proposition 4** \( Z_{MSM-FS1} = Z_{MSM-FS2} = Z_{MSM} \) if there exists an optimal solution to MSM for which the corresponding graph \( G' \) is Hamiltonian.

**Proof** Because \( Z_{MSM-FS1} \leq Z_{MSM-FS2} \leq Z_{MSM} \), we prove the claim by showing that an optimal solution to MSM satisfying the assumptions of the proposition corresponds to a feasible solution to MSM-FS1 having the same objective function value. Consider an optimal solution \((z, y)\) to MSM satisfying the assumptions of the proposition, and define \( T \) as the set of arcs in a Hamiltonian cycle in graph \( G' \) induced by \( z \). Let \( u^*_m = 1 \) if \((l, m)\) exists in \( T \), and set \( z^*_m = z_m - \sum_{l \in N^-_m} t_{lm} u^*_m \). By construction, \((u^*, z^*, y)\) is feasible to (2), and has the same objective function value as does the optimal MSM solution. This completes the proof. \( \square \)

### 3 Cutting-plane Algorithm for MSM-FS1

In this section, we focus on the development of a cutting-plane algorithm for solving MSM-FS1. In (2), we revise the definition of \( r_l \) as \( r_l = z_l + \sum_{m \in N^-_l} t_{ml} u_{ml} \), i.e., \( r_l \) denotes the sink’s dwelling time at location \( l \) plus the time that it takes for the sink to reach \( l \) from its previous sink location, and thus represents the total amount of time data is sent to location \( l \). Using this revised definition of \( r_l \), we can reformulate MSM-FS1 as follows.

\[
\text{Max } \sum_{l \in L} r_l \tag{20a}
\]

s.t. \( \sum_{l \in L} \left( \sum_{j \in S^+_l} e_{ij} y_{ij}^l + \sum_{j : i \in S^-_j} \gamma_j y_{ji}^l \right) \leq \bar{E}_i, \quad \forall i \in N \) \tag{20b}

\[
\sum_{j \in S^+_l} y_{ij}^l - \sum_{j : i \in S^-_j} y_{ji}^l = d_ir_l, \quad \forall i \in N, \ l \in L \tag{20c}
\]

\[
r_l \leq M_lv_l, \quad \forall l \in L \tag{20d}
\]

\[
r_m \geq \sum_{l \in N^+_l} t_{lm} u_{lm}, \quad \forall l \in L, \ m \in N^+_l \tag{20e}
\]

Constraints (2e)–(2i), and (2k)

\[
r_l \geq 0, \quad \forall l \in L. \tag{20g}
\]
Now consider the following master problem relaxation of (20).

Max \[ \sum_{l \in L} r_l \] (21a)

s.t. Constraints (20b)–(20d), (20g), and (2k) (21b)
\[ t_{lm} w_{lm} \leq r_m, \quad \forall l \in L, \ m \in N^+_l \] (21c)
\[ w_{lm} \leq v_l, \quad \forall l \in L, \ m \in N^+_l \] (21d)
\[ v_l \in \{0, 1\}, \quad \forall l \in L \] (21e)
\[ w_{lm} \in \{0, 1\}, \quad \forall l \in L, \ m \in N^+_l. \] (21f)

Here, auxiliary binary variable \( w_{lm} \) can equal one only when \( t_{lm} \leq r_m (\leq D) \) and \( v_l = v_m = 1 \). Because the delay constraints present in MSM-FS1 are absent in (21), the optimal objective function value to (21) matches that of MSM, and therefore provides an upper bound on the optimal objective function value of (20). Moreover, any feasible solution to (21) induces a directed graph \( G' = (L', A') \) with \( L' = \{l \in L : v_l = 1\} \) as its node set and \( A' = \{(l, m) : w_{lm} = 1\} \) as its arc set. (More precisely, after (21) is solved, one should postprocess the solution and set \( w_{lm} = 1 \) if permitted by (21c), for each \( l \in L \) and \( m \in N^+_l \).) By Proposition 4, if \( G' \) is Hamiltonian, then any optimal solution to (21) can be used to construct an optimal solution to (2). Let us define the following sets with respect to an optimal solution \((\mathbf{r}, \mathbf{y}, \mathbf{v}, \mathbf{w})\) to (21).

\[ \mathcal{V}^q = \{l \in L : v_l = q\}, \quad \text{for } q = 0, 1; \] (22)
\[ \mathcal{W}^1 = \{(l, m) : l \in L, \ m \in N^+_l, \ \text{and } \mathbf{w}_{lm} = 1\} \] (23)
\[ \mathcal{W}^0 = \{(l, m) : l \in L, \ m \in N^+_l, \ \text{and } \mathbf{w}_{lm} = 0\} \cup \{(l, m) : l \in L, \ m \in L \setminus N^+_l\}. \] (24)

The following subproblem seeks a Hamiltonian cycle over \( \mathcal{V}^1 \), where edge \((l, m)\) has an objective function cost coefficient of \( d_{lm} = 0 \) if \((l, m) \in \mathcal{W}^1 \), and a cost of \( d_{lm} = 1 \) if \((l, m) \in \mathcal{W}^0 \) (including the case when \( m \notin N^+_l \)). If the optimal objective function value to subproblem (25) below equals 0, then any of its optimal solutions represents a Hamiltonian cycle in \( G' \), which in turn indicates that the current optimal solution to the master problem corresponds to an optimal solution to MSM-FS1.

\[ \text{Min } \sum_{l \in \mathcal{V}^1} \sum_{m \in \mathcal{V}^1} d_{lm} x_{lm} \] (25a)
\[
\text{s.t. } \sum_{m \in V^1} x_{ml} = 1, \quad \forall l \in V^1 \quad (25b)
\]
\[
\sum_{m \in V^1} x_{lm} = 1, \quad \forall l \in V^1 \quad (25c)
\]
\[
\sum_{l \in S} \sum_{m \in S} x_{lm} \leq |S| - 1, \quad \forall S \subset V^1, \; 2 \leq |S| \leq |V^1| - 2 \quad (25d)
\]
\[
x_{lm} \in \{0, 1\}, \quad \forall l \in V^1, \; m \in V^1. \quad (25e)
\]

Note that (25) is an instance of the Traveling Salesman Problem (TSP) [2]. Let \( n^* \) denote the optimal objective function value of (25). If \( n^* \) is positive, then any Hamiltonian cycle in \( G' \) (if one exists) uses at least \( n^* \) arcs from \( W^0 \), and we must add an appropriate cutting plane to the master problem (21) to ensure the existence of a Hamiltonian cycle in \( G' \) either by modifying \( r \)-values (which possibly modifies \( W^0 \) and \( W^1 \)) or the composition of \( V^1 \). We next describe our method of generating cutting planes for the master problem.

Let \((r, y, v, u)\) be a feasible solution to (20); this solution is also feasible to (21) by letting \( w_{lm} = u_{lm} \). The following proposition presents a class of cutting planes for the master problem (21) that are valid for all feasible solutions of (20).

**Proposition 5** Suppose that the optimal objective function value of the subproblem (25) given \( v \) and \( w \) is \( n^* > 0 \). The following inequality for (21) cuts off the solution containing \( v \) and \( w \), and is valid for all feasible solutions to (20):

\[
2 \sum_{l \in V^1} (1 - v_l) + \sum_{(l,m) \in W^0} w_{lm} \geq n^* + \sum_{l \in V^0} v_l. \quad (26)
\]

**Proof** First note that (26) cuts off \((v, w, W)\) when \( n^* > 0 \), because the left-hand-side of (26) evaluates to 0 for this solution while the right-hand-side equals \( n^* \). We say that location \( l \) is active if \( v_l = 1 \), and is inactive otherwise. Suppose that a feasible solution \((\bar{r}, \bar{y}, \bar{v}, \bar{u})\) to (20) induces a Hamiltonian cycle \( H \) in which there exists \( p \geq 0 \) inactive sink locations in \( V^1 \), and \( r \geq 0 \) active sink locations in \( V^0 \). Note that if \( p = |V^1| \), then (26) reduces to

\[
\sum_{(l,m) \in W^0} w_{lm} \geq n^* - 2|V^1| + r,
\]

which is valid because \( n^* \leq |V^1| \) and \( \sum_{(l,m) \in W^0} w_{lm} = r \). (The equality holds because in this case \( H \) contains exactly \( r \) arcs with both endpoints in \( V^0 \).) Therefore, we will assume in the remainder
of this proof that \( p \leq |V^1| - 1 \), i.e., \( H \) visits at least one sink location from \( V^1 \).

Suppose by contradiction that the solution associated with \( H \) violates (26). Then we must have

\[
2p + \sum_{(l,m) \in W^0} \hat{u}_{lm} \leq n^* - 1 + r. \tag{27}
\]

Hence, \( H \) contains at most \( n^* - 1 + r - 2p \) arcs from \( W^0 \). Index these nodes as \( i^1, j^1_1, \ldots, j^1_{q_1}, \ldots, i^t, j^t_1, \ldots, j^t_{q_t}, i^1 \), where \( i^1, i^2, \ldots, i^t \) belong to \( V^1 \) and all other nodes belong to \( V^0 \). (If \( q_e = 0 \) for some \( e = 1, \ldots, t \), then the tour proceeds from \( i^e \) to \( i^{e+1} \), where \( i^{e+1} \equiv i^1 \).) We build a new cycle \( H' \) by first removing all nodes in \( V^0 \) from \( H \), so that \( H' \) initially consists of \( i^1, i^2, \ldots, i^t, i^1 \) (as shown in Figure 3). Note that in removing nodes \( j^e_1, \ldots, j^e_{q_e} \) from \( H \), for some \( e \in \{1, \ldots, t\} \) with \( q_e \geq 1 \), a total of \( q_e + 1 \) arcs in \( W^0 \) are removed from \( H' \) while adding at most one arc \((i^e, i^{e+1})\) in \( W^0 \) to \( H' \). Hence, \( H' \) now contains at most \( n^* - 1 - 2p \) arcs in \( W^0 \). Next, suppose that we expand \( H' \) to include all \( p \) nodes in \( V^1 \) that were not contained in \( H \). These \( p \) nodes can be inserted to \( H' \) in any arbitrary order, with each insertion requiring at most two inactive arcs. Now, \( H' \) consists of at most \( n^* - 1 \) arcs in \( W^0 \), which contradicts the assumption that the optimal objective function value of (25) is \( n^* \). This completes the proof. \( \square \)

**Remark 2.** The last step in the proof of Proposition 5 can alternatively be stated in the following way: If \( p > 0 \), include all sink locations \( i^{t+1}, \ldots, i^{t+p} \) in \( V^1 \) not present in \( H' \) (and \( H \)) by simply extending \( H' \) to visit all locations in the order \( i^1, i^2, \ldots, i^{t+p}, i^1 \), which in the worst case adds \( p + 1 \) more arcs in \( W^0 \) to \( H' \). Then, \( H' \) spans \( V^1 \) and contains at most \( n^* - p \) arcs in \( W^0 \) when \( p > 0 \), and at most \( n^* - 1 \) arcs in \( W^0 \) when \( p = 0 \). This suggests the following alternative valid inequality for master problem (21):

\[
1 + \sum_{l \in V^1}(1 - v_l) + \sum_{(l,m) \in W^0} w_{lm} \geq n^* + \sum_{l \in V^0} v_l. \tag{28}
\]

Note that while (28) is valid and cuts off solutions that are not cut off by (26), it does not cut off an infeasible master problem solution when the subproblem yields an optimal objective function value of \( n^* = 1 \). However, we can add both (26) and (28) to the master problem in our cutting-plane scheme, or add (26) when \( n^* = 1 \) and (28) otherwise. A brief computational study reveals that the use of (26) by itself in the cutting-plane algorithm is the most effective implementation. \( \square \)

Now, we present a method that permits us to set certain \( w \)-values equal to zero in a preprocessing phase. Let \( Z_{LB} \) denote a lower bound on the optimal objective function value of (2), e.g., as
computed via the objective function value of a feasible solution to (2). To determine if a given arc \((l, m) \in A\) could be traversed by the sink in an optimal solution to (2), we check a necessary condition for it to be used in at least one feasible solution having an objective function value that is no less than \(Z_{LB}\). Specifically, if

\[
\left( \sum_{i \in N} d_i \right) \left( \min_{i : m \in S_i} \{e_{im}^m\} \right) t_{lm} + \left( \sum_{i \in N \setminus \{i : m \in S_i^m\}} d_i \right) \gamma t_{lm} + (Z_{LB} - t_{lm}) \sum_{i : m \in S_i^m} \left( d_i \min_{l \in L, j \in S_i^l} \{e_{ij}^l\} \right) > \sum_{i : m \in S_i^m} E_i, \tag{29}
\]

then \((l, m)\) cannot be included in any optimal solution to (2) and hence \(w_{lm} = 0\) is valid for (21).

The first two terms in (29) provide a lower bound on the total energy expended by sensor nodes that can communicate with the sink at location \(m\) during the sink’s travel on arc \((l, m)\), while the third term is a lower bound on the combined energy expended by these sensors during the rest of the network lifetime.

**Remark 3.** We can also use the information from an optimal solution to MSM to add valid inequalities to MSM-FS1. Let \(L' \subset L\) and suppose that \(Z_{TSP(L')} > Z_{MSM}\), where \(Z_{TSP(L')}\) is the minimum TSP tour length over the sink locations in \(L'\). Then, any TSP tour over a subset of \(L\) that visits all sink locations in \(L'\) is not associated with a feasible to (2) because any such solution would have an objective function value greater than \(Z_{MSM}\). Therefore,

\[
\sum_{l \in L'} v_l \leq |L'| - 1 \tag{30}
\]
is valid for (2). In particular, if \( t_{lm} > \frac{Z_{\text{MSM}}}{2} \), then \( u_{lm} = 0 \) is valid for (2). □

### 4 Benders Decomposition

The cutting-plane algorithm presented in Section 3 for the MSM-FS1 starts from a relaxation of the original lifetime maximization problem whose optimal solution may not admit a Hamiltonian cycle of the active sink locations. The algorithm then induces the existence of a Hamiltonian cycle by iteratively adding inequalities of form (26) to the master problem. In this section, we propose a decomposition approach that starts from a tour over a subset of the sink locations and reaches an optimal solution by adding appropriate Benders cuts [6]. Our decomposition approach makes the sink’s routing decisions in the following master problem.

\[
\text{Max } \sum_{l \in L} \sum_{m \in N_l^+} t_{ml} u_{ml} + \theta \tag{31a}
\]

s.t. \( \sum_{m \in N_m^-} u_{ml} = v_l, \quad \forall l \in L \tag{31b} \)

\( \sum_{m \in N_m^-} u_{ml} = \sum_{m \in N_i^+} u_{lm}, \quad \forall l \in L \tag{31c} \)

\( \sum_{l \in S} \sum_{m \in S} u_{lm} \geq v_k + v_r - 1, \quad \forall S \subset L, k \in S, r \in \overline{S} \tag{31d} \)

\( v_l \in \{0, 1\}, \quad \forall l \in L \tag{31e} \)

\( u_{lm} \in \{0, 1\}, \quad \forall l \in L, \; m \in N_l^+, \tag{31f} \)

where \( \theta \) is defined as follows, given values \( \mathbf{u} = \overline{\mathbf{u}} \) and \( \mathbf{v} = \overline{\mathbf{v}} \)

\[
\theta = \text{Max } \sum_{l \in L} z_l \tag{32a}
\]

s.t. \( \sum_{l \in L} \left( \sum_{j \in S_i^l} e_{ij}^l y_{ij}^l + \sum_{j \in S_j^l} \gamma y_{ji}^l \right) \leq E_i, \quad \forall i \in N \tag{32b} \)

\( \sum_{j \in S_i^l} y_{ij}^l - \sum_{j \in S_j^l} y_{ji}^l = d_i \left( z_l + \sum_{m \in N_l^-} t_{ml} \overline{u}_{ml} \right), \quad \forall i \in N, \; l \in L \tag{32c} \)

\( z_l \leq M_l \overline{u}_l, \quad \forall l \in L \tag{32d} \)

\( z_l \geq 0, \quad \forall l \in L \tag{32e} \)
\[ y_{ij}^l \geq 0, \quad \forall i \in N, \ j \in S_i^l, \ l \in L. \]  

(32f)

Problems (31) and (32) together provide an equivalent reformulation for (2). Let \( f_i, g_{il}, \) and \( h_l \) denote the dual variables associated with constraints (32b), (32c), and (32d), respectively. Taking the dual of (32), we get

\[
\begin{align*}
\theta &= \min \sum_{i \in N} \bar{E}_i f_i + \sum_{i \in N} \sum_{l \in L} d_i \left( \sum_{m \in N_i^-} t_{ml} \bar{u}_{ml} \right) g_{il} + \sum_{l \in L} M_l \bar{u}_l h_l \\
\text{s.t.} \quad &\sum_{i \in N} -d_i g_{il} + h_l \geq 1, \quad \forall l \in L \\
&g_{il} - g_{jl} + e_{ij} f_i + \gamma f_j \geq 0, \quad \forall i \in N, \ j \in S_i \setminus \{s\}, \ l \in L \\
&g_{il} + e_{is} f_i \geq 0, \quad \forall i \in N, \ s \in S_i^l, \ l \in L \\
&h_l \geq 0, \quad \forall l \in L \\
&f_i \geq 0, \quad \forall i \in N.
\end{align*}
\]

(33a) - (33f)

If problem (33) (which serves as the Benders dual subproblem) has an unbounded direction \((\tilde{f}, \tilde{g}, \tilde{h})\), we add the following feasibility cut to the Benders master problem (31).

\[
\sum_{i \in N} \bar{E}_i \tilde{f}_i + \sum_{i \in N} \sum_{l \in L} \sum_{m \in N_i^-} (d_i t_{ml} \tilde{g}_{il}) u_{ml} + \sum_{l \in L} (M_l \tilde{u}_l) v_l \geq 0.
\]

(34)

When (33) has an optimal solution \((f^*, g^*, h^*)\), we add the following optimality cut to the master problem. Note that (33) is never infeasible, because setting \( h_l = 1 \), for all \( l \in L \), and all other variables equal to zero, is always feasible.

\[
\theta \leq \sum_{i \in N} \bar{E}_i f_i^* + \sum_{i \in N} \sum_{l \in L} \sum_{m \in N_i^-} (d_i t_{ml} g_{il}^*) u_{ml} + \sum_{l \in L} (M_l h_l^*) v_l.
\]

(35)

5 Computational Results

We examine the computational effectiveness of the proposed algorithms in this paper on a collection of randomly generated instances. We generated these instances by randomly placing sensor nodes and sink locations in a circular area of radius 25. A sensor node can communicate with sensor nodes and sink locations in a disk of radius 15 around it. We assume an initial energy and fixed
data generation rate of $E_i = 500$ and $d_i = 50$ for all sensor nodes. To calculate the required energy for transmission of a unit of data along the link $(i, j)$, we use an energy model similar to that of [25]. More specifically, we use $e_{ij} = 0.0005 + 0.00013\mu_{ij}^4$ and $\gamma = 0.0005$, where $\mu_{ij}$ denotes the distance between two nodes $i$ and $j$ that can communicate with one another while the sink is at a sink location $l \in L$.

We have implemented all mixed-integer programs in C++ calling CPLEX version 11.0 via ILOG Concert Technology 2.5. All codes are compiled using Microsoft Visual C++ 2008 and the computational experiments are carried out on a PC having an Intel Core2 Quad processor Q9500 and 4 GB of memory, running Windows 7. A 1000-second time limit is imposed on all running times.

We first provide a comparison between solving MSM-FS1 by the cutting-plane algorithm of Section 3 and by solving problem (2) directly using CPLEX. For this experiment, we generate random instances of two different sizes ($|N|, |L|) = (50, 15)$ and $(|N|, |L|) = (60, 20)$ as well as two different levels $D = 0.2$ and $D = 0.4$ for the maximum tolerable delay. For each combination of problem size and delay tolerance level, ten random networks are generated. We then generate two instances of MSM-FS1 associated with the two different sink speeds, and solve these instances both directly via CPLEX (which we call “CPLEX” in our computational tables), and also by the cutting-plane algorithm in Section 3. Our CPLEX implementation for MSM-FS1 uses a special callback function inside the branch-and-bound tree to add the violated subtour elimination constraints at each node. This callback function is an implementation of the following separation procedure [10], where $\overline{u}_{lm}$ and $\overline{v}_l$ denote current values of variables $u_{lm}$ and $v_l$, respectively.

- Build a complete directed graph $G$ having $L$ as its node set, and fix the capacity of each arc $(l, m)$ in $G$ to $\overline{u}_{lm}$.
- For every pair of sink locations $l$ and $m$ such that $\overline{v}_l + \overline{v}_m > 1$, solve a maximum flow problem from $l$ and $m$ on $G$.
- If the capacity of the resulting minimum cut $(S, \overline{S})$ is less than $\overline{v}_l + \overline{v}_m - 1$, that is, if $\sum_{l \in S} \sum_{m \in \overline{S}} \overline{u}_{lm} < \overline{v}_k + \overline{v}_r - 1$, then add the violated inequality to the linear programming relaxation at the current node.

Valid inequalities (26) are also added to the master problem (21) using a callback function. We add these inequalities only at branch-and-bound nodes having integer solutions. We do not include inequalities based on (29) in our preprocessing step because efficient deployment of these inequalities
Table 1: Comparison of the direct solve and cutting-plane algorithms for MSM-FS1

<table>
<thead>
<tr>
<th></th>
<th>CPLEX $v = 25$</th>
<th>CPLEX $v = 50$</th>
<th>Cutting-Plane</th>
<th>Cutting-Plane</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time</td>
<td>Subtours</td>
<td>Time</td>
<td>Cuts</td>
</tr>
<tr>
<td>MSM-FS1-50-15-01</td>
<td>16.6</td>
<td>8</td>
<td>14.5</td>
<td>1</td>
</tr>
<tr>
<td>MSM-FS1-50-15-02</td>
<td>7.0</td>
<td>8</td>
<td>14.3</td>
<td>1</td>
</tr>
<tr>
<td>MSM-FS1-50-15-03</td>
<td>0.6</td>
<td>0</td>
<td>0.8</td>
<td>0</td>
</tr>
<tr>
<td>MSM-FS1-50-15-04</td>
<td>0.3</td>
<td>0</td>
<td>0.3</td>
<td>0</td>
</tr>
<tr>
<td>MSM-FS1-50-15-05</td>
<td>5.6</td>
<td>16</td>
<td>3.4</td>
<td>2</td>
</tr>
<tr>
<td>MSM-FS1-50-15-06</td>
<td>0.9</td>
<td>0</td>
<td>0.9</td>
<td>0</td>
</tr>
<tr>
<td>MSM-FS1-50-15-07</td>
<td>4.7</td>
<td>8</td>
<td>5.2</td>
<td>1</td>
</tr>
<tr>
<td>MSM-FS1-50-15-08</td>
<td>0.6</td>
<td>0</td>
<td>0.9</td>
<td>0</td>
</tr>
<tr>
<td>MSM-FS1-50-15-09</td>
<td>0.5</td>
<td>0</td>
<td>0.5</td>
<td>0</td>
</tr>
<tr>
<td>MSM-FS1-50-15-10</td>
<td>3.8</td>
<td>8</td>
<td>3.1</td>
<td>1</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td>4.1</td>
<td>4.8</td>
<td>4.4</td>
<td>0.6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>CPLEX</th>
<th>CPLEX</th>
<th>Cutting-Plane</th>
<th>Cutting-Plane</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time</td>
<td>Subtours</td>
<td>Time</td>
<td>Cuts</td>
</tr>
<tr>
<td>MSM-FS1-60-20-01</td>
<td>18.5</td>
<td>4</td>
<td>66.5</td>
<td>17</td>
</tr>
<tr>
<td>MSM-FS1-60-20-02</td>
<td>2.9</td>
<td>0</td>
<td>3.4</td>
<td>0</td>
</tr>
<tr>
<td>MSM-FS1-60-20-03</td>
<td>19.6</td>
<td>32</td>
<td>24.4</td>
<td>5</td>
</tr>
<tr>
<td>MSM-FS1-60-20-04</td>
<td>31.2</td>
<td>36</td>
<td>33.1</td>
<td>8</td>
</tr>
<tr>
<td>MSM-FS1-60-20-05</td>
<td>2.4</td>
<td>0</td>
<td>8.2</td>
<td>0</td>
</tr>
<tr>
<td>MSM-FS1-60-20-06</td>
<td>25.8</td>
<td>16</td>
<td>23.4</td>
<td>2</td>
</tr>
<tr>
<td>MSM-FS1-60-20-07</td>
<td>7.2</td>
<td>0</td>
<td>20.7</td>
<td>2</td>
</tr>
<tr>
<td>MSM-FS1-60-20-08</td>
<td>47.4</td>
<td>48</td>
<td>77.0</td>
<td>59</td>
</tr>
<tr>
<td>MSM-FS1-60-20-09</td>
<td>223.5</td>
<td>52</td>
<td>355.8</td>
<td>31</td>
</tr>
<tr>
<td>MSM-FS1-60-20-10</td>
<td>25.3</td>
<td>56</td>
<td>21.4</td>
<td>14</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td>40.4</td>
<td>28.4</td>
<td>63.4</td>
<td>13.8</td>
</tr>
</tbody>
</table>

Table 3 presents the results of running the Benders decomposition algorithm of Section 4 for MSM-FS1 versus a direct solve by CPLEX for two different sink speeds and two problem sizes $(|N|, |L|) = (50, 15)$ and $(|N|, |L|) = (60, 20)$. Here, we assume a maximum tolerable delay of $D = 0.2$ time units. All cutting planes, including the subtour elimination and Benders feasibility and optimality cuts, are added using a callback routine. The Benders feasibility and optimality cuts are added only at branch-and-bound nodes having integer solutions. The results clearly favor the use of Benders decomposition for both sink speeds, as the average time for Benders decomposition is significantly shorter than that of CPLEX. Table 4 provides a comparison of the performance of the two algorithms for the same instances and the same sink speeds but with an increased maximum tolerable delay of $D = 0.4$ time units for all instances. In this case, the results for $v = 25$ still suggest using the Benders decomposition algorithm. However, results for instances with
Table 2: Performance of the cutting-plane algorithm for MSM-FS1 under increased delay tolerance

<table>
<thead>
<tr>
<th></th>
<th>CPLEX</th>
<th>Cutting-Plane</th>
<th></th>
<th>CPLEX</th>
<th>Cutting-Plane</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time</td>
<td>Subtours</td>
<td>Time</td>
<td>Cuts</td>
<td>Time</td>
</tr>
<tr>
<td>MSM-FS1-50-15-01</td>
<td>4.6</td>
<td>52</td>
<td>25.0</td>
<td>51</td>
<td>7.5</td>
</tr>
<tr>
<td>MSM-FS1-50-15-02</td>
<td>12.3</td>
<td>68</td>
<td>17.6</td>
<td>19</td>
<td>5.4</td>
</tr>
<tr>
<td>MSM-FS1-50-15-03</td>
<td>1.5</td>
<td>0</td>
<td>2.6</td>
<td>0</td>
<td>1.8</td>
</tr>
<tr>
<td>MSM-FS1-50-15-04</td>
<td>23.7</td>
<td>80</td>
<td>89.2</td>
<td>27</td>
<td>11.1</td>
</tr>
<tr>
<td>MSM-FS1-50-15-05</td>
<td>25.4</td>
<td>56</td>
<td>137.2</td>
<td>57</td>
<td>19.1</td>
</tr>
<tr>
<td>MSM-FS1-50-15-06</td>
<td>5.5</td>
<td>16</td>
<td>7.4</td>
<td>5</td>
<td>1.6</td>
</tr>
<tr>
<td>MSM-FS1-50-15-07</td>
<td>16.7</td>
<td>56</td>
<td>30.5</td>
<td>42</td>
<td>23.5</td>
</tr>
<tr>
<td>MSM-FS1-50-15-08</td>
<td>1.6</td>
<td>0</td>
<td>1.3</td>
<td>0</td>
<td>32.4</td>
</tr>
<tr>
<td>MSM-FS1-50-15-09</td>
<td>5.0</td>
<td>32</td>
<td>15.8</td>
<td>9</td>
<td>15.7</td>
</tr>
<tr>
<td>MSM-FS1-50-15-10</td>
<td>22.8</td>
<td>94</td>
<td>343.9</td>
<td>134</td>
<td>20.8</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td><strong>11.9</strong></td>
<td><strong>45.4</strong></td>
<td><strong>67.1</strong></td>
<td><strong>34.4</strong></td>
<td><strong>13.9</strong></td>
</tr>
</tbody>
</table>

|                  | CPLEX     | Cutting-Plane |                  | CPLEX     | Cutting-Plane |
| MSM-FS1-60-20-01 | 44.1      | 250           | > 1000.0         | > 1030    | 25.2         | 208           | > 1000.0        | > 1021    |
| MSM-FS1-60-20-02 | 3.6       | 8             | 3.3              | 0         | 4.6          | 14            | 10.4            | 0         |
| MSM-FS1-60-20-03 | 20.3      | 92            | > 1000.0         | > 366     | 133.1        | 468           | > 1000.0        | > 366     |
| MSM-FS1-60-20-04 | 52.4      | 166           | > 1000.0         | > 280     | 30.1         | 278           | > 1000.0        | > 1046    |
| MSM-FS1-60-20-05 | 154.0     | 170           | > 1000.0         | > 128     | 22.4         | 150           | 46.7            | 6         |
| MSM-FS1-60-20-06 | 47.4      | 166           | 292.7            | 108       | 66.7         | 688           | > 1000.0        | > 544     |
| MSM-FS1-60-20-07 | 39.4      | 276           | > 1000.0         | > 682     | 37.7         | 310           | 138.8           | 101       |
| MSM-FS1-60-20-08 | 18.4      | 48            | > 1000.0         | > 45      | 18.3         | 112           | 382.8           | 90        |
| MSM-FS1-60-20-09 | 326.9     | 168           | 998.4            | 78        | 355.3        | 658           | > 1000.0        | > 362     |
| MSM-FS1-60-20-10 | 73.5      | 518           | > 1000.0         | > 462     | 54.7         | 820           | > 1000.0        | > 216     |
| **Average**      | **78.0**  | **186.2**     | **> 829.6**      | **> 317.9** | **74.8**     | **370.6**     | **> 657.9**     | **> 375.2** |

$v = 50$ suggest using a direct solve by CPLEX rather than the Benders decomposition algorithm. The combination of lower travel times due to increased sink speed and the increase in maximum tolerable delay makes the underlying sink graphs for $v = 50$ denser than their counterparts for lower sink speeds and shorter maximum tolerable delay levels. We conclude that a direct solve of (2) by CPLEX tends to outperform Benders decomposition on instances having relatively dense sink graphs. One explanation for the effectiveness of Benders decomposition on sparse sink graphs is that sparse graphs contain fewer cycles, which in turn requires fewer Benders decomposition iterations.

We conclude this section by providing a comparison between the running times of formulations (15) and (16) in solving MSM-FS2 in Table 5. These results are obtained by solving MSM-FS2 on the same networks used in Table 4. The maximum tolerable delay is set to $D = 0.4$. The subtour elimination constraints in (15) are initially relaxed and added inside a callback function using the foregoing separation procedure. In addition to the running times for both formulations (in CPU seconds), we report the number of subtour elimination constraints added to (15) for each instance. These results suggest that solving (15) is generally faster, despite the fact that it is not polynomially sized. Also, the results suggest that MSM-FS2 tends to be more challenging and computationally complex than MSM-FS1.
### Table 3: Comparison of the direct solve and Benders decomposition algorithms for MSM-FS1

<table>
<thead>
<tr>
<th></th>
<th>CPLEX</th>
<th>Benders</th>
<th>CPLEX</th>
<th>Benders</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time</td>
<td>Subtours</td>
<td>Time</td>
<td>Subtours</td>
</tr>
<tr>
<td>MSM-FS1-50-15-1</td>
<td>16.6</td>
<td>8</td>
<td>2.0</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSM-FS1-50-15-2</td>
<td>7.0</td>
<td>0</td>
<td>2.3</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSM-FS1-50-15-3</td>
<td>0.6</td>
<td>0</td>
<td>3.3</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSM-FS1-50-15-4</td>
<td>0.3</td>
<td>1</td>
<td>1.7</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSM-FS1-50-15-5</td>
<td>5.6</td>
<td>16</td>
<td>2.0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSM-FS1-50-15-6</td>
<td>0.9</td>
<td>0</td>
<td>3.5</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSM-FS1-50-15-7</td>
<td>4.7</td>
<td>8</td>
<td>1.7</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSM-FS1-50-15-8</td>
<td>0.6</td>
<td>0</td>
<td>1.3</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSM-FS1-50-15-9</td>
<td>0.5</td>
<td>0</td>
<td>1.7</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSM-FS1-50-15-10</td>
<td>3.8</td>
<td>8</td>
<td>1.9</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>4.1</td>
<td>4.8</td>
<td>2.1</td>
<td>1.7</td>
</tr>
</tbody>
</table>

### Table 4: Performance of the Benders decomposition algorithm under increased delay tolerance

<table>
<thead>
<tr>
<th></th>
<th>CPLEX</th>
<th>Benders</th>
<th>CPLEX</th>
<th>Benders</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time</td>
<td>Subtours</td>
<td>Time</td>
<td>Subtours</td>
</tr>
<tr>
<td>MSM-FS1-50-15-01</td>
<td>4.6</td>
<td>52</td>
<td>3.07</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSM-FS1-50-15-02</td>
<td>12.3</td>
<td>68</td>
<td>3.057</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSM-FS1-50-15-03</td>
<td>1.5</td>
<td>0</td>
<td>4.352</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSM-FS1-50-15-04</td>
<td>23.7</td>
<td>80</td>
<td>3.213</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSM-FS1-50-15-05</td>
<td>25.4</td>
<td>56</td>
<td>2.888</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSM-FS1-50-15-06</td>
<td>5.5</td>
<td>16</td>
<td>4.322</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSM-FS1-50-15-07</td>
<td>16.7</td>
<td>56</td>
<td>2.714</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSM-FS1-50-15-08</td>
<td>1.6</td>
<td>0</td>
<td>2.121</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSM-FS1-50-15-09</td>
<td>5.0</td>
<td>32</td>
<td>2.668</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSM-FS1-50-15-10</td>
<td>22.8</td>
<td>94</td>
<td>3.556</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>11.9</td>
<td>45.4</td>
<td>3.2</td>
<td>6.9</td>
</tr>
</tbody>
</table>

### Table 4: Performance of the Benders decomposition algorithm under increased delay tolerance

<table>
<thead>
<tr>
<th></th>
<th>CPLEX</th>
<th>Benders</th>
<th>CPLEX</th>
<th>Benders</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time</td>
<td>Subtours</td>
<td>Time</td>
<td>Subtours</td>
</tr>
<tr>
<td>MSM-FS1-50-15-01</td>
<td>4.6</td>
<td>52</td>
<td>3.07</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSM-FS1-50-15-02</td>
<td>12.3</td>
<td>68</td>
<td>3.057</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSM-FS1-50-15-03</td>
<td>1.5</td>
<td>0</td>
<td>4.352</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSM-FS1-50-15-04</td>
<td>23.7</td>
<td>80</td>
<td>3.213</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSM-FS1-50-15-05</td>
<td>25.4</td>
<td>56</td>
<td>2.888</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSM-FS1-50-15-06</td>
<td>5.5</td>
<td>16</td>
<td>4.322</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSM-FS1-50-15-07</td>
<td>16.7</td>
<td>56</td>
<td>2.714</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSM-FS1-50-15-08</td>
<td>1.6</td>
<td>0</td>
<td>2.121</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSM-FS1-50-15-09</td>
<td>5.0</td>
<td>32</td>
<td>2.668</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSM-FS1-50-15-10</td>
<td>22.8</td>
<td>94</td>
<td>3.556</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>11.9</td>
<td>45.4</td>
<td>3.2</td>
<td>6.9</td>
</tr>
</tbody>
</table>

NA: Instance not solved due to memory limitations.
Table 5: Comparison of solution times for MSM-FS2 models (15) and (16) $v = 25$

<table>
<thead>
<tr>
<th></th>
<th>Formulation (15)</th>
<th>Formulation (16)</th>
<th>Formulation (15)</th>
<th>Formulation (16)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time Subtours</td>
<td>Time</td>
<td>Time Subtours</td>
<td>Time</td>
</tr>
<tr>
<td>MSM-FS2-50-15-01</td>
<td>5.3</td>
<td>50</td>
<td>8.8</td>
<td>7.1</td>
</tr>
<tr>
<td>MSM-FS2-50-15-02</td>
<td>8.9</td>
<td>80</td>
<td>16.5</td>
<td>30.0</td>
</tr>
<tr>
<td>MSM-FS2-50-15-03</td>
<td>3.3</td>
<td>0</td>
<td>1.3</td>
<td>3.9</td>
</tr>
<tr>
<td>MSM-FS2-50-15-04</td>
<td>19.8</td>
<td>108</td>
<td>21.0</td>
<td>&gt;1000.0</td>
</tr>
<tr>
<td>MSM-FS2-50-15-05</td>
<td>21.5</td>
<td>78</td>
<td>46.7</td>
<td>&gt;1000.0</td>
</tr>
<tr>
<td>MSM-FS2-50-15-06</td>
<td>1.7</td>
<td>0</td>
<td>2.0</td>
<td>1.8</td>
</tr>
<tr>
<td>MSM-FS2-50-15-07</td>
<td>13.7</td>
<td>84</td>
<td>24.5</td>
<td>35.7</td>
</tr>
<tr>
<td>MSM-FS2-50-15-08</td>
<td>1.6</td>
<td>18</td>
<td>1.6</td>
<td>21.1</td>
</tr>
<tr>
<td>MSM-FS2-50-15-09</td>
<td>6.8</td>
<td>32</td>
<td>16.7</td>
<td>&gt;1000.0</td>
</tr>
<tr>
<td>MSM-FS2-50-15-10</td>
<td>32.9</td>
<td>120</td>
<td>138.8</td>
<td>&gt;1000.0</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td><strong>11.5</strong></td>
<td><strong>57.0</strong></td>
<td><strong>27.8</strong></td>
<td><strong>410.0</strong></td>
</tr>
<tr>
<td>MSM-FS2-60-20-01</td>
<td>61.6</td>
<td>258</td>
<td>815.8</td>
<td>&gt;1000.0</td>
</tr>
<tr>
<td>MSM-FS2-60-20-02</td>
<td>4.9</td>
<td>0</td>
<td>3.3</td>
<td>2.9</td>
</tr>
<tr>
<td>MSM-FS2-60-20-03</td>
<td>35.7</td>
<td>104</td>
<td>493.3</td>
<td>&gt;1000.0</td>
</tr>
<tr>
<td>MSM-FS2-60-20-04</td>
<td>60.8</td>
<td>202</td>
<td>522.8</td>
<td>NA</td>
</tr>
<tr>
<td>MSM-FS2-60-20-05</td>
<td>96.1</td>
<td>174</td>
<td>&gt;1000.0</td>
<td>&gt;1000.0</td>
</tr>
<tr>
<td>MSM-FS2-60-20-06</td>
<td>26.7</td>
<td>86</td>
<td>117.7</td>
<td>&gt;1000.0</td>
</tr>
<tr>
<td>MSM-FS2-60-20-07</td>
<td>54.0</td>
<td>398</td>
<td>179.7</td>
<td>&gt;1000.0</td>
</tr>
<tr>
<td>MSM-FS2-60-20-08</td>
<td>18.7</td>
<td>94</td>
<td>126.6</td>
<td>&gt;1000.0</td>
</tr>
<tr>
<td>MSM-FS2-60-20-09</td>
<td>408.2</td>
<td>194</td>
<td>634.5</td>
<td>NA</td>
</tr>
<tr>
<td>MSM-FS2-60-20-10</td>
<td>45.0</td>
<td>378</td>
<td>&gt;1000.0</td>
<td>&gt;1000.0</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td><strong>81.2</strong></td>
<td><strong>188.8</strong></td>
<td><strong>489.4</strong></td>
<td><strong>NA</strong></td>
</tr>
</tbody>
</table>

### 6 Conclusions

In this paper, we proposed exact algorithms for solving a version of the lifetime maximization problem in WSNs with a mobile sink. Our approach differs from those in the literature as it tries to find an optimal cyclic trajectory for the sink in the sensor field when the sink’s travel times between its different locations are not negligible. We formulated several mixed-integer linear programs to model different variants of this problem. We then focused on one of these formulations and provided cutting-plane and decomposition algorithms to solve it. Our computational experiments indicated that the proposed algorithms can potentially improve the solvability of the underlying model.

An important future research direction would focus on developing efficient exact algorithms for MSM-FS2, which proves to be a computationally challenging problem. MSM-FS2 generally yields a higher network lifetime and seems to be more appropriate for applications in which real-time transmission of data is less crucial. Another problem may investigate the case in which the sink trajectory does not need to be a cycle and can take on more general forms, e.g., a figure-eight trajectory. Furthermore, we also suggest that our research can be extended to the case of a delay tolerant mobile sink model as formulated in [27, 28].

From a practical point of view, our models are primarily intended for finding an optimal sink trajectory. Given a specific trajectory for the sink, another future research problem would devise
a distributed algorithm for finding a set of optimal routing decisions for the sensor nodes in a
decentralized manner. A distributed routing algorithm might be more valuable than a centralized
one in certain applications, e.g., when the routing algorithm can be built into the network protocol.
While there exist several distributed algorithms for MSM (see, e.g., [12]), none of them can be
directly applied to the models proposed in this paper. Hence, the development of distributed
algorithms for solving the models presented in this paper poses another future research challenge.

References


lation and distributed protocols for wireless sensor networks lifetime maximization. In IEEE

the lifetime of wireless sensor networks with mobile sinks. Computers & Operations Research,
2012 (to appear).


networks. In Proceedings of the 37th Annual Allerton Conference on Communication, Control,
and Computing, volume 37, pages 1191–1200, Monticello, IL, September 1999.


