On Mobile Sensor Data Collection Using Data Mules

Arun Das, Anisha Mazumder, Arunabha Sen
School of Computing, Informatics and Decision System Engineering
Arizona State University
Tempe, Arizona 85287, USA
Email: {arun.das, amazumde, asen}@asu.edu

Nathalie Mitton
Inria
40 Avenue Halley
59650 Villeneuve D’ASCQ, France
Email: nathalie.mitton@inria.fr

Abstract—The sensor data collection problem using data mules has been studied fairly extensively in the literature. However, in most of these studies, while the mule is mobile, all sensors are stationary. The objective of most of these studies is to minimize the time needed by the mule to collect data from all the sensors and return to the data collection point from where it embarked on its data collection journey. The problem studied in this paper has two major differences with these earlier studies. First, in this study we assume that both the mule as well as the sensors are mobile. Second, we do not attempt to minimize the data collection time. Instead, we minimize the number of mules that will be needed to collect data from all the sensors, subject to the constraint that the data collection process has to be completed within some pre-specified time. We show that the mule minimization problem is NP-Complete and analyze the problem in two settings. We provide solutions to the problem in both settings by first transforming the problem to a generalized version of the minimum flow problem in a network, and then solving it optimally using Integer Linear Programming. Finally, we evaluate our algorithms through experiments and present our results.

I. INTRODUCTION

In the prevalent literature, “data mules” are referred to mobile devices that travel to, and collect data from sensors located at sparsely dispersed points in a deployment area. The mules then subsequently bring back the collected data to a central collection point [1], [2]. From an energy saving perspective, data mules offer an attractive alternative to the sensor data collection process carried out by multi-hop forwarding techniques. Data mules travel to the vicinity of sensors in the deployment area and once within the communication range of the sensors, start collecting data from these sensors. Since the amount of data stored in different sensors may vary, the data collection time required by the mule for each sensor may also vary. Although data collection using data mules may result in energy savings, it might also result in increased delay (or latency) for data collection. Accordingly, a number of studies have also been undertaken to find intelligent paths for the mules with the objective of minimizing the delay [3].

Although sensor data collection problems using data mules have been studied fairly extensively in the literature, in most of these studies, while the mule is mobile, the sensors are assumed to be stationary. The objective of a majority of these studies is to minimize the time needed by the mule to collect data from all the sensors and return to the central collection point. The problem studied in this paper has two major differences with earlier studies. First, in this study we assume that both mules and sensors are mobile. Second, we do not attempt to minimize the data collection time, instead, we minimize the number of mules required to collect data from all sensors, subject to the constraint that the entire data collection process has to be completed within some pre-specified time. We term this problem as the Mule Minimization Problem (MMP). It may be noted that stationary sensors can be viewed as a special case of mobile sensors, hence a solution technique for the MMP is equally applicable to both mobile and stationary sensors. We now outline some of the specifics of the MMP.

In the MMP, we assume that a central controller has the knowledge of: (i) the number of sensors in the deployment area, (ii) the trajectories of their movement, (iii) their location at every instance of time during the data collection period $T$, (iv) their speed, and (v) the amount of data available on each sensor. From this information, a centralized controller computes (i) the minimum number of mules required to read data from all sensors within the pre-specified time $T$, and (ii) the trajectories that the mules should follow in order to accomplish this task. We illustrate the problem with the help of an example. Figure 1 shows the trajectories of six mobile sensors on a two dimensional deployment area and their locations at various instances of time. We assume that the speed of the sensors is uniformly 1 unit/sec. (where the unit of measure can be feet, meter, etc.). The location of the sensors moving at this speed at various instances of time between the time interval [0–46] are also shown in Figure 1, however, to retain clarity, not all locations of each sensor are shown.

![Fig. 1: Locations and trajectories of six sensors $S_1, \ldots, S_6$ between the time interval [0–46]](image)
process, (the location at the first instance of time when data is transferred between a sensor and mule), and also excludes the travel time to the collection center from the end of the data collection process, (the location at the instance of time when data from all sensors have been read by one or more mules).

For the sake of simplicity, for the example of Figure 1 we assume that (i) a mule can collect data from a sensor only when the distance between the mule and the sensor is at most one unit, (ii) the speed of the mule is equal to that of the sensors, i.e. 1 unit/sec, and (iii) the rate of data transfer between the sensor and the mule is 1 unit/sec (where the unit of measure can be megabytes, kilobytes, etc.). The solution to the problem of Figure 1 is shown in Figure 2, where only one mule is sufficient to collect all data from six sensors within the pre-specified time of \( T = 46 \). In Figure 2, the trajectory of the mobile data mule is shown with a thick red line, and the locations where the mule collects data from the sensors are shown with hatched rectangles. Specifically, the mule collects 3 units of data from \( S_6 \) during time interval [10-13], 2 units of data from \( S_1 \) during time interval [13-15], 2 units of data from \( S_2 \) during time interval [17-19], 3 units of data from \( S_3 \) during time interval [23-26], 4 units of data from \( S_3 \) during time interval [33-37], and finally 6 units of data from \( S_3 \) during time interval [38-44] at a uniform data transfer rate of 1 unit/sec.

As shown in Figure 1, our model allows different sensors to have different amounts of data to transfer. This implies that each sensor may require different amounts of time to transfer such data to a mule. This raises an important question with respect to the available data collection infrastructure of the mobile data mules: Whether fragmented data collection from sensors is allowed or not? In other words, whether a single mule should collect all available data from a sensor, or can multiple mules collect fragments of the data from a sensor that can then be put together by a central system? If multiple mules are allowed to pick up fragments of data from a sensor, then there has to be synchronization between the mules to determine which mule picks up which part of the data. Additionally, the mules must also possess a level of intelligence to facilitate such synchronization by reading parts of the sensor data at specific intervals of time. In this paper we consider both versions of the problem, one in which fragmented data collection is allowed, i.e. mules have sufficient intelligence to allow synchronization and are capable of reading specific parts of the sensor data, and the other, when fragmented data collection is disallowed, i.e. mules do not possess such intelligence and is necessary that a single mule read a sensor’s data in its entirety. As discussed in Section III, the complexity of the solution for the second version is considerably higher than the first.

The rest of the paper is organized as follows: in Section II we outline the related work on this topic, in Section III we formally state the MMP, show that it is NP-Complete, and provide solution techniques for the two settings of the problem. Section IV discusses the experiments conducted using our techniques, and in Section V we conclude the paper.

II. RELATED WORK

As indicated earlier, to the best of our knowledge, the mule minimization problem with a constraint on the data collection time has not been studied. Most of the previous work either consider different problems and assumptions, or focus on similar issues but with different goals and objectives. In [4], [5] the authors focus on the problem of choosing the path of a data mule that traverses through a sensor field with sensors generating data at a given rate. To this purpose, the authors of [4] designed heuristic algorithms to find a path that minimizes the buffer overflow at each sensor node, that they later extended to multiple data mules and viewed it as a vehicle routing problem (VRP) [6]. In these works, however, it was assumed that data mules need to travel to the sensor nodes’ exact location to collect data (i.e., excluding remote communication). This assumption facilitates TSP-type formulations for their problem and makes the data mule path selection problem similar to a packet routing problem, such as the one studied in [7]. However, these formulations under-utilize communication capabilities, as data mules can use wireless communication to collect data from nodes without visiting their exact locations.

Zhao and Ammar [8] studied the problem of optimally controlling the motion of a data mule in mobile ad-hoc networks. A data mule, called a message ferry, mediates communications between sparsely deployed stationary nodes. They considered remote communication, but path selection was done based on a TSP-like formulation. They extended their work to multiple data mules in [9] and presented heuristic algorithms. In [3], the authors proposed to adapt the motion of the mule to minimize the full delay of data gathering. Ma and Yang [10] discussed the path selection problem under different assumptions. Their objective was to maximize the network lifetime, which is defined as the time until the first node dies (i.e. minimum of the lifetime of all nodes). They considered remote wireless communication and also multi-hop communication among nodes. When the path of a mule is given, they showed the problem of maximizing the network lifetime is formulated as a flow maximization problem that has a polynomial time algorithm. Choosing the mule’s path is done by their heuristic algorithm that uses a divide and conquer approach to find a near optimal path for each part of the path.

Other approaches like the one in [11] are also inspired from vehicle networks to transfer data. They are called carry-and-forward techniques and offer more opportunistic data delivery.
They can thus neither guarantee any QoS, nor limit the number of mules. Finally, as mentioned, all these works do not consider mobile sensors. To the best of our knowledge, the only study to consider mobile sensors is [12], but this work assumes a given number of mules and does not try to minimize this number, but instead proposes the best coverage possible at a given time.

III. Mule Minimization Problem

Our technique for solving the Mule Minimization Problem (MMP) is to first transform the problem into a network flow problem, and then utilize Integer Linear Programming (ILP) to solve the network flow problem. As indicated in Section I, in this paper we consider two variants of the problem: (i) when the mules have sufficient intelligence that allow fragmented data collection, i.e. the task of reading data from a single sensor can be distributed to different mules, and (ii) when the mules do not possess such intelligence, and the entirety of a single sensor’s data must be read by a single mule. In Section III-A we present our solution for the MMP where mules have such intelligence, and in Section III-B, we extend the technique from Section III-A to address the scenario where the mules do not possess such intelligence.

A. Fragmented Data Collection - Mules with Intelligence

Although the MMP is a continuous time domain problem (as the mobile sensors and mules can be anywhere in the deployment area at a given time), our approach to solving the MMP is to discretize both time and space. In our technique, time is discretized into equal intervals of length \( \delta \), and space into equal intervals of length \( \varepsilon \). The discretization of time and space allows the possibility of degradation of the quality of the solution, i.e. our approach may not be able to find the absolute minimum number of mules needed to collect data from all sensors. However, the advantages of such discretization is lower computation time, as the computational complexity of our solution is inversely proportional to the magnitude of the variables \( \delta \) and \( \varepsilon \). Thus our technique provides a direct mechanism to manage the trade-off between the quality of the solution (measured in terms of accuracy) and the cost of the solution (measured in terms of computation time).

We formally setup the problem as follows: We consider a set of \( n \) mobile sensors \( A = \{a_1, \ldots, a_n\} \) moving on a one dimensional plane (i.e., a line)\(^1\) over time instances \( 0, \ldots, T \). It may be noted that although the movement of sensors is restricted to one dimension, there is no restriction on the direction of their movement, i.e. they can move either left and/or right, and can change directions arbitrarily. Let \( p(a_t) = x(a_t) \) be the location of sensor \( a_t \) at time instance \( t \) where \( x(a_t) \) denotes the \( x \)-coordinate of \( a_t \) at time \( t \). We assume that data from a sensor \( a_t \) can be collected by a mule \( M_j \) only if the distance between \( a_t \) and \( M_j \) is less than the communication range of the sensor and mule, denoted by \( r \).

Theorem 1. Mule Minimization Problem is NP-complete.

Proof: The problem instance of MMP where the sensors are stationary (a special case of mobile sensors), is equivalent to the Geometric Disk Cover Problem which is known to be NP-complete [13].

To find the solution for the MMP, we first transform it to a generalized version of the minimum flow problem on a directed graph \( G = (V, E) \). In this formulation, individual flows correspond to a path from the source to the destination node in \( G = (V, E) \). The number of flows provides the number of mules required, and each path corresponds to the required trajectory of a mule as it moves through the deployment area collecting data from the sensors. Since we transform the MMP into a generalized version of the Minimum Flow Problem (MFP) [14], we first outline the MFP and then its generalized version, GMFP.

Minimum Flow Problem (MFP): Given a capacitated network \( G = (V, E) \) with a non-negative capacity \( c(i, j) \) and with a non-negative lower bound \( l(i, j) \) associated with each edge \((i, j)\) and two special nodes, a source node \( S \) and a sink node \( D \), a flow is defined to be a function \( f : E \rightarrow \mathbb{R}^+ \) satisfying the following conditions:

\[
\sum_{j \in V} f(i, j) - \sum_{j \in V} f(j, i) = \begin{cases} 
F, & i = S \\
0, & i \neq S, D \\
-F, & i = D 
\end{cases}
\]

\[
l(i, j) \leq f(i, j) \leq c(i, j)
\]

for some \( F \geq 0 \) where \( F \) is the value of the flow \( f \). The MFP finds a flow \( f \) for which \( F \) is minimized.

Generalized Minimum Flow Problem (GMFP): The generalized version of the MFP is similar to the MFP, except that the lower bound on the flow requirement \( l(i, j) \) is no longer associated with an edge \((i, j)\), but associated with a set of edges \( E_k \subseteq E \) of the graph \( G = (V, E) \), and is denoted by \( l_k \). Formally, the problem can be stated as follows:

Given a capacitated network \( G = (V, E) \) with a non-negative capacity \( c(i, j) \) associated with each edge \((i, j)\), a set of subsets \( E' \) of the edge set \( E \) (i.e. \( E' = \{E_1, \ldots, E_p\} \)), where \( E_k \subseteq E, \forall k, 1 \leq k \leq p \), a lower bound on the flow requirement \( l_k \) associated with each \( E_k \), \( 1 \leq k \leq p \), and two special nodes, a source node \( S \) and a sink node \( D \). A flow is defined to be a function \( f : E \rightarrow \mathbb{R}^+ \) satisfying the following conditions:

\[
\sum_{j \in V} f(i, j) - \sum_{j \in V} f(j, i) = \begin{cases} 
F, & i = S \\
0, & i \neq S, D \\
-F, & i = D 
\end{cases}
\]

\[
\forall E_k, 1 \leq k \leq p, \exists \ l_k, a \ lower \ bound \ of \ flow \ in \ E_k, implying \ that \ the \ total \ flow \ on \ the \ set \ of \ edges \ in \ E_k \ is \ such \ that \ \sum_{(i,j) \in E_k} f(i, j) \geq l_k, \ and \ f(i, j) \leq c(i, j), \forall (i, j) \in E_k 
\]

for some \( F \geq 0 \), where \( F \) is the value of the flow \( f \). The GMFP finds a flow \( f \) for which \( F \) is minimized.

It may be noted that when \( |E_k| = 1, \forall k, 1 \leq k \leq p \), and \( p = |E| \), the GMFP reduces to MFP.

MMP Graph Construction

We now outline the MMP graph construction process through an example where the movements of the sensors are restricted to a one dimensional space, i.e. the sensors are allowed to move either left or right on a straight line and are allowed to change directions arbitrarily. This restriction is imposed only to explain the graph construction process in a lucid way. Once the construction process is understood, the same principles can
be followed for constructing the MMP graph when the sensors move in a two or three dimensional space. It may be recalled that data from each sensor can be collected by one or more mules, within a pre-specified data collection time $T$, and the goal of the MMP is to collect data from all sensors with as few mules as possible within time $T$.

Figure 3 illustrates our example, our example has three sensors and a pre-specified data collection time $T = 2$, each sensor has one unit of data that must be read by a mule and the rate of data transfer available to all mules is one unit of data per time step. As seen in Figure 3, sensor $a_1$ is at location $x_1$ at time $t_0$ and in location $x_2$ at time $t_1$. Similarly, the sensor $a_2$ is at location $x_3$ at time $t_0$ and in location $x_4$ at time $t_1$, and sensor $a_3$ is at location $x_5$ at time $t_0$ and at location $x_6$ at time $t_1$. Although in this example, all sensors are moving left at the same speed, the sensors are free to move in either direction and at different speeds. A mule can collect data from a sensor only if the sensor is within the communication range $r$. If we assume that $r = \varepsilon$, as shown in Figure 3, then for this example, in order to collect one unit of data from sensor $a_1$, there must be a mule at $(x_0, t_0)$ or $(x_1, t_0)$ or $(x_2, t_0)$ or $(x_0, t_1)$ or $(x_1, t_1)$, where $(x_i, t_j)$ indicates location $x_i$ at time $t_j$. Using a similar reasoning we can conclude that in order to collect data from sensor $a_2$, there must be a mule at $(x_3, t_0)$ or $(x_5, t_0)$ or $(x_6, t_0)$ or $(x_5, t_1)$ or $(x_3, t_1)$ or $(x_4, t_1)$ or $(x_5, t_1)$ or $(x_3, t_2)$. Also, in order to collect data from sensor $a_3$, there must be a mule at $(x_5, t_0)$ or $(x_6, t_0)$ or $(x_7, t_0)$ or $(x_4, t_1)$ or $(x_5, t_1)$ or $(x_6, t_1)$.

The MMP graph $G = (V, E)$ is a directed graph and is constructed in the following way: It may be noted that for a mule $M_j$ to collect data from a sensor $a_i, 1 \leq i \leq n$ the distance between the mule and the sensor cannot exceed the communication range $r$. For this reason, in the above example, to collect data from sensor $a_1$, there must be a mule at $(x_0, t_0)$ or $(x_1, t_0)$ or $(x_2, t_0)$ or $(x_0, t_1)$ or $(x_1, t_1)$. Corresponding to each sensor $a_k, 1 \leq k \leq n$, there exists a set of potential $LT_k = (\text{location}, \text{time})$ pairs of the form $(X_{k,i}, T_{k,j})$, and a mule must be in at least one of these locations at the specific time to be able to collect one unit of data from the sensor $a_k$.

Also, the time taken to collect one unit of data from a sensor is inversely proportional to the rate of data transfer available to the mule. If the rate of data transfer is assumed to be uniform for all mules at $\mu$ units of data per $\delta$ unit of time, then if $d_k$ units of data have to be collected from sensor $a_k$, then at least $d_k' = \lceil \frac{d_k}{\mu \delta} \rceil$ elements of the set $LT_k$ must be chosen in order to satisfy the requirement that $d_k$ units of data are collected from sensor $a_k$. In the example of Figure 3, with $\delta = 1$, $\mu = 1$, and $d_1 = d_2 = d_3 = 1$ we will have $LT_1 = \{(X_{1,1}, T_{1,1}), (X_{1,2}, T_{1,2}), \ldots, (X_{1,5}, T_{1,5})\}$, $LT_2 = \{(X_{2,1}, T_{2,1}), (X_{2,2}, T_{2,2}), \ldots, (X_{2,6}, T_{2,6})\}$, $LT_3 = \{(X_{3,1}, T_{3,1}), (X_{3,2}, T_{3,2}), \ldots, (X_{3,6}, T_{3,6})\}$.

For each $LT_k = \{(X_{k,1}, T_{k,1}), \ldots, (X_{k,p_k}, T_{k,p_k})\}$, $1 \leq k \leq n$, in graph $G = (V, E)$ we introduce (i) $X_{k,i}$ type nodes, (ii) $T_{k,j}$ type nodes, and (iii) a directed edge from node $X_{k,i}$ to node $T_{k,j}, \forall i, 1 \leq i \leq p_k$. It may be noted that the $(X_{i,j}, T_{i,j})$ pair need not be unique and that the pairs $(X_{i,j}, T_{i,j})$ and $(X_{j,k}, T_{k,l})$ may represent the same $(\text{location}, \text{time})$ pair. In case of non-unique $(X_{i,j}, T_{i,j})$ pairs, only one pair of nodes are created in the graph $G = (V, E)$. In our example the pairs $(X_{2,2}, T_{2,2}) = (X_{3,1}, T_{3,1}) = (x_3, t_0)$ thus only one pair of nodes corresponding to location $x_3$ at time $t_0$ will be created in $G$. This construction is shown in Figure 4. The $X_{i,j}$ type nodes are referred to as location nodes, and $T_{i,j}$ type nodes are referred to as time nodes. In addition to these nodes, we also add one source node $S$ and one sink node $D$. In addition to the directed edges of type $X_{k,i} \rightarrow T_{k,j}$, we also include three additional types of edges in $G$:

1) Mobility edges: If a mule located at $x_0$ at time $t_b$, can move to a location $x_1$, at time $t_d$, then in the graph $G = (V, E)$, we add a directed edge from the node $t_b$ to $x_1$.

It may be noted that whether the mule can move from location $x_0$ at time $t_b$ to a location $x_c$ at time $t_d$, depends on (i) distance between locations $x_0$ and $x_c$, (ii) time interval between $t_b$ and $t_d$, and (iii) speed of the mule.

2) Source edges: There is a directed edge from the source node $S$ to all location nodes.

3) Sink edges: There is a directed edge from all time nodes to the sink node $D$.

The capacity $c(i, j)$ is set to 1 for all edges in $G = (V, E)$.

As discussed earlier, an instance of the GMFP has a set of subsets $E'$ of the edge set $E$ (i.e. $E' = \{E_1, \ldots, E_p\}$, where $E_k \subseteq E, \forall k, 1 \leq k \leq p$), with a lower bound on the flow requirement $l_k$ associated with each $E_k$. If $l_k$ is the lower bound of flow in $E_k$, the GMFP requires that there must exist at least $l_k$ edges $(i, j) \in E_k$ such that $\sum_{(i,j) \in E_k} f(i, j) \geq l_k$ and $f(i, j) \leq c(i, j)$. In the graph $G = (V, E)$, we set $E_k = LT_k, \forall k, 1 \leq k \leq p$. In our example, since $LT_1 = \{(x_0, t_0), (x_1, t_0), (x_2, t_0), (x_3, t_1), (x_4, t_1), (x_5, t_1), (x_6, t_1)\}$, we set $E_1 = \{(x_0 \rightarrow t_0), (x_1 \rightarrow t_0), (x_2 \rightarrow t_0), (x_3 \rightarrow t_1), (x_4 \rightarrow t_1)\}$. We set the lower bound of the flow requirement in $E_k, 1 \leq k \leq p$ to be $d_k'$, i.e., $l_k = d_k'$, where $d_k'$ is the number of time units required to collect $d_k$ units of data from sensor $a_k$ at $\mu$ units of data per $\delta$ unit of time. In this example if $\delta = 1$, $\mu = 1$ and $d_1 = d_2 = d_3 = 1$, then at least three edges in the edge set $\{(x_0 \rightarrow t_0), (x_1 \rightarrow t_0), (x_2 \rightarrow t_0), (x_3 \rightarrow t_1), (x_4 \rightarrow t_1)\}$ must have a flow of one unit. The GMFP graph $G = (V, E)$ constructed for the problem instance with three sensors in Figure 3 is shown in Figure 4. The directed edge set $E_1, E_2$ and $E_3$, corresponding to three sensors, $a_1, a_2$ and $a_3$ are shown enclosed in three rectangular boxes, colored red, yellow and blue respectively.

**Solution of MMP**

We solve the MMP problem by solving the GMFP using Integer Liner Programming. The ILP formulation is as follows:

\[
\text{minimize } F
\]

subject to,

\[
\sum_{j \in V} f(i, j) - \sum_{j \in V} f(j, i) = \begin{cases} 
F, & i = S \\
0, & i \neq S, D \\
-F, & i = D
\end{cases}
\]
We prove that the minimum number of mules required to collect data from all the sensors is equal to the solution of the GMFP on graph $G=(V,E)$, and the trajectories of the mules can be constructed from the solution of the GMFP.

**Theorem 2.** Any valid flow of the GMFP provides the minimum number of mules needed to collect data from all the mobile sensors within the specified data collection time $T$. The solution also provides the trajectory that the mules need to follow in order to collect data from the sensors.

**Proof:** If the minimum number of mules needed to collect data from all the sensors is $m$, there exists $m$ flows (paths) from the source to the destination node in $G$. Suppose that the location-time pair of a mule $M_i, 1 \leq i \leq m$ is given by, $(l_{i,1}, t_{i,1}), (l_{i,2}, t_{i,2}), \ldots, (l_{i,q_i}, t_{i,q_i})$. Being at these locations at these times, enables the mules to collect $\mu$ units of data from the sensors. A flow of one unit (or a path) for the source node $S$ to the destination node $D$ can be constructed in the following way: A path from $S$ to $D$ will be $S \rightarrow l_{i,1} \rightarrow \ldots \rightarrow l_{i,q_i} \rightarrow D$. Since such a path can be constructed from $S$ to $D$ for every mule $M_i$, there will be $m$ unit flows from $S$ to $D$.

If the solution to the GMFP is $m$ unit flows from $S$ to $D$, then $m$ mules are sufficient to collect data from all the sensors in the deployment area. Because of the way the constraints are set up, each unit flow corresponds to a path from $S$ to $D$ where the intermediate nodes are of the type $X_{k,i}$ and $T_{k,i}$ and edges are of the form $location \rightarrow time$. Suppose that there is a flow from $S$ to $D$ of the form $S \rightarrow l_a \rightarrow t_b \rightarrow l_c \rightarrow t_d \rightarrow l_e \rightarrow t_f \rightarrow D$. From this flow, we can construct a trajectory of a mule, where it moves from location $l_a$ at time $t_b$ to location $l_c$ at time $t_d$ to location $l_e$ at time $t_f$, collecting at least a part of the data to be collected from the sensors. Since $m$ flows are sufficient to satisfy the lower bound constraints imposed on the graph by each sensor (which is equal to the amount of data to be collected from each sensor), we can conclude that $m$ mules are sufficient to collect all data from all sensors.

### B. Unfragmented Data Collection - Mules without Intelligence

As discussed earlier, if multiple mules are allowed to pick up fragments of data from a sensor, then a central synchronization mechanism must exist and mules must be capable of reading parts of a sensor’s data. The previous section (Section III-A) addressed this scenario, and in this section we address the scenario where such synchronization capabilities are unavailable and a single mule is responsible to collect data from a given sensor in its entirety. First, we note that if the amount of data to be collected from a sensor is more than $\mu$ units then the solution proposed in Section III-A may not be able to guarantee that the entire data from a sensor will be collected by a single mule. We explain this with the help of the following example.

Consider a scenario where data has to be collected from two sensors $S_1$ and $S_2$. Sensor $S_1$ has $2\mu$ units of data to provide, and the sensor $S_2$ has $\mu$ units of data to provide. Suppose that due to the locations of the sensors, their speeds of movements, and the data collection threshold time $T$, there are only two $(location, time)$ pairs $(l_1, t_1), (l_2, t_2)$ where data collection from $S_1$ is feasible. Similarly, there are two $(location, time)$ pairs $(l_3, t_3)$ and $(l_4, t_4)$ where data collection from $S_2$ is feasible. Suppose also, that due to the speed of movement of the mules, it is possible for a mule to travel from location $l_1$ to $l_2$ within time interval $[t_1-t_2]$ and also to travel from location $l_3$ to $l_4$ within time interval $[t_3-t_4]$. In addition, suppose that $T$ is at least as large as the time interval between $[t_1-t_2]$ and $[t_3-t_4]$, but is less than the time interval $[t_1-t_2]$. To illustrate our example further, suppose that $t_1 = 1, t_2 = 2, t_3 = 1$ and $T = 2$. In this case there can be two optimal solutions:

1. **Solution 1:** Mule $M_1$ collects $2\mu$ units from $S_1$ at time $t_1$, and $\mu$ units from $S_2$ at time $t_2$.
2. **Solution 2:** Mule $M_1$ collects $\mu$ units from $S_1$ at time $t_1$, and $\mu$ units from $S_2$ at time $t_2$.

Clearly, in Solution 1, only one mule collects the entire data $(2\mu$ units) from $S_1$, but in Solution 2, one mule collects only half the data ($\mu$ unit) from $S_1$ and the other mule collects the rest. However, there is no way for the optimal solution for the GMFP on graph $G=(V,E)$ (Section III-A), to distinguish between these two solutions as the minimum flow in both these cases will be two. However, we show that the version of the MMP where a single mule is required to collect the entire data from a single sensor, can be solved by constructing a new graph $G'=(\mathcal{V}, \mathcal{E})$ and solving the GMFP on this new graph. We describe the construction process of graph $G'$ next.

As discussed earlier, in the graph shown in Figure 4, corresponding to each sensor $a_k, 1 \leq k \leq n$, there exists a set of edges $E_k = (location, time)$ pairs of the form $(X_{k,i}, T_{k,i})$, such that a mule is able read $\mu$ units of data from sensor $a_k$ if it is present at location $X_{k,i}$ at time $T_{k,i}$. In Figure 4, all such $(location, time)$ pairs are enclosed within a black rectangle, which we now refer to as a layer. To accommodate the requirement that a single mule collects all the data from a given sensor, we construct graph $G'=(\mathcal{V}, \mathcal{E})$ by replicating the sole layer of $G$, $n$ times in $G'$. We keep the structure of the edges and nodes in each layer as is, but distinguish the nodes and edges of different layers by associating them with...
the layers they belong to. For example, location and time nodes of the form $X_{k,i,m}$ and $T_{k,j,m}$ in $V$, is respectively represented as $X_{k,i,m}$ and $T_{k,j,m}$ for each layer $m, 1 \leq m \leq n$ in $V$. Similarly, (location, time) edges of the form $(X_{k,i,m}, T_{k,j,m})$ in $E$, is represented as $(X_{k,i,m}, T_{k,j,m})$ for each layer $m, 1 \leq m \leq n$ in $E$. The set of edges of the form $(X_{k,i,m}, T_{k,j,m})$ that appear in layer $m, 1 \leq m \leq n$ is referred to as Edges of Layer $m$ and is denoted by $\mathcal{E}_m$.

Corresponding to each sensor $a_k$, in each layer $m, 1 \leq k, m \leq n$, there exists a set of edges $\mathcal{E}_{k,m} = (\text{location, time})$ pairs of the form $(X_{k,i,m}, T_{k,j,m})$ signifying a location $X_{k,i,m}$ where a mule can be present at time $T_{k,j,m}$ to read $\mu$ units of data from sensor $a_k$. We define Edges Across Layers for Sensor $k$ as $\mathcal{E}ALS_k$ to be the set of all edges across all layers that signify this (location, time) pair where a mule can be present to collect $\mu$ units of data from sensor $a_k$. That is:

$$\mathcal{E}ALS_k = \bigcup_{m=1}^{n} \mathcal{E}_{k,m}, \quad \forall k = 1, \ldots, n$$

In addition to introducing the nodes and edges discussed above, for each layer $m, 1 \leq m \leq n$ we introduce additional nodes $u_m, v_m$, and an edge $u_m \rightarrow v_m$, as shown in Figure 5. A source node $S$ is also introduced in $G$ and is connected to all $u_m, 1 \leq m \leq n$ nodes. For each layer $m, 1 \leq m \leq n$ we connect node $v_m$ to all location nodes in layer $m$, that is $v_m \rightarrow X_{k,i,m}, \forall X_{k,i,m} \in \mathcal{E}_m$. Lastly, we introduce a sink node $D$ and connect all time nodes of the form $T_{k,j,m} \in E$ to $D$ as shown in Figure 5. Note that for clarity, all nodes and edges are not shown in Figure 5.

![Fig. 5: MMP graph constructed from the problem instance of Fig. 3](image)

The MMP for mules without intelligence can be solved by solving a generalized version of the MFP, although it may be noted that this generalization is different from the version of the MFP for mules with intelligence (Section III-A). It may be recalled that in the solution to the GMFP discussed in Section III-A, the lower bound on the flow requirement $l_k$ was associated with a set of edges $E_k \subseteq E$ of the graph $G = (V, E)$. In this version of GMFP titled New Generalized Minimum Flow Problem (NGMFP), the lower bound on the flow requirement $l_k$ is no longer associated with a set of edges, but instead with a set of set of edges $\mathcal{E}ALS_k \subseteq \mathcal{E}$.

The lower bound requirement of the NGMFP states that there should be at least $l_k$ units of flow through the edges of at least one set of edges $\mathcal{E}_{k,m}, 1 \leq m \leq n$ for all $k, 1 \leq k \leq n$. Because of the structure of the graph $G$, this lower bound requirement, together with the constraint that the upper bound of capacity of each edge set to one, the solution of the NGMFP on $G$ results in the solution of the MMP for mules without intelligence when we set $l_k = d_k', where $d_k'$ is the number of time units required to collect $d_k$ data units from sensor $a_k$, at $\mu$ units of data per $\delta$ unit of time. The NGMFP can be solved by using Integer Liner Programming (ILP), and is formulated using the following inputs:

Given (i) a set of sensors $\mathcal{A} = \{a_1, \ldots, a_n\}$, and a weight $d_k'$ representing the time units needed to collect $d_k$ data units from sensor $a_k$ at $\mu$ units of data per $\delta$ unit of time, (ii) a directed graph $G = (V, E)$, with subsets of edges associated with each layer $\mathcal{E}_m, 1 \leq m \leq n$, and a subset of edges associated with each sensor $\mathcal{E}ALS_k, 1 \leq k \leq n$, and (iii) capacity of all edges set to one. We first outline the variable used for the ILP:

For each sensor $a_k$, $\mathcal{E}_m$, and directed edge $(i,j)$:

$$y_{k,m} = \begin{cases} 1, & \text{if } \sum_{(i,j) \in (\mathcal{E}ALS_k \cap \mathcal{E}_m)} f(i,j) \geq d_k' \\ 0, & \text{otherwise} \end{cases}$$

The objective of the ILP is as follows:

$$\text{minimize } F$$

subject to,

$$\sum_{j \in V} f(i,j) - \sum_{j \in V} f(j,i) = \begin{cases} F, & i = S \\ 0, & i \neq S, D \\ -F, & i = D \end{cases}$$

$$\forall k, m, 1 \leq k, m \leq n, \text{ if the edge } (i,j) \in (\mathcal{E}ALS_k \cap \mathcal{E}_m) \sum_{m=1}^{n} f(i,j) \geq d_k' \times y_{k,m}$$

$$\forall y_{k,m} \geq 1, \forall k = 1, \ldots, n$$

$$\forall (i,j) \in E, f(i,j) \leq c(i,j)$$

$$\forall f(i,j) = 0/1$$

$$\forall y_{k,m} = 0/1$$

Using a similar reasoning from Theorem 2, it can be shown that a valid flow for the NGMFP provides the minimum number of mules (and their trajectories), required to collect data from all mobile sensors within the specified data collection time $T$, such that data collected from a single sensor is not fragmented across multiple mules. Due to a lack of space we leave out the formal proof.

**Extension to higher dimensions:** We solved the MMP problem by constructing a graph $G = (V, E)$ from an instance of the MMP problem and solving the GMFP (NGMFP) on it. We provided the explanation for the graph construction process through an example, where the movements of sensors were restricted to one dimension. However, our solution technique for the MMP is not restricted to only one dimensional movement.
of the sensors. A critical component of the graph is the directed edges of the form \( \text{location} \rightarrow \text{time} \) node pairs. If the locations of the sensors are restricted to one dimension, \( \text{location} \rightarrow \text{time} \) node pair takes the form \((x) \rightarrow t\), where \(x\) is the location and \(t\) is the time. If the locations of the sensors are expanded to two or three dimensions the \( \text{location} \rightarrow \text{time} \) node pairs will take the form \((x, y) \rightarrow t\) or \((x, y, z) \rightarrow t\), respectively to capture the two or three dimensional coordinates. However, such a representation will not in any way affect the generalized minimum flow based approach to the solution of the MMP.

![Graph](image)

Fig. 6: (a) Trajectories and available data units of 5 sensors in the time interval [0-10], (b) Number of mules vs. rate of data transfer, varying mule speeds in time interval [0-8], \(\varepsilon = 1.0, \delta = 1.0, r = 1\)

We present the experimental results of the solution techniques for the MMP in this section. For our experiments we considered 5 mobile sensors in a 2-dimensional deployment area over the time interval [0-10]. The sensor trajectories and their available data units considered for our experiments are shown in Figure 6(a). The sensor trajectories were specified by unique parametric equations and thus not all speeds considered were uniform and constant.

We used IBM CPLEX Optimizer 12.5 to solve ILPs to compute solutions for the MMP of Figure 6(a) under both intelligent and unintelligent mule settings. To discretize the deployment area, we set \(\delta = 1\) and \(\varepsilon = 1\). We then investigated the impact of different sensor and mule parameters, namely, the data transfer rate \(\mu\) per \(\delta\) unit of time, and the speed of the mule, on the total number of mules needed to gather data from all sensors in \(T = 8\) time. Figure 6(b) shows the required number of mules at a given speed, as the available rate of data transfer \(\mu\) is varied. For this specific problem instance and the given mule parameters, the number of mules required to read all sensor data within \(T = 8\) time does not vary under the intelligent and unintelligent mule settings. However, the computed mule trajectories were observed to be different in these two settings. Our experiments confirmed that for a given mule speed, increasing the data transfer rate lowers the number of mules required.

In our experiments, we also varied the variables \(\varepsilon\) and \(\delta\), that are used to discretize time and space. Our observations indicate that smaller values of \(\varepsilon\) and \(\delta\) allow our solution technique to be closer to the optimal solution in a continuous setting (when space and time are not discretized). Smaller values of \(\varepsilon\) and \(\delta\) increases the granularity of the solution space by increasing the number of \((\text{location, time})\) pairs considered by the ILP. Though this may result in solutions closer to the optimal, smaller values of \(\varepsilon\) and \(\delta\) increases the cost of computation considerably.

V. CONCLUSION

In this paper we studied the Mule Minimization Problem (MMP) where both the sensors and mules were mobile, and the objective was to minimize the number of mules required to collect data from all sensors within a pre-specified time \(T\). We showed that the MMP is NP-Complete and analyzed the problem in two settings. We provided solutions for the MMP by transforming the problem to a generalized version of the minimum flow problem in a network, and then solving it optimally using Integer Linear Programming. We evaluated our approach through experiments and presented our results.

REFERENCES


