On Mobile Sensor Data Collection Using Data Mules

Arun Das, Arunabha Sen, Anisha Mazumder, Nathalie Mitton

On Mobile Sensor Data Collection Using Data Mules

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

Nathalie Mitton
INRIA Lille - North Europe
University of Lille 1
40 Avenue Halley, 59650 Villeneuve D'ASCQ, France
Email: {nathalie.mitton}@inria.fr

***Abstract*—The sensor data collection problem using data mules have 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 the 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 provide a solution by first transforming it 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 extensive simulation and present the results.**

I. Introduction (1 page)

Mobile devices that can travel to the locations of sparsely dispersed sensors in a deployment area, collect data from the sensors and bring the data back to the central collection center has been referred in the literature as "data mules". From energy saving perspective, the data mules offer an attractive alternative to the sensor data collection process through a multi-hop forwarding technique. The data mules travel to the vicinity of the sensors in the deployment area and when they are within the communication range of the sensors, start collecting data from the sensors. Since the amount of data stored in different sensors may be different, the data collection times for the mule from different sensors may be different. A robot was used as a data mule for underwater environmental monitoring in [] [Vasilescu et al. 2005] and a UAV (unmanned aerial vehicle) was used as a data mule in for structural health monitoring [] [Mascarenas et al. 2008]. Although data collection using data mules may result in energy savings, but it might also result in increased delay (or latency) of data collection. Accordingly a number of studies have been undertaken to find cleaver paths for the mules that would minimize the delay [].

Although the sensor data collection problem using data mules have been studied fairly extensively in the literature, 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 the earlier studies. First, in this study we assume that both the mule as well as the sensors are *mobile*. It may be noted that as mobile sensors can be viewed as a special case of stationary sensors, our solution technique is equally applicable to both mobile and stationary sensors. 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* (MMP) is NP-complete and provide a solution by first transforming it 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 extensive simulation and present the results.

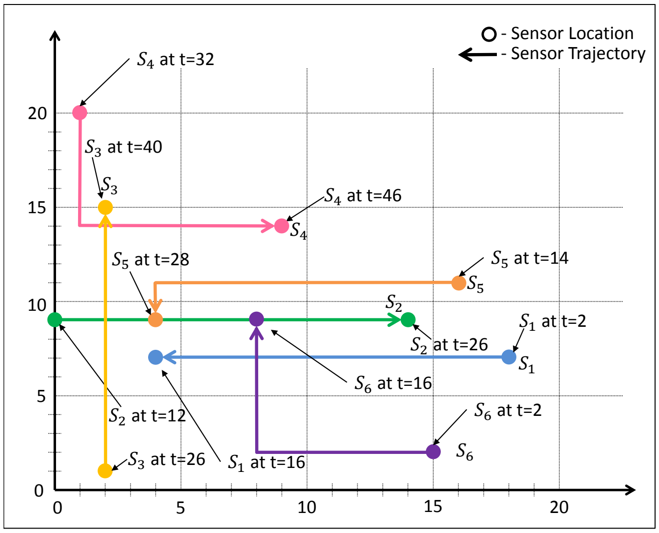

Fig. 1: The locations of six tags $T_1, \ldots, T_6$ at start time $t = 0$, their trajectories, and their locations at time $t = 14$.

For the Mule Minimizing Problem, 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 $\mathcal{T}$, and (iv) their speed. From this set of information, the centralized controller computes the minimum number of mules that will be needed to read data from all the sensors and the trajectories that the mules should follow in order to accomplish this task within the pre-specified time $\mathcal{T}$. We illustrate the problem with the help of

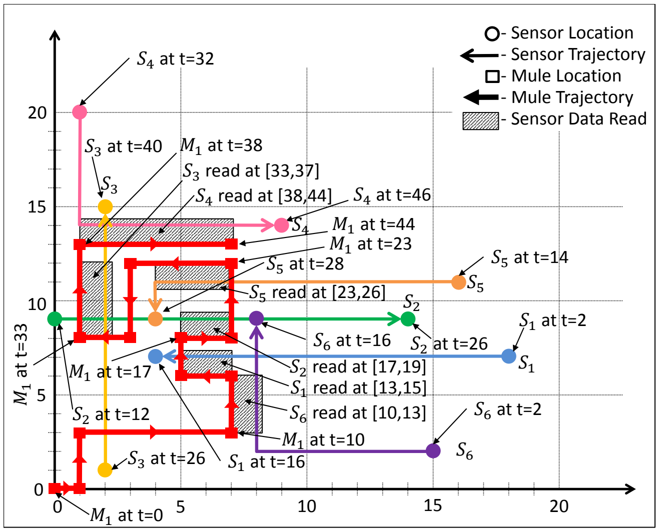

Fig. 2: One single reader that can read all six targets of Fig 1, its trajectories, and the *rendezvous points*, where the reader reads the tags. The trajectory of the reader is shown in thick red line and the rendezvous points are shown in small black rectangles.

an example as shown in figures 1, 2 and table 1. The figure 1 shows the trajectories of six mobile tags on a two dimensional deployment area and their locations at various instances of time. We assume that the speed of both the sensors and the readers are 1 unit/sec (unit may a feet, yard, meter etc.) . The location of the sensors moving at this speed at various instances of time between $t = 0$ to $t = 46$ is shown pictorially in figure 1 and also presented in the rows 2 to 7 in table 1.

The goal of the path planning problem is to find the minimum number of mules needed to collect data from all the sensors. For the sake of simplicity, in this example we have assumed that a mule can collect data from a sensor only when the distance between the mule and the sensor is at most one unit. The solution to problem of figure 1 is shown in figure 2, where only one mule is sufficient to collect all data from six sensors within the specified time of $\mathcal{T} = 50$. The trajectory of the mobile reader is shown in bold red line in 2. In the example of figure 1, the locations where the mule collects data from the sensors (rendezvous points) are shown in a shaded box in figure 2 and in bold letters in Table 1. For example, the mule collects 3 units of data from the sensor S_6 during time $t = 10$ to $t = 13$, 2 units of data from S_1 during $t = 13$ to $t = 15$, 2 units of data from S_2 during $t = 17$ to $t = 19$, 3 units of data from S_5 during $t = 23$ to $t = 26$, 4 units of data from S_3 during $t = 33$ to $t = 37$, and 6 units of data from S_4 during $t = 38$ to $t = 44$.

It may be noted that in our model different sensors may have different amount of data to transfer to the mule(s) and as such it might take different amount of times to transfer data from a sensor to a mule. In the example of figure 1, the sensors S_1, and S_2 have 2 units, the sensors S_6, and S_5 have 3 units, the sensors S_5 have 4 units of data to transfer to the mule(s). In section III, we discuss the path planning problem in detail and provide our solution technique. When a sensor has more than one unit of data to transfer to the mule, it gives rise a question which is not pertinent when each sensor has only one unit of data to transfer. In the example, S_4 has six units of data to transfer: the question is weather only one mule should collect all the data from a sensor or multiple mules should collect parts of the data from a sensor, to be put together at the collection center by the central node. If multiple mules pick up parts of the data from a sensor, then there has to be some synchronization between them to determine which mule picks up which part of the data. In order to achieve this, the mules must have a level of intelligence to carry out such synchronization. In this paper we consider two versions of the problem, one in which the mules have such intelligence and the other where they do not have such intelligence, and as such a single mule has to pick up the entire data from a sensor. As will be seen in section III, the complexity of the solution of the second version is considerably more than the first.

The rest of the paper is organized as follows.

II. Related Work (1/2 page)

As indicated earlier, to the best of our knowledge, the mule minimization problem with a constraint on the data collection time, have not been studied in the past. Somasundara et al. [2004] studied the problem of choosing the path of a data mule that traverses through a sensor field with sensors generating data at a given rate. They designed heuristic algorithms to find a path that minimizes the buffer overflow at each sensor node. In their subsequent work [Somasundara et al. 2007], they presented a heuristic algorithm for multiple data mules case based on the formulation as a vehicle routing problem (VRP). Gu et al. [2006] presented an improved algorithm for the same problem settings as [Somasundara et al. 2004]. In these works, it is assumed that data mules need to go to the sensor nodes exact location to collect data (i.e., no remote communication). This assumption facilitates TSP-like formulations of the problem and makes the path selection problem of a data mule similar to packet routing problem such as the one studied in [Meliou et al. 2006]. However, these formulations result in underutilized communication capability, since data mules can actually collect data from nodes without visiting their exact locations via wireless communications. Zhao and Ammar [2003] studied the problem of optimally controlling the motion of a data mule in mobile ad-hoc networks. A data mule, which is called a message ferry in their work, mediates communications between sparsely deployed stationary nodes. They considered the remote communication, but path selection is done based on a TSP-like formulation. They extended their work to multiple data mules case in [Zhao et al. 2005] and presented heuristic algorithms.

Ma and Yang [2006] discussed the path selection problem under different assumptions. Their objective is 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 the remote wireless communication and also multihop communication among nodes. When the path of data 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 path of data mule is done by their heuristic algorithm that uses the divide and conquer approach and finds a near optimal path for each part of the nodes.

III. Path Planning Problem (3 page)

We first transform the Mule Minimization Problem (MMP) into a network flow problem and then utilize integer linear programming to solve the network flow problem. As indicated in section I, in this paper we have considered two versions of the problem, one in which the mules have such intelligence and the other where they do not have such intelligence. In subsection III-A we present the our solution for the MMP where the mules have such intelligence. We extend the solution presented in subsection III-A to cover the scenario where the mules do not have such intelligence in subsection III-B.

A. 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 the solution discretizes both time and space. We discretize time into equal intervals of length δ and space into equal intervals of length ε. Discretization of time and space leaves open the possibility of degradation of the quality of the solution, in the sense, that it might not find the absolute minimum number of mules needed to collect data from all the sensors. However, such discretization leads to lower computation time, as the computational complexity of our solution is inversely related the magnitude of δ and ε. Thus our solution offers a clear 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 consider a set of n mobile sensors $\mathcal{A} = \{a_0, \ldots, a_{n-1}\}$ moving on a one dimensional plane (i.e., a line)[1] over time instances $0, \ldots, \mathcal{T}$. It may be noted that although the movement of tags is restricted to one dimension, there is no restriction on the direction of their movement, in that they can move towards the left and/or right, and in fact can move towards the left for a while before changing direction and moving towards the right. Let $p(a_i, t) = x(a_i, t))$ be the location of sensor a_i at time instance t where $x(a_i, t)$ denote the x-coordinate of a_i at time t. We assume that data from a sensor a_i can be *collected* by a mule $M_j$only if the distance between them is less than communication range r of the mule and the mobile sensor.

Theorem 1. *Mule Minimization Problem is NP-complete.*

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

We find the solution of the MMP by transforming it to a *a generalized version of the minimum flow problem* on a directed graph $G = (V, E)$. In our formulation, each flow corresponds to a path from the source to the destination node in $G = (V, E)$. The number of flows provides the number mules and the each path corresponds to the 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) [**?**], first we state the MFP and then it's generalized version, GMFP.

Minimum Flow Problem: (MFP) Given a capacitated network $G = \langle V, E\rangle$ 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 minimum flow problem is to determine 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 $G = \langle V, E\rangle$ and is denoted by l_k. Formally, the problem can be described 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 $E_k, \forall k, 1 \leq k \leq p$, 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 there must exist at least one edge$(i, j) \in E_k$ such that $l_k \leq f(i, j) \leq c(i, j)$ for some $F \geq 0$, where F is the value of the flow f. The generalized minimum flow problem is to determine a flow f for which F is minimized.

[1] We present the formulation in one dimension for clarity of explanation and brevity. Once the underlying principle for solution of the problem is understood, extension to higher dimensions is straightforward as same principle apply.

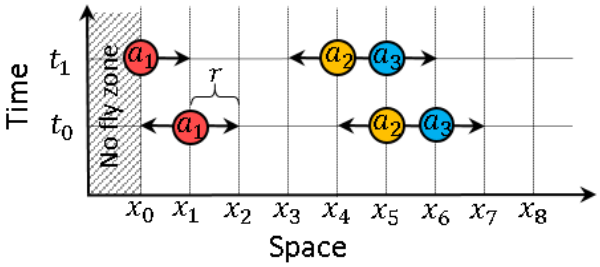

Fig. 3: Locations of three tags on one dimensional space (line) at two different instances of time

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

1) MMP Graph Construction: In this subsection, we first describe the MMP Graph construction process through an example, where the movements of the sensors are restricted to move on a one dimensional space (i.e., the sensors can move only *left* or *right* on a *straight line*, but they are allowed to change direction). We impose this restriction only to explain the graph construction process in a simple way. Once the construction process is understood, the same principle can be followed for constructing the MMP graph where the sensors are moving in a two or a three dimensional space. It may recalled that data from each sensor has to be collected by one or more mules, within the pre-specified data collection time $\mathcal{T}$. The goal of the MMP is to collect data from all the sensors with as few mules as possible within time $\mathcal{T}$. As shown in figure 3, our example has three sensors and pre-specified data collection time $\mathcal{T} = \in$. The sensor a_1 is at location x_1 at time t_0 and in location x_0 at time t_1. Similarly, the sensor a_2 is at location x_5 at time t_0 and in location x_4 at time t_1 and the sensor a_3 is at location x_6 at time t_0 and in location x_5 at time t_1. Although in this example, all the sensors are moving on the left direction at the same speed, the sensors are neither required to move in the same direction nor at the same speed. A mule can collect data from a sensor only if the sensor is within the sensing radius r of the mule. If we assume that $r = \varepsilon$, as shown in figure 3, then in this example, in order to collect data from the 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) indicate location x_i at time t_j. Using similar reasoning we can conclude that in order to collect data from the sensor a_2, there must be a mule at (x_4, t_0) or (x_5, t_0) or (x_6, t_0) or (x_3, t_1) or (x_4, t_1) or (x_5, t_1). Also, in order to collect data from the 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 sensing range of the mule. For this reason, in the above example, to collect data from the 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 $LT_k = (location, time)$ pair of the form $(X_{k,i}, T_{k,j})$, and a mule must be in at least one of these locations at a specific time to be able to collect one unit of data from the sensor a_k. In other words, if D_k units of data have to be collected from the sensor a_k, at least D_k elements of the set LT_k (say, $(X_{k,i}, T_{k,j})$) must be *chosen* in order to *satisfy* the requirement that D_k units of data have to be collected from the sensor a_k. In the example of figure 3, we will have $LT_1 = \{(X_{1,1}, T_{1,1}), (X_{1,2}, T_{1,2}), \ldots, (X_{1,5}, T_{k,5})\}$, $LT_2 = \{(X_{2,1}, T_{2,1}), (X_{2,2}, T_{2,2}), \ldots, (X_{2,6}, T_{k,6})\}$, $LT_3 = \{(X_{3,1}, T_{3,1}), (X_{3,2}, T_{3,2}), \ldots, (X_{3,6}, T_{k,6})\}$.

Suppose that

$$LT_k = \{(X_{k,1}, T_{k,1}), (X_{k,2}, T_{k,2}), \ldots, (X_{k,p_k}, T_{k,p_k})\}$$

. Corresponding to each $LT_k, 1 \leq k \leq n$, in the graph $G = (V, E)$, we will have, (i) $X_{k,i}$ type nodes, (ii) $T_{k,i}$, type nodes, and (iii) a directed edge from the node $X_{k,i}$ to the node $T_{k,i}, \forall i, 1 \leq i \leq p_k$.

It may be noted that $X_{i,j}$ (or $T_{i,j}$) need not be *unique* in the sense that $X_{i,j}$ and $X_{k,l}$ may represent the same location, just as $T_{i,j}$ and $T_{k,l}$ may represent the same time. For the example shown in figure 3, $X(2,2) = X(3,1) = x_5$ and $T(2,2) = T(3,1) = T(3,2) = T(3,3) = t_0$. In case of non-unique ($X_{i,j}$, $T_{i,j}$) pairs, only one pair of nodes is created in the graph $G = (V, E)$. In our example, LT_1 is $\{(x_0, t_0), (x_1, t_0), (x_2, t_0), (x_0, t_1), (x_1, t_1)\}$, LT_2 is $\{(x_4, t_0), (x_5, t_0), (x_6, t_0), (x_3, t_1), (x_4, t_1), (x_5, t_1)\}$ and LT_3 is $\{(x_5, t_0), (x_6, t_0), (x_7, t_0), (x_4, t_1), (x_5, t_1), (x_6, t_1)\}$. Since in this example $X_{2,2} = X_{3,1} = x_5$ and $T_{2,2} = T_{3,1} = t_0$, in the graph we create only one node pair (x_5, t_0). Although x_5 or t_0, may appear as a part of another node pair, such as (x_5, t_1), because $X_{2,6} = X_{3,5} = x_5$ and $T_{2,6} = T_{3,5} = t_1$, the pair (x_5, t_0) (or (x_5, t_1)) will appear only once. This shown in figure 4. In addition to these nodes, we add one source node S and one sink (destination) node D.

In addition to the directed edges from node type $X_{k,i}$ to the node type $T_{k,i}$, we will have three additional types of edges:

1) Mobility edges: If a mule located at x_a at time t_b, can move to a location x_c at time t_d, then in the graph $G = (V, E)$, we add a directed edge from the node t_b to x_c. It may be noted that whether the mule can move from location x_a at time t_b to a location x_c at time t_d, depends on (i) the distance between the locations x_a and x_b, (ii) the time interval between t_c and t_d, and (iii) the speed of the mule.
2) Source edges: There is a directed edge from the source node S to all unique nodes of the form $X_{k,i}$.
3) Sink edges: There is a directed edge from all unique nodes of the form $T_{k,i}$ to the sink node D.

The capacity $c(i, j)$ of the source and the sink edges is set equal to 1 and the capacity of mobility edges and the edges of the form $X_{k,i} \rightarrow T_{k,i}$ are set equal to 1.

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 $1 \leq f(i, j) \leq c(i, j)$. 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

is $\{(x_0,t_0),(x_1,t_0),(x_2,t_0),(x_0,t_1),(x_1,t_1)\}$, we set $E_1 = \{(x_0 \rightarrow t_0),(x_1 \rightarrow t_0),(x_2 \rightarrow t_0),(x_0 \rightarrow t_1),(x_1 \rightarrow t_1)\}$. We set the lower bound of 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 units of data that has to be collected by the mule from the sensor a_k. In this example if $d_1 = 3$, at least three edges in the edge set $\{(x_0 \rightarrow t_0),(x_1 \rightarrow t_0),(x_2 \rightarrow t_0),(x_0 \rightarrow t_1),(x_1 \rightarrow t_1)\}$ must have a flow of *one unit*.

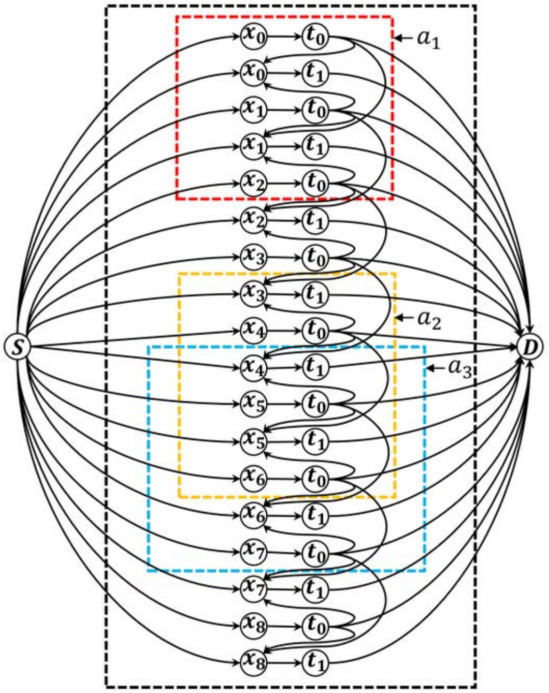

Fig. 4: MMP graph for mules with intelligence constructed from the instance of the problem shown in figure 3

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 brown, yellow and blue respectively in figure 4. It may be noted that not all source, destination and mobility edges are shown in figure 4 for the sake of clarity in the diagram.

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

$Objective : minimize\ F$
$subject\ to$

1. $\sum_{j \in V} f(i,j) - \sum_{j \in V} f(j,i) = \begin{cases} F, & i = S \\ 0, & i \neq S, D \\ -F, & i = D \end{cases}$
2. $\forall E_k, 1 \leq k \leq p$, if the edge $(i,j) \in E_k$, $\sum f_{i,j} \geq d_k$
3. $\forall (i,j) \in E, f_{i,j} \leq c(i,j)$
4. $\forall f(i,j) = 0/1$

We prove that the minimum number of readers required to collect data from all the sensors is equal to the solution of the of generalized minimum flow problem in graph $G = (V,E)$, and the trajectories of readers can be constructed from the solution of generalized minimum flow problem.

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 $\mathcal{T}$. It 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 mule $R_i, 1 \leq i \leq m$ 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, enabled the mules to collect 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 $S \rightarrow l_{i,1} \rightarrow, \ldots, l_{i,q_i \rightarrow D}$. Since such a path from the constructed from S to D for every mule R_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 readers 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 $location \rightarrow time$ or $location \rightarrow time$. 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 satisfy lower bound constraints imposed on the graph by each sensor (which is the amount of data to be collected from each sensor), we can conclude that m readers are sufficient to collect all the data from all the sensors.

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 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, the our solution technique for the RMP is not restricted to only one dimensional movement of the sensors. A critical component of the graph is the directed edges of the form *location* → *time* node pairs. If the locations of the sensors are restricted to one dimension *location* → *time* node pair takes the form *(x)* → *t*, where x is the location and t is the time. If the locations of the sensors are restricted to two or three dimensions *location* → *time* node pair will take the form *(x, y)* → *t* or *(x, y, z)* → *t*, i. e., the location will be specified by two or three dimensional coordinates. However, such a representation will not any way affect the generalized minimum flow based approach to the solution of the RMP.

B. Mules without intelligence

As discussed earlier, if multiple mules pick up parts of the data from a sensor, then there has to be some synchronization between them to determine which mule picks up which part of the data. In order to achieve this, the mules must have a level of intelligence to carry out such synchronization. The previous sub-section addressed this scenario. In this section we address the scenario where the mules do not have such synchronization capability and as such only one mule has to collect all the data from a sensor. First we note that if the amount of data to be collected from a sensor is more than one unit then the solution proposed in the sub-section III-A may not be able to guarantee that the entire data from will be collected by one mule only. We explain this with the help of the follwing example.

Consider a scenario where data has to be collected from two sensors S_1 and S_2. The sensor S_1 has two units of data to provide and the sensor S_2 has only one unit of data to provide. Suppose that due to locations of the sensors, their speeds of movements and data collection threshold time $\mathcal{T}$, 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_2, t_2) and (l_3, t_3) 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 and t_2 and also to travel from location l_2 to l_3 within time interval t_2 and t_3. In addition, suppose that the $\mathcal{T}$ is at least as great as the time interval between t_1 and t_2 and the time interval between t_2 and t_3, but is leas than the time interval t_1 and t_3. To make our example concrete, suppose that $t_1 = 1$, $t_2 = 2$, $t_3 = 1$ and $\mathcal{T} = 2$. In this case there can be two optimal solutions:
Solution 1: Mule 1 collects S_1 data from l_1 at time t_1 and l_2 at time t_2 and Mule 2 collects S_2 data from l_3 at time t_1.
Solution 2: Mule 1 collects S_1 data from l_1 at time t_1. Mule 2 collects S_2 data from l_3 at time t_1 and S_1 data from l_2 at time t_2.

Clearly, in Solution 1, only one mule collects entire data (two units) from S_1, but in Solution 2, one mule collects only half the data (one unit) from S_1 and the other mule collects the rest. However, there is no way for the optimal solution for the GMFP on the graph $G = (V, E)$ (whose construction rules are given in section III-A), to distinguish between these two solutions since the minimum flow in both of these two cases will be two. However, we show that the version of the MMP where a mule is required to collect the entire data from a sensor, can be solved by constructing a new graph $G' = (V', E')$ and solving the GMFP on this new graph. We describe the $G' = (V', E')$ construction process 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 $E_k = (location,\ time)$ pair of the form $(X_{k,i}, T_{k,j})$, and a mule must be in at least one of these locations at a specific time to be able to collect one unit of data from the sensor a_k. In figure 4, three such sets of edges E_1, E_2, E_3 corresponding to three different sensors a_1, a_2, a_3 are shown in enclosed in three rectangular boxes colored red, blue and yellow. In figure 4, all $(location,\ time)$ pairs are shown enclosed in a black rectangle, which will be referred to as *layer* from now on. In order to ensure that, only one mule collects all the data from a sensor, solving the GMFP on the graph $G = (V, E)$, comprising of only one layer of $(location,\ time)$ edges is not enough. In this case, we need to construct a new graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ comprising of n layers (as shown in figure 5), where n is the number of sensors. The edges and the structure of nodes and edges in each layer is identical. The subset of $\mathcal{E}$ which appears in layer $i, 1 \leq i \leq n$, will be referred to as *Edges in Layer i* and is denoted by $\mathcal{EL}_i, i, 1 \leq i \leq n$, where $\mathcal{EL}_i = \{E_{i,1}, \ldots, E_{i,n}\}, \forall i, 1 \leq i \leq n$. It may be recalled that in sub-section III-A, we had a similar scenario where the edge set $E = \{E_1, \ldots, E_n\}$, corresponding to n sensors. The edge set $\mathcal{EL}_{i,j} i, 1 \leq i, j \leq n$, of the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ corresponds the edge set $E_j, 1 \leq i \leq n$, of the graph $G = (V, E)$. In addition to $\mathcal{EL}_i$, we define another subset of $\mathcal{E}$ called *Edges Across Layers for Sensor i* and denote it by $\mathcal{EALS}_i, 1 \leq i \leq n$, where $\mathcal{EALS}_i = \{E_{1,i}, \ldots, EL_{n,i}\}$. In addition to creating the layers of nodes and edges, in $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we also add n additional edges $u_i \to v_i, 1 \leq i \leq n$, as shown in figure 5. The source node S is connected to all nodes $u_i, 1 \leq i \leq n$ and all v_i nodes play the same role in $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ as the node S played in the graph $G = (V, E)$. All *time* nodes of the (*location, time*) pair (edges) is connected to the destination node D. The graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ constructed from the problem instance of figure 3 is shown in figure 5, although not all nodes are shown in the figure for the sake of clarity.

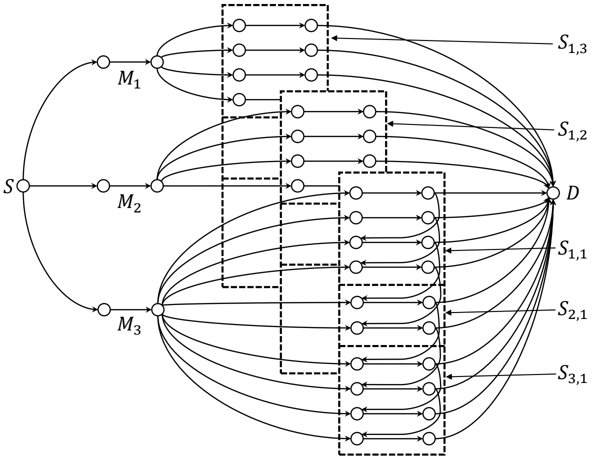

Fig. 5: MMP graph constructed from the instance of the problem shown in figure 3

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 generalized version of the MFP discussed in subsection III-A for solving MMP for mules with intelligence. It may be recalled that in the *Generalized Minimum Flow Problem* (GMFP) discussed in subsection III-A, the lower bound on the flow requirement $l(i, j)$ was associated with a set of edges $E_k \subseteq E$ of the graph $G = (V, E)$ and is denoted by l_k. In this version of *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 a set of set of edges $\mathcal{EALS}_k \subseteq E$ of the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{EALS}_k = \{E_{1,k}, \ldots, EL_{n,k}\}$. The lower bound requirement NGMFP states that *there should be at least l_k units of flow through the edges of at least one set of edges $E_{i,k}, 1 \leq i \leq n$.* Because of the structure of the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, 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 $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ results in the solution of the MMP for mules without intelligence, if we set $l_k = d_k$, where d_k is the amount of data to be collected from sensor a_k by a single mule. The NGMFP can be solved by using Integer Liner Programming. The ILP is formulated with the following input. Given a set of sensors $\mathcal{A} = \{a_1, \ldots, a_n\}$, and a weight d_k, representing the data to be collected by a mule from the sensor a_k. Directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with subsets of edges associated with each layer $\mathcal{EL}_i, 1 \leq i \leq n$ and subset of edges associated with each sensor $\mathcal{EALS}_i, 1 \leq i \leq n$ and capacity of all edges is one. We first outline the variables for the ILP:

For each sensor a_p, $\mathcal{EL}_q$ and directed edge (i, j):

$$y_{p,q} = \begin{cases} 1, & \text{if } \sum_{(i,j) \in (EALS_p \cap EL_q)} f(u, v) \geq d_p \\ 0, & \text{otherwise} \end{cases}$$

$$Objective : minimize\ F$$
$$subject\ to$$

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

2. $\forall p, q, 1 \leq p, q \leq n,$ `if the edge` $(i, j) \in (EALS_p \cap EL_q),$ `then` $\sum f_{i,j} \geq d_p \times y_{p,q}$

3. $\sum_{q=1}^{n} y_{p,q} \geq 1, \forall p = 1, \ldots, n$

4. $\forall (i, j) \in \mathcal{E}, f_{i,j} \leq c(i, j)$

5.$f(i, j) = 0/1, \forall (i, j) \in E$

6. $y_{p,q} = 0/1, \forall p, q, 1 \leq p, q \leq n$

IV. Experimental Results (1 page)

A. Path Planning Algorithm

In this section we present the results of the Path Planning algorithm in a mobile setting. We considered 5 mobile tags in a 2-dimensional deployment area over the time interval [0-10]. The tag trajectories considered for our experiments are shown in Figure **??** and were specified by parametric equations. The speed of the tags considered were not constant and uniform as the parametric equation of each tag trajectory was different.

We used IBM CPLEX Optimizer 12.5 to solve the ILP to compute the minimum number of readers required to read all 5 tags. We investigated the impact of different reader parameters, namely, the sensing-range radius r and the speed of readers d, on the total number of readers required to read all tags. The granularity of the discretized deployment area was specified by setting $\varepsilon = 0.5, \delta = 1.0$, and the total observation time was [0-10]. We varied the reader speed d from 0.2 to 3.0, and the sensing-range r from 0.5 to 1.25. Figure **??** shows a part of our results that, for a given reader speed d, highlight the change in the number of readers required when the sensing range r is varied. Our experiments showed that for a given reader speed, increasing the sensing range lowers the number of readers required to read all tags.

The variables ε and δ, used to discretize time and space were also varied in our experiments. Our observations indicate that smaller values of ε and δ allow our solution technique to be closer to the optimal solution in a continuous setting when space and time are not discretized. A smaller value of ε increases the granularity of the deployment space implying that a larger number of locations are considered by our solution technique. This may in turn result in fewer number of required readers to read all tags. On the other hand, smaller values of δ increases the number of possible locations the readers can be at a particular time interval, which could also lead to a fewer number of required readers. It may be noted that although smaller values of ε and δ can increase the accuracy of the solution, the cost of computation also increases considerably. For the example depicted in Figure **??**, we examined different values of $\delta = 0.5, 1, 2, 4, 8$ and $\varepsilon = 0.5, 1$ with $r = 1, d = 1$ in the time interval [0-8]. For these set of tag trajectories, the change of values for δ and ε has no impact on the minimum number of readers required to read all tags, in this case 3. However, the computation time significantly increased when smaller values of ε and δ were used.

V. conclusion

The conclusion goes here.