Delft University of Technology

Delft Center for Systems and Control

Technical report 14-042

Monitoring of traffic networks using mobile sensors*

Z. Cong, B. De Schutter, M. Burger, and R. Babuška

If you want to cite this report, please use the following reference instead:
Z. Cong, B. De Schutter, M. Burger, and R. Babuška, "Monitoring of traffic networks using mobile sensors," *Proceedings of the 17th International IEEE Conference on Intelligent Transportation Systems (ITSC 2014)*, Qingdao, China, pp. 792–797, Oct. 2014.

Delft Center for Systems and Control
Delft University of Technology
Mekelweg 2, 2628 CD Delft
The Netherlands
phone: +31-15-278.24.73 (secretary)
URL: https://www.dcsc.tudelft.nl

Monitoring of traffic networks using mobile sensors

Zhe Cong, Bart De Schutter, Mernout Burger, and Robert Babuška

***Abstract*— In this paper, we consider using mobile sensors (unmanned aerial vehicles) to monitor the traffic situation in a traffic network. We aim at finding optimal paths for mobile sensors such that the target links in the traffic network are covered; in addition, we also aim at minimizing energy consumption of mobile sensors. This problem is recast as a multiple rural postman problem. In order to solve this problem, we subsequently translate it into a multiple traveling salesman problem, by mapping the real traffic network into a virtual network, and then solve it by using mixed-integer linear programming. A simulation-based case study is used to illustrate our approach.**

I. INTRODUCTION

A. Background

Traffic information is important for traffic management and control. In order to obtain real-time traffic information, surveillance systems are often used in traffic networks. Traditionally, fixed sensors are put on roads, but they cannot move once installed. In order to improve the surveillance performance, some advanced technologies have been introduced. For example, floating car data (FCD) is a method to determine the traffic speed on the traffic network [1], based on the collection of localization data, e.g., vehicle speed, direction and time information using mobile phones or on-board GPS devices. This means that every vehicle acts as a sensor. In contrast to conventional methods such as cameras or street embedded sensors, no additional hardware on the traffic network is necessary. However, the reliability of travel time estimation based on FCD highly depends on the percentage of floating cars that participate in the traffic flow [2].

In this paper, we introduce an alternative method for monitoring traffic networks using mobile sensors. The mobile sensors here are unmanned aerial vehicles (UAVs) moving above the traffic network at a constant speed. The advantage of using mobile sensors is that they can freely move, not influencing or being influenced by the traffic situation. Moreover, they are not limited by physical restrictions of the network, e.g., by narrow roads or uneven terrain. In reality, each mobile sensor has a limited local field of view around itself, and cannot cover the whole domain of interest all the time. In order to monitor a large-scale traffic network with limited mobile sensor resources, all mobile sensors thus have to fly around to update the monitored traffic information. Therefore, in this paper we study the monitoring problem by using multiple mobile sensors. For the sake of simplicity, we assume that the mobile sensors can perfectly capture the traffic situation below them (e.g. by taking photos with a high-quality camera) when they are monitoring roads. Under this assumption, the main goal of this paper is to find *optimal covering paths* in a network for mobile sensors w.r.t. the minimal total travel time and the minimal total energy consumption.

The authors are with the Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands, email: {z.cong,b.deschutter,m.burger,r.babuska}@tudelft.nl

B. Related work

As will be explained in Section II below, the mobile sensor monitoring problem can be generalized as finding a least-cost tour[1] on a specified set of arcs in a graph. It is closely related to two types of arc routing problems — the Chinese postman problem and the rural postman problem.

The Chinese postman problem has been posed by Kwan [3]. Generally speaking, the goal is to seek a minimum-cost closed tour that visits all arcs of a graph. Hardgrave and Nemhauser [4] have shown that this problem can be immediately transformed into a traveling salesman problem [5], and then solved by dedicated traveling salesman algorithms. However, more direct approaches are also possible (see e.g. [6]). A comprehensive literature on this topic can be found in [7].

In real life, there are only a few practical contexts where it is necessary to service all arcs of a network, so most of the arc routing applications are usually modeled as rural postman problems. Given a graph and a subset of selected arcs on that graph, the rural postman problem is to find a closed tour on the graph, traversing each arc of the subset at least once, with the total cost minimized. This type of problems underlies several applications in contexts such as street sweeping [8], garbage collection [9], mail delivery [10], and so on. Interested readers are referred to [11] for detailed information.

The problem proposed in this paper is a variant of the rural postman problem, including two types of links for mobile sensors — one represents the physical roads that are going to be monitored (each of them must be visited once and only once) and the other type are the aerial links that are just used for traversal (they can be visited if needed). This problem will be translated into

[1] In this paper, a tour refers to a traversal from an origin to a destination for a mobile sensor.

a multiple traveling salesman problem, and solved by a mixed-integer linear programming (MILP) approach.

C. Contribution and paper structure

The main contributions of this paper are w.r.t. the literature are:

- We define the mobile sensor monitoring problem as a multiple rural postman problem; to the best knowledge of the authors, this has not been done before in the literature;
- We translate the multiple rural postman problem into a multiple traveling salesman problem, which is then solved by an MILP method;
- We also include the energy consumption as a performance criterion for the rural postman problem.

The rest of the paper is structured as follows. Section II presents the mobile sensor monitoring problem, by introducing a real traffic network and a coverage network. Section III mathematically formulates the optimization problem, by stating the objectives and constraints, which leads to a mixed-integer linear optimization problem. We test our approach and solve the proposed problem in a simulation-based case study in Section IV. Section V concludes the paper and gives some potential topics for future work.

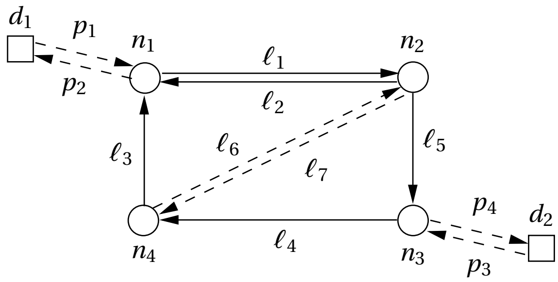
II. PROBLEM DESCRIPTION

In order to solve the mobile sensor monitoring problem, we consider two networks — one is the real traffic network, which has to be monitored by mobile sensors, and the other one is a virtual network defined according to the real network, used to solve the optimal covering paths problem. We will introduce these two networks in more detail next.

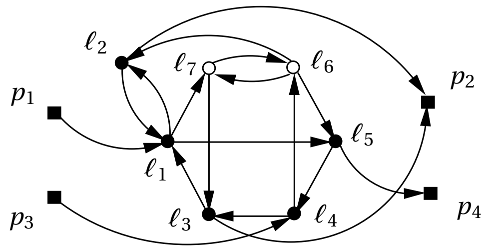
A. Physical network

We consider a real traffic network containing L_{road} physical roads to be monitored, and L_{air} aerial links that are only used for traversal by mobile sensors. Moreover, we have D depots, where mobile sensors start and finish their tours. Taking the network in Figure 1(a) as an example, the solid lines ℓ_1-ℓ_5 are physical roads, the dashed lines ℓ_6 and ℓ_7 are pure aerial paths, and the squares d_1 and d_2 are depots. Moreover, the circles n_1-n_4 represent nodes that connect physical roads or aerial links, and the dashed lines p_1-p_4 are virtual links connecting the real network and the depots.

For the sake of simplicity, we assume that each depot only has one incoming link and one outgoing link. In each depot d, there are N_d mobile sensors that can be used to monitor the traffic network. The total number of mobile sensors is therefore $N = \sum_{d=1}^{D} N_d$. Each mobile sensor is



(a) Real network



(b) Coverage network

Fig. 1. Example: mapping a real traffic network into a coverage network. The physical roads ℓ_1-ℓ_5 are mapped into vertices indicated by black dots, the pure aerial links ℓ_6 and ℓ_7 are mapped into vertices indicated by circles, and the depots d_1 and d_2 are mapped into vertices indicated by black squares.

considered as a fixed-wing UAV, which is typically modeled as a vehicle moving in a two-dimensional (2D) plane at a constant speed.

This setting corresponds to a multiple rural postman problem, where the L_{road} links corresponding to physical roads out of $L_{\text{road}}+L_{\text{air}}$ links in the network are required to be visited for surveillance purposes by N mobile sensors. In order to solve this problem, we transform it into a multiple traveling salesman problem [12] by mapping the real traffic network into a virtual network (see Section II-B) such that a mixed-integer linear programming solution method can be applied.

B. Coverage network

We call this virtual network a coverage network, which is defined according to the real traffic network. It is represented by a graph $\mathscr{G} = (\mathscr{V}, \mathscr{A})$, where $\mathscr{V}$ is the set of vertices associated with the links in the real network, and $\mathscr{A} = \{(i, j) : i, j \in \mathscr{V}\}$ is the set of arcs connecting the vertices.

The sets $\mathscr{V}$ and $\mathscr{A}$ are constructed as follows.

1) Vertex set: The vertices consist of three groups:

- The first group (denoted by the set $\mathscr{V}_{\text{moni}}$) represents the links that should be monitored by mobile sensors in the real network. This group refers to the physical roads in the real network, so the cardinality of the set $\mathscr{V}_{\text{moni}}$ is $|\mathscr{V}_{\text{moni}}| = L_{\text{road}}$. In this paper, we assume that the mobile sensors can perfectly monitor the traffic situation on each link, so the vertices in the first group only need to be visited once.

- The second group (denoted by the set $\mathcal{V}_{\text{travel}}$) represents the links in the real network that can be used by mobile sensors for traversal. Note that this group not only contains the L_{air} aerial links, but also the L_{road} physical roads, because mobile sensors can fly over them, too. Moreover, any vertex in this group is allowed to be visited multiple times for traveling purposes. However, since the optimization variable x_{ijn} (see its definition in Section II-B.2) only has a binary value, it is unknown how many times the vertices i and j are visited by a mobile sensor n. In order to tackle this issue, we may duplicate the vertices that correspond to the physical roads and aerial links multiple times. To illustrate this, let us define the set that contains the L_{air} aerial links and the L_{road} physical roads as $\mathcal{V}_{\text{travel,base}}$. For a vertex $i \in \mathcal{V}_{\text{travel,base}}$, we define a set $\mathcal{C}_i$ including both the vertex i itself and its corresponding duplications, $\mathcal{C}_i = \{i, i^{[2]}, \dots, i^{[C_i]}\}$, with $i^{[c]}$ the cth ($c \in \{2, 3, \dots, C_i\}$) duplication of vertex i, and C_i a constant indicating the maximum number of times that vertex i can be visited by the same mobile sensor. In this way, the set $\mathcal{V}_{\text{travel}}$ is defined as:

$$\mathcal{V}_{\text{travel}} = \bigcup_{i \in \mathcal{V}_{\text{travel,base}}} \mathcal{C}_i \tag{1}$$

- The third group represents the virtual links that connect depots and the real network. This group can be further divided into two subgroups: vertices that represent the outgoing links of the depots (e.g. p_1 and p_3 in Figure 1(b)) and vertices that represents the incoming links of the depots (e.g. p_2 and p_4 in Figure 1(b)). The first subgroup (denoted by the set $\mathcal{V}_{\text{orig}}$) can be considered as origin vertices, and the second subgroup (denoted by the set $\mathcal{V}_{\text{term}}$) can be considered as terminal vertices for mobile sensors. Since each depot only has one outgoing link and one incoming link, we have $|\mathcal{V}_{\text{orig}}| = |\mathcal{V}_{\text{term}}| = D$. Moreover, if vertex $t \in \mathcal{V}_{\text{term}}$ represents the incoming link of depot d and vertex $o \in \mathcal{V}_{\text{orig}}$ represents the outgoing link of depot d, we define the relationship between t and o as $t = \text{term}(o)$. For the coverage problem, we consider that initially mobile sensors are put on origin vertices, and the set of mobile sensors on a origin vertex is denoted by $\mathcal{N}_o$ with $o \in \mathcal{V}_{\text{orig}}$. Moreover, the set of all the mobile sensors are

$$\mathcal{N} = \bigcup_{o \in \mathcal{V}_{\text{orig}}} \mathcal{N}_o \tag{2}$$

As a result, we have the vertex set $\mathcal{V} = \mathcal{V}_{\text{moni}} \cup \mathcal{V}_{\text{travel}} \cup \mathcal{V}_{\text{orig}} \cup \mathcal{V}_{\text{term}}$. Taking Figure 1 as an example, links ℓ_1-ℓ_7 are mapped into intermediate vertices, links p_1 and p_3 are mapped into original vertices, and links p_2 and p_4 are mapped into terminal vertices.

2) Arc set: The arcs represent the physical restrictions of link connections in the real network. For example, in Figure 1, a mobile sensor can only move from link ℓ_1 to links ℓ_2, ℓ_5, and ℓ_7. If it is going to move from link ℓ_1 to link ℓ_4, it has to choose link ℓ_5 as a transmission link. Since all links in the real network are directed, there are two types of arcs connected to each vertex: incoming arcs and outgoing arcs. Therefore, we define a neighborhood $\mathcal{I}_i$ for each vertex $i \in \mathcal{V}$, which indicates the set of vertices that are directly connected with vertex i by one of its incoming arcs, and similarly a neighborhood $\mathcal{O}_i$ is defined to indicate the set of vertices that are directly connected with vertex i by one of its outgoing arcs. Moreover, a binary-valued variable x_{ijn} is associated with each arc (i, j), indicating whether ($x_{ijn} = 1$) or not ($x_{ijn} = 0$) vertex j is visited directly after vertex i by mobile sensor n.

III. PROBLEM FORMULATION

Using the network definitions and the variables described in the previous section, we can define the optimization problem of this paper. We first describe several objective functions, which define the terms we want to minimize. After that the constraints on the optimization variables are given.

A. Objective function

The objective function used in this paper establishes a trade-off between travel time and energy consumption for all the mobile sensors. Moreover, we assume that each mobile sensor has an associated fixed cost incurred whenever this mobile sensor is activated. A trade-off can be made between the different objectives by using the weighting variables $\alpha_1, \alpha_2, \alpha_3 \geqslant 0$ in the objective function

$$J = \alpha_1 J_{\text{energy}} + \alpha_2 J_{\text{time}} + \alpha_3 J_{\text{active}} \tag{3}$$

The definitions of J_{energy}, J_{time}, and J_{active} are given next.

1) *Energy consumption*: This is based on the dynamics of the mobile sensors. In this paper, we assume that the energy consumption is related to the speed of the mobile sensors, and that it is proportional to the length of the link in the real network. The energy consumed when a mobile sensor visits the vertex i is

$$e_i = f(u) \cdot l_i, \ \ \forall i \in \mathcal{V} \tag{4}$$

where f represents the relationship between the speed of the mobile sensor and energy consumption per unit distance, u denotes the speed of the mobile sensor, and l_i is the length of the real link that corresponds to the vertex i in the coverage network. The total energy consumption is

$$J_{\text{energy}} = \sum_{i \in \mathcal{V}} e_i \left(\sum_{j \in \mathcal{O}_i} \sum_{n \in \mathcal{N}} x_{ijn} \right) \tag{5}$$

Here, the factor $\sum_{n \in \mathcal{N}} x_{ijn}$ indicates the number of times that arc (i, j) is visited by mobile sensors.

2) *Travel time*: This depends on the length of each link in the real network, and the speed of the mobile sensor when it moves on that link. The time spent on vertex i is formulated as

$$\tau_i = \frac{l_i}{u}, \ \forall i \in \mathcal{V}, \tag{6}$$

with u the speed of the mobile sensor. The total travel time is formulated similarly as the total energy consumption:

$$J_{\text{time}} = \sum_{i \in \mathcal{V}} \tau_i \left(\sum_{j \in \mathcal{O}_i} \sum_{n \in \mathcal{N}} x_{ijn} \right) \tag{7}$$

3) *Fixed activation cost*: If a mobile sensor is used to monitor the traffic network, a fixed cost c_0 is assigned to that mobile sensor. The total cost of using mobile sensors is formulated as

$$J_{\text{active}} = c_0 \sum_{o \in \mathcal{V}_{\text{orig}}} \sum_{i \in \mathcal{O}_o} \sum_{n \in \mathcal{N}} x_{oin} \tag{8}$$

B. Constraints

1) Assignment constraints: First, for any origin vertex $o \in \mathcal{V}_{\text{orig}}$, there are N_o mobile sensors that can be used to monitor the network. However, we allow that some mobile sensors can stay at the depot without visiting a link in the network, and hence these mobile sensors are idle. This means that each arc $(o, i) \in \mathcal{A}$ with $o \in \mathcal{V}_{\text{orig}}, i \in \mathcal{O}_o$ can be visited by a mobile sensor n at most once:

$$\sum_{i \in \mathcal{O}_o} x_{oin} \leqslant 1, \ \forall o \in \mathcal{V}_{\text{orig}}, \forall n \in \mathcal{N}_o \tag{9}$$

If a mobile sensor enters a depot, it cannot exit again:

$$\sum_{i \in \mathcal{O}_o} x_{oin} = 0, \ \forall o \in \mathcal{V}_{\text{orig}}, \forall n \notin \mathcal{N}_o \tag{10}$$

When mobile sensors finish their tours and return to the depots, there are two different cases:

- **Case A**: A mobile sensor can return to any depot. In this case, we only need to ensure that the number of the mobile sensors does not exceed the capacity of the depot (it is interpreted as a terminal vertex capacity C_t):

$$N_o + \sum_{n \in \mathcal{N}} \sum_{i \in \mathcal{I}_t} x_{itn} - \sum_{n \in \mathcal{N}_o} \sum_{j \in \mathcal{O}_o} x_{ojn} \leqslant C_t, \quad \text{with } t = \text{term}(o), \ \forall o \in \mathcal{V}_{\text{orig}} \tag{11}$$

  Moreover, we should guarantee that the total number of outgoing mobile sensors equals the total number of returning mobile sensors:

$$\sum_{o \in \mathcal{V}_{\text{orig}}} \sum_{n \in \mathcal{N}_o} \sum_{j \in \mathcal{O}_o} x_{ojn} = \sum_{t \in \mathcal{V}_{\text{term}}} \sum_{n \in \mathcal{N}_o} \sum_{i \in \mathcal{I}_t} x_{itn}, \quad \text{with } t = \text{term}(o) \tag{12}$$

- **Case B**: Each mobile sensor must return to its original depot. We first should ensure the number of outgoing mobile sensors is the same as the number of returning mobile sensors for each depot:

$$\sum_{n \in \mathcal{N}_o} \sum_{j \in \mathcal{O}_o} x_{ojn} = \sum_{n \in \mathcal{N}_o} \sum_{i \in \mathcal{I}_t} x_{itn}, \quad \text{with } t = \text{term}(o), \ \forall o \in \mathcal{V}_{\text{orig}} \tag{13}$$

  Then we add a so-called cycle imposement constraint, which has been introduced by Burger et al. [13]. Generally speaking, this method associates a unique current k_o with each origin vertex o ($o \in \mathcal{V}_{\text{orig}}$), and let $k_t = k_o$ if $t = \text{term}(o)$, with k_t the current at terminal vertex t. If a vertex j is preceded by another vertex i, these two vertices share the same current $k_i = k_j$. In this way, the tour is imposed to make the mobile sensors return to the same depot as the one they started from:

$$k_{on} = o, \ \forall o \in \mathcal{V}_{\text{orig}}, \forall n \in \mathcal{N}_o \tag{14}$$

$$k_{tn} = k_{on}, \text{with } t = \text{term}(o), \tag{15}$$

$$k_{in} - k_{jn} \leqslant (D-1)(1 - x_{ijn} - x_{jin}), \ \forall i, j \in \mathcal{V} \tag{16}$$

  with D the number of depots, and where we assume without loss of generality that $\mathcal{V}_{\text{orig}} = \{1, 2, \ldots, D\}$.

A physical road $i \in \mathcal{V}_{\text{moni}}$ should be visited once and only once by any mobile sensor, which means that each vertex i is succeeded and preceded by exactly one vertex. We have

$$\sum_{n \in \mathcal{N}} \sum_{j \in \mathcal{O}_i} x_{ijn} = 1, \ \forall i \in \mathcal{V}_{\text{moni}}, \tag{17}$$

$$\sum_{n \in \mathcal{N}} \sum_{j \in \mathcal{I}_i} x_{jin} = 1, \ \forall i \in \mathcal{V}_{\text{moni}}, \tag{18}$$

For a link $i \in \mathcal{V}_{\text{moni}} \cup \mathcal{V}_{\text{travel}}$, we have to ensure the time that each mobile sensor n enters the given link the same number of times as the sensor exits that link:

$$\sum_{h \in \mathcal{I}_i} x_{hin} = \sum_{j \in \mathcal{O}_i} x_{ijn}, \ \forall i \in \mathcal{V}_{\text{moni}} \cup \mathcal{V}_{\text{travel}}, \forall n \in \mathcal{N} \tag{19}$$

2) Travel time constraints: As it is known from the traveling salesman literature [12], the assignment constraints mentioned above do not avoid subtours in the network, which means that tours could be formed between vertices in $\mathcal{V}_{\text{moni}} \cup \mathcal{V}_{\text{travel}}$ only and not be connected to any depot. One way to tackle this issue is to use so-called cycle elimination constraints. The idea behind these constraints is to assign an additional variable u_i to each vertex i, representing a vertex voltage. These vertex voltages have bounded values, and they increase at each vertex along the route until a terminal depot is reached. By using these constraints, if a subtour exists, then no terminal depot is included there, so the voltages at the vertices in this route will increase to infinity, which is a contradiction.

One of the well-known approaches is called the MTZ cycle elimination constraints approach, which was introduced by Miller et al. [14]. We follow this method in this paper. However, instead of using the vertex voltages, which have no practical meaning in our application, we introduce an arrival time variable T_{in}, which is the time that a mobile sensor n arrives at the vertex $i \in \mathcal{V}$, to act as the vertex voltage. The arrival time of each mobile sensor on the origin vertices is defined as:

$$T_{on} = 0, \ \ \forall o \in \mathcal{V}_{\text{orig}}, \forall n \in \mathcal{N} \tag{20}$$

If a vertex j directly succeeds a vertex i for the same mobile sensor, the arrival time T_{jn} is equal to the arrival time T_{in}, plus the travel time spent by mobile sensor n on vertex i.

$$T_{jn} = T_{in} + \tau_i, \ \text{ if } x_{ijn} = 1, \tag{21}$$

for all $i, j \in \mathcal{V}_{\text{moni}} \cup \mathcal{V}_{\text{travel}}$. Equation (21) is equivalent to

$$(T_{in} - T_{jn} + \tau_i) x_{ijn} = 0, \ \ \forall i, j \in \mathcal{V}_{\text{moni}} \cup \mathcal{V}_{\text{travel}}, \forall n \in \mathcal{N} \tag{22}$$

and it can be redefined as a linear inequality constraint by using the big-M method [15]

$$T_{in} - T_{jn} + \tau_i + M_{\text{t}} x_{ijn} \leqslant M_{\text{t}}, \tag{23}$$

$$T_{jn} - T_{in} - \tau_i + M_{\text{t}} x_{ijn} \leqslant M_{\text{t}}, \tag{24}$$

with M_{t} a large positive constant.

3) Energy level constraints: The energy level constraints guarantee that all mobile sensors return to the depots before their batteries are depleted or before they run of fuel. Similar to the travel time constraints, for each mobile sensor, we associated an initial energy level E_{on} by using

$$E_{on} = E_n^{\max}, \ \ \forall o \in \mathcal{V}_{\text{orig}}, \forall n \in \mathcal{N} \tag{25}$$

with $E_n^{\max}$ the maximum energy level that mobile sensor n can have. Using (4), we can determine the energy level of mobile sensor n visiting a vertex j directly after a vertex i: $E_{jn} = E_{in} - e_i$. Therefore,

$$(E_{in} - E_{jn} - e_i) x_{ijn} = 0, \ \ \forall i, j \in \mathcal{V}, \forall n \in \mathcal{N} \tag{26}$$

Similarly, by using the big-M method, we have

$$E_{in} - E_{jn} - e_i + M_{\text{e}} x_{ijn} \leqslant M_{\text{e}}, \tag{27}$$

$$E_{jn} - E_{in} + e_i + M_{\text{e}} x_{ijn} \leqslant M_{\text{e}}, \tag{28}$$

with M_{e} a large positive constant. We have to ensure that each mobile sensor will never run out of energy,

$$0 \leqslant E_{in} \leqslant E_n^{\max}, \ \ \forall i \in \mathcal{V}, \forall n \in \mathcal{N} \tag{29}$$

IV. CASE STUDY

We will demonstrate the use of the proposed method for the monitoring problem in an artificial network as shown in Figure 2. The MILP problem was implemented via the Tomlab toolbox for Matlab using CPLEX as the solver. Note that in this case study our main purpose is to illustrate our algorithm for finding optimal paths for mobile sensors to cover the target links in the traffic network. Therefore, the experiment does not involve any professional simulators for UAVs or traffic flows. These topics will be considered in the future work, as discussed in Section V.

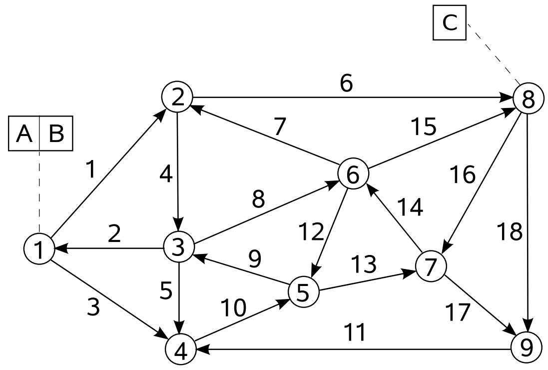


Fig. 2. Case study: a real traffic network with two depots. Mobile sensors A and B are put on one depot, and mobile sensor C are put on the other depot.

A. Simulation settings

This case study network has 38 links (only physical roads are shown in Figure 2), 9 nodes and 2 depots. The length of each link can be found in Table I. The first 18 links represent the physical roads, and the remaining 20 links are travel links.

We put mobile sensor A and B on the depot connected to node 1, and put mobile sensor C on the depot connected to node 8. The capacity of each depot is 2. We allow a mobile sensor to return to any depot when its tour is finished. For each mobile sensor, the initial arrival time (in [s]) is $T_0 = [0; 0]$, and the initial energy level (in [%]) is $E_0 = [100; 100]$. In this case study, the energy-speed function f in (4) is modeled by a second-order polynomial in the speed u, formulated as:

$$f(u) = au^2 + bu + c, \tag{30}$$

with u the speed, and $a, b, c \geqslant 0$ constants. The speed is defined with a fixed value $u = 120$ [km/h], and we choose the parameters as $a = 0.5$, $b = 2$, and $c = 1$. Moreover, the weights in (3) for the objective function are set as $\alpha_1 = 0.01$, $\alpha_2 = 1000$, and $\alpha_3 = 1$.

B. Simulation results

The simulation results are summarized in Table II. We can see that only mobile sensor A and C are activated, but mobile sensor B is idle. For both mobile sensor A and C, they still return to their own original depots although we do not enforce them to do so. Moreover, the travel time spent and the energy spent by each mobile sensor is similar to each other — the total travel time for sensor A and C is 1.76 hours versus 1.81 hours, and the energy level left at the end of the tour is 15.6% versus 13.2%.

TABLE I

LENGTHS OF LINKS

Link no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Start node	1	3	1	2	3	2	6	3	5	4	9	6	5	7	6	8	7	8
End node	2	1	4	3	4	8	2	6	3	5	4	5	7	6	8	7	9	9
Length (km)	22	12	18	18	8	34	18	21	11	13	30	18	15	19	24	25	13	27

Link no.	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38
Start node	5	1	1	8	4	2	2	1	2	3	3	9	4	7	4	5	5	9	1	2
End node	1	6	7	1	2	5	7	9	9	7	8	3	6	4	8	8	9	6	8	5
Length (km)	20	29	33	49	25	25	30	42	40	24	43	33	23	22	44	35	18	30	49	25

TABLE II

SIMULATION RESULTS OF CASE B

Mobile sensors	Link no.	Arrival time [h]	Energy level [%]
A	1	0	100
	6	0.18	91.2
	18	0.47	77.6
	11	0.69	66.8
	10	0.94	54.8
	9	1.05	49.6
	5	1.14	45.2
	31	1.21	42.0
	7	1.40	32.8
	38	1.55	25.6
	19	1.76	15.6
B	Not activated		
C	16	0	100
	14	0.21	90.0
	12	0.37	82.4
	13	0.52	75.2
	17	0.64	69.2
	30	0.75	64.0
	2	1.03	50.8
	3	1.13	46.0
	23	1.28	38.8
	4	1.48	28.8
	8	1.63	21.6
	15	1.81	13.2

V. CONCLUSIONS AND FUTURE WORK

In this paper we have formulated the mobile sensor monitoring problem as a multiple rural postman problem, where energy consumption is also taken into account. By translating our problem into a multiple traveling salesman problem, we can use mixed-integer linear programming to solve it. A simple case study has been provided to give an example of finding optimal coverage paths in the traffic network by using the proposed method.

For future work, we will first consider a fully dynamic monitoring problem. In this case, the mobile sensors move in the network not only based on the shortest distance and minimal energy consumption, but also based on the dynamic traffic situation. In addition, the dynamics of the UAVs also have to be considered. In order to do all this, UAV simulators (e.g. X-Plane or RMUS) and dynamic traffic flows simulators (e.g. OmniTRANS or VISSIM) will be used in the experiments. Moreover, we will compare the performance of our methods with that of other monitoring technologies such as FCD or fixed sensors. It is also interesting to consider additional approaches to solve the mobile sensor monitoring problem, e.g., direct solution methods for the multiple rural postman problem without making use of the transformation to the multiple traveling salesman problem.

ACKNOWLEDGMENT

Research supported by the China Scholarship Council (CSC), the European COST Action TU1102, and by the European Union Seventh Framework Programme [FP7/2007-2013] under grant agreement no. 257462 HYCON2 Network of Excellence.

REFERENCES

[1] B. S. Kerner, C. Demir, R. G. Herrtwich, S. L. Klenov, H. Rehborn, M. Aleksic, and A. Haug, "Traffic state detection with floating car data in road networks," in *Proceedings of the 8th International IEEE Conference on Intelligent Transportation Systems (ITSC 2005)*, 2005, pp. 44–49.

[2] C. D. Fabritiis, R. Ragona, and G. Valenti, "Traffic estimation and prediction based on real time floating car data," in *Proceedings of the 11th International IEEE Conference on Intelligent Transportation Systems (ITSC 2008)*, Beijing, China, 2008, pp. 197–203.

[3] M. Kwan, "Graphic programming using odd or even points," *Chinese Math*, vol. 1, no. 273-277, p. 110, 1962.

[4] W. W. Hardgrave and G. L. Nemhauser, "On the relation between the traveling-salesman and the longest-path problems," *Operations Research*, vol. 10, no. 5, pp. 647–657, 1962.

[5] E. L. Lawler, J. K. Lenstra, A. R. Kan, and D. B. Shmoys, *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*. Wiley New York, 1985, vol. 3.

[6] P. Miliotis, G. Laporte, and Y. Nobert, "Computational comparison of two methods for finding the shortest complete cycle or circuit in a graph," *RAIRO-Operations Research-Recherche Opérationnelle*, vol. 15, no. 3, pp. 233–239, 1981.

[7] H. A. Eiselt, M. Gendreau, and G. Laporte, "Arc routing problems, Part I: The Chinese postman problem," *Operations Research*, vol. 43, no. 2, pp. 231–242, 1995.

[8] L. D. Bodin and S. J. Kursh, "A computer-assisted system for the routing and scheduling of street sweepers," *Operations Research*, vol. 26, no. 4, pp. 525–537, 1978.

[9] E. Beltrami and L. D. Bodin, "Networks and vehicle routing for municipal waste collection," *Networks*, vol. 4, no. 1, pp. 65–94, 1974.

[10] L. Levy and L. Bodin, "The arc oriented location routing problem," *INFOR*, vol. 27, no. 1, pp. 74–94, 1989.

[11] H. A. Eiselt, M. Gendreau, and G. Laporte, "Arc routing problems, Part II: The rural postman problem," *Operations Research*, vol. 43, no. 3, pp. 399–414, 1995.

[12] T. Bektas, "The multiple traveling salesman problem: an overview of formulations and solution procedures," *Omega*, vol. 34, no. 3, pp. 209–219, 2006.

[13] M. Burger, B. De Schutter, and J. Hellendoorn, "Micro-ferry scheduling problem with time windows," in *Proceedings of the 2012 American Control Conference (ACC)*, 2012, pp. 3998–4003.

[14] C. E. Miller, A. W. Tucker, and R. A. Zemlin, "Integer programming formulation of traveling salesman problems," *Journal of the ACM*, vol. 7, no. 4, pp. 326–329, 1960.

[15] H. A. Taha, *Operations Research: An Introduction, 4th Ed.* Indianapolis, USA: Macmillan Publishing Company, 1987.