UNIVERSIDAD NACIONAL DE INGENIERÍA
FACULTAD DE CIENCIAS

TESIS

"VEHICLE ROUTING PROBLEM FOR INFORMATION COLLECTION IN WIRELESS NETWORK"

PARA OBTENER EL GRADO ACADÉMICO DE DOCTOR
EN CIENCIAS CON MENCIÓN EN MATEMÁTICA

ELABORADO POR :
LUIS ERNESTO FLORES LUYO

ASESOR :
Dr. ELADIO TEÓFILO OCAÑA ANAYA

Co-ASESORA :
Dra. ROSA MARÍA VIDEIRA DE FIGUEIREDO

LIMA–PERÚ
2018

Dedicatoria

A mis padres Luis e Isabel, a mis hermanos

A Lizeth, Caroline y Alberto.

AGRADECIMIENTOS

A mis padres, Luis Flores e Isabel Luyo, que me dieron la oportunidad de estudiar Matemática en la Universidad Nacional de Ingeniería, por sus consejos y por el amor tan grande que me han dado.
A mi familia Lizeth, Caroline y Alberto por su paciencia y comprensión.

A mi asesor de Perú, Dr. Eladio Ocaña Anaya por darme sabios consejos que fueron determinantes para obtener mi Licenciatura, Maestría y además por animarme a hacer mi doctorado.

A mi asesora de Francia Dra. Rosa Figueiredo por guiarme y darme las herramientas necesarias para poder culminar mi trabajo y por sus sabias aclaraciones .

A Agostinho Agra un investigador Portugués que admiro mucho y que fue pieza clave en la finalización de mi trabajo.

A mis amigos de CERI Moez, Bruno, Thiago, Marco, Bouaziz, Waad, Elvys, Driss, Zied, Chadine, Abir y Olfa.

Agredecer además a Concytec, ya que gracias al financiamiento de Cienciactiva he podido realizar mis estudios de doctorado.

RESUMEN

En esta tesis damos un primer acercamiento al problema de construir rutas de vehículo para optimizar el recojo de información generada en las estaciones, vía física y wireless. Construimos un primer modelo matemático MIP y se propone tres posibles funciones objetivos, las cuales serán comparadas. Para este primer modelo asumimos que no es posible enviar la información por partes, es decir, se envía toda la información o no se envía nada, además veremos que este modelo solo puede resolver de forma exacta hasta un máximo de 10 estaciones y con un tiempo $T = 30$, lo cual sugiere encontrar mejores modelos.

En este trabajo construimos también otros tres modelos matemáticos, modelo discreto, modelo visitas, modelo evento, todos ellos permiten enviar una parte de la información acumulada en una estación cercana hacia el vehículo, estos modelos serán comparados de acuerdo a su velocidad, en estos modelo podemos exhibir algunas instancias de 20 estaciones y un tiempo de T igual a 72 y otra de 8 estaciones y un tiempo de 240.

Debido a que en los problemas reales el número de estaciones es mayor necesitamos de métodos no exactos llamado heurísticas, las cuales nos permite obtener soluciones cercanas a la exacta, en este trabajo daremos algunas heurísticas como heurística greedy, heurística de inserción, heurística fix and relax, heurística de intercambio, y por último haremos comparaciones entre ellas de acuerdo a la velocidad y a la calidad de la solución.

Abstract

The vehicle routing problem is one of the most studied problems in Operations Research. Different variants have been treated in the past 50 years and with technological advances, new challenges appear. In this thesis, we introduce a new variation of the VRP appearing in wireless networks. The new characteristic added to this well-know problem is the possibility of pick-up information via wireless transmissions. In the context considered here, a unique base station is connected with the outside and a vehicle is responsible for collecting information via wireless connection to the vehicle when it is located in another sufficiently close station. Simultaneous transmissions are permitted. Time of transmission depends on the distance between stations, the amount of information transmitted, and other physical factors (e.g obstacles along the way, installed equipment). Information to be sent outside of the network is continuously generated in each station at a constant rate. The first contribution of this thesis is the introduction of a mixed ILP formulation for a variation in which it is only possible to send all the information or nothing during a wireless transmission. For this model three different strategies are investigated: maximizing total amount of information extracted an the end of the time horizon; maximizing the average of the information in the vehicle at each time point; and maximizing the satisfaction of each station at the end of the time horizon. Each strategy is translated as a different objective function for the mixed ILP formulation. The problem is then reformulated by accepting the option of sending only part of the information during a wireless transmission and considering only the first strategy,(i.e. maximizing the amount of information extracted at the end of the horizon time). For this new version, we present three mixed ILP formulations, each one with advantages and disadvantages. These mixed ILP models are compared according to the CPU time, amount of information collected, gap of unresolved instances, etc. Because in real life we need to solve problems with a large number of stations, in this thesis, we also propose heuristics methods for the second version of the problem introduced. We build some heuristics that do not depend on the mixed ILP model (as for example Greedy heuristics) and also matheuristcs. In our matheuristics our best model (a vehicle event model) is used as a base for the development of construction of Heuristics as well as local search heuristics.

Contents

Introduction

"The beauty of mathematics
only shows itself to more patient
followers. "

Maryam Mirzakhani.

The intensive research on VRP is due not only to its computational complexity but also to the numerous applications in fields such as logistic, maritime transportation, telecommunications, production, among many others. Different variants of the VRP have been treated in the past 50 years Among the best known we can cite the VRP with Heterogeneous fleet (Baldacci et al., 2008), the VRP with Multiple Depots (Montoya-Torres et al., 2015), the Pickup-and-delivery VRP (Dethloff, 2001), the Stochastic VRP (Gendreau et al., 1996), the VRP with Time Windows (Agra et al., 2013; Bräysy et Gendreau, 2005), the VRP with Backhauls (Toth et Vigo, 1997), the Dynamic VRP (Psaraftis, 1995). Several hybrid variants of the problem are also described in the literature, most of them inspired by real-life scenarios

Technological advances in network architectures add new features and applications to routing problems (An et al., 2015; Kavitha et Altman, 2009; Velásquez-Villada et al., 2014; Płaczek, 2012). In this thesis, we are interested in adding to this well-know problem the possibility of pick-up information via wireless transmission, this new problem has many applications,for example in underwater surveillance, (Basagni et al., 2014), they address an application of underwater monitoring involving a set of S surfacing nodes and $|S|$ underwater stations, see figure 1 from (Basagni et al., 2014). They look for a routing to an autonomous underwater vehicle (AUV) during a time period T. The AUV must leave and return to a surface node while information generated by the set of underwater nodes is collected along a path that physically visits each underwater station where information is collected. The strategy adopted by the authors is the maximization of the value of the information collected considering that the value of the information reduces along the time. They propose an Integer Linear Programming (ILP) formulation able to solve the problem with $|S| \in \{4, 5, 9, 12\}$ in a time that varies from a few hours to a few days.

Another application of this work is in the area of delay tolerant network (DTN) (Fall, 2003; Jain et al., 2004), DTN are wireless networks designed to tolerate long delays or disruptions. In a DTN, disconnections can occur due to mobility of nodes and/or fail-

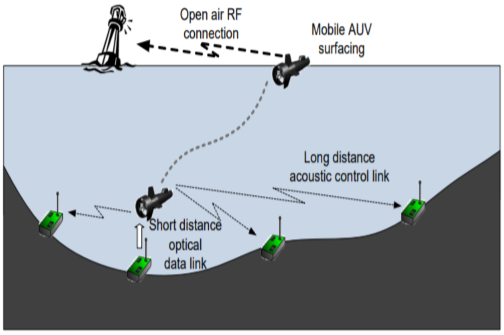


Figure 1 – Maximizing the Value of Sensed Information in Underwater Wireless Sensor Networks via an Autonomous Underwater Vehicle.

ures of energy, which means a communication path between a source node and a destination node is not guaranteed to exist. Such network architecture was proposed to provide connectivity in difficult environments, which include, discontinuous connectivity, variable transmission rate, restrictions of energy or other resources. One of the applications of DTN's that has gained popularity is to provide web service to remote locations. For example, Daknet (Pentland et al., 2004) is a network for rural connectivity that uses buses and local transport to carry messages and web connection to small and remote villages. Daknet establishes small kiosks in rural villages without web connectivity, allowing people to send email and information in an off-line manner. The existent transport infrastructure is used to provide data transmission between a remote village and the connected world. Another important application of DTN is to provide connection for remote military stations which dispose of a set of vehicles to pick-up and deliver wireless information (Malowidzki et al., 2016; Redi et Ramanathan, 2011).

This work could be applied in Wireless Sensor Networks (WSN) (Akyildiz et al., 2002),WSN are often used in critical applications such as habitat monitoring, war surveillance, submarine surveillance and monitoring, detection of biological, chemical and nuclear attack (Mainwaring et al., 2002; Winkler et al., 2008; Basagni et al., 2014; Vieira et al., 2015). In these applications, sensors are deployed in a area and are used to store information to be sent it the future to a control center via base stations. Classically, data is retransmitted by the sensors to the base stations through a multi-hop routing protocol (Biswas et Morris, 2004). However, problems can occur with multi-hop routing. First, an excessive number of jumps reduces the lifetime of sensors in applications where network failure is a critical factor. Second, as the number of base stations is limited, nodes near the base stations are left without much energy before the other sensors causing a non-uniformity in energy consumption. Mobile elements, such as unmanned aerial vehicles, have been incorporated in the WSN design to solve these two problems (Teh et al., 2008).

The last two applications mentioned here from both technologies, DTN and WSN, need to provide vehicle routing strategies with wireless information transmission to the vehicles involved. Innovation and research appears most in the development of routing protocols (Bhoi et al., 2017; Celik et Modiano, 2010; Moghadam et al., 2011; Velásquez-Villada et al., 2014) while there is still a gap in the development of vehicle routing strategies. When developing efficient routing protocols, many authors consider that the vehicle route is already defined which is justified by applications taking advantage of an existing transportation infrastructure (Velásquez-Villada et al., 2014). In (Celik et Modiano, 2010), the vehicle route is assumed to exist but authors suppose vehicles can adjust their position (in order to receive information from some stations) and study the delay performance in the network. Other works, like (Kavitha et Altman, 2009), suppose an architecture is defined for the vehicle routing (cycle path or zig-zag path) and study the best placement for such architecture. To the best of our knowledge, the authors in (Basagni et al., 2014) are the only ones to investigate the vehicle routing problem from scratch together with the wireless transmission planning. They address an application of underwater wireless sensor networks for submarine monitoring. The authors considered a scenario with a set of S surfacing nodes and $|S|$ underwater nodes where they look for a routing to an autonomous underwater vehicle (AUV) during a time period T. The AUV must leave and return to a surface node while information generated by the set of underwater nodes is collected along a path that physically visits each station where information is collected. The information generated in a given underwater node i at a time point t_1 which arrives to a surface node at a time point t_2 has a given value $v_{it_1t_2}$. The strategy adopted by the authors is the maximization of the value of the information collected. The authors in (Basagni et al., 2014) propose an Integer Linear Programming (ILP) formulation able to solve the problem with $|S| \in \{4, 5, 9, 12\}$ in a time that varies from a few hours to a few days.

Organization of the thesis

The thesis is organized as follows : In the first part of this thesis an introduction presented our motivation to study the problem considered in this thesis. Also, the state of the art and the applications of the problem have been presented.

In the first chapter a MILP formulation to the problem is presented in Section 1.3 with three different objective functions being discussed at this section. Computational experiments are presented on Section 1.4. Periodicity on the remaining information at the stations after a sequence of vehicle routings is solved is discussed in Section 1.4.1.

The chapter two is organized as follows. Section 2.2 formally describes the VRP being solved while notations and assumptions are presented. Three MILP formulations to the problem are presented in Section 2.3 with the discussion on how the models can be strengthened. Computational experiments are presented on Section 2.4. Finally, some conclusions and research directions are presented at Section 3.3.2.

In the third and last chapter we construct heuristics that will allow giving approximate solutions to the optimum in a considerably low time, most of the heuristics discussed in this chapter work with the vehicle model event seen in chapter two.

Finally, we present the conclusions of the thesis and some perspectives.

Chapter 1

Vehicle routing problem for information collection in wireless networks

Contents

1.1 Introduction

Advances in computer network architecture add continuously new features to vehicle routing problems. In this work, the Wireless Transmission Vehicle Routing Problem (WT-VRP) is studied. It looks for a route to the vehicle responsible for collecting information from stations as well as an efficient information collecting planning. The new feature added here is the possibility of picking up information via wireless transmission, without visiting physically the stations of the network. The WT-VRP has applications in underwater surveillance and environmental monitoring. We discuss three criteria for measuring the efficiency of a solution and propose a mixed integer linear programming formulation to solve the problem. Computational experiments were done to access the numerical complexity of the problem and to compare solutions under the three criteria proposed.

1.2 Description of the problem

The wireless network is modeled by a directed graph $D = (V, A)$. The node set $V = \{1, \ldots, n\}$ represents the n stations of the network and the arc set A represents m directed paths connecting pair of stations in V. A base station is regarded as node 1. Weights t_{ij} and d_{ij} are associated to each arc (path) $(i, j) \in A$ representing, respectively, the time it takes to travel from station i to station j and the distance among these two stations. Let $T = \{1, 2, \ldots, \bar{T}\}$ be the time horizon considered. At the beginning of the time horizon, each station $j \in V \setminus \{1\}$ contains an amount C_j of information. For each station $i \in V \setminus \{1\}$, information is generated at a rate of r_j units per time point in T. Thus, the amount of information at station j at each time point $k \in T$, denoted by q_{jk}, is proportional to the elapsed time from the last extraction (either physically or through a wireless connection), i.e.,

$$q_{jk} = \begin{cases} C_j + kr_j, & \text{if station } j \text{ has not been visited before time point } k, \\ (k - t_{last})r_j, & \text{otherwise, where } t_{last} \text{ is the time of the last extraction.} \end{cases}$$

Only the base station is appropriated equipped to sending information outside the network. A unique vehicle is in charge of collecting data from all the stations in $V \setminus \{1\}$ and of transporting it to the base station. There is no capacity limit associated to the vehicle. At the beginning of the time horizon, the vehicle is located at the base station and at the end of the time horizon, it must return to the base station. Multiple visits are allowed to each node in V. Information can only be transmitted when the vehicle is located in one of the stations in V, i.e., no transmission is allowed while the vehicle is moving on an arc $(i, j) \in A$. We also assume that, once a station i starts a transmission to the vehicle, all the current information located in i at that moment must be transmitted.

Wireless transmission can be used to transfer data from a station $j \in V$ to the vehicle located in a station $i \in V \setminus \{j\}$. However, wireless transmission is only possible for close enough stations. Let r_{cov} be the maximum distance allowing wireless transmission. A station j can wireless transfer its data to (the vehicle located in) station i whenever $d_{ij} \leq r_{cov}$. We define a set $W = \{(i,j) \in V \times V \mid d_{ij} \leq r_{cov}\}$.

We assume that a transmission occurs with a fixed transmission power of P_t. The received power P_r is given by

$$P_r = \alpha P_t D^{-\eta}$$

where η is the pathloss parameter (which we shall take in this paper to be equal 2 but our results carry on to arbitrary positive value of η) and where D is the distance between the receiver and transmitter. We shall assume that the vehicle have an antenna with an elevation of 1 unit. The coordinates of a sensor are given by $(x_s, y_s, 0)$ and that of the vehicle are $(x_b, y_b, 1)$, which means, the antenna on the vehicle is elevated by one unit with respect to the sensors. Thus, if $d = \sqrt{(x_s - x_b)^2 + (y_s - y_b)^2}$ then $D = \sqrt{1 + d^2}$. We use a linear approximation of the Shannon capacity (as a function of the power) for the data transmission rate (Tse et Viswanath, 2005) and write it as

$$Thp(d) = \log\left(1 + \frac{\alpha P_r}{\sigma}\right) \sim \frac{\beta P_r}{2\sigma} = \frac{\beta P_t}{2\sigma} D^{-2} = \frac{\beta P_t}{2\sigma}(1 + d^2)^{-1}.$$

Here β is the Antenna's gain and σ is the noise at the receiver (we assume independent channels and thus there are no interferences of other transmissions on the received signal from our sensor).

Thus, the time necessary for a wireless transmission of $\bar{q}$ units of data between stations j and i is

$$\alpha_{ji}(1 + d_{ij}^2)\bar{q}, \tag{1.1}$$

with $\alpha_{ji} = \dfrac{\beta P_t}{2\sigma}$ depending on physical factors in stations i and j.

Simultaneous transmission is possible from a set of at most M stations to the vehicle located in a station $i \in V$. In this case, the simultaneous data transfer finishes only when each individual wireless transmission finishes. As a consequence, the time of a simultaneous transmission corresponds to the highest maximum individual wireless transmission.

The version of the VRP treated in this paper, denoted as Wireless Transmission VRP (WT-VRP), consists of finding a feasible routing for the vehicle together with an efficient planning for collecting information from stations $V \setminus \{1\}$. The criteria for measuring the efficiency of a collection planning will be discussed in the next section.

1.3 Mathematical formulation

In this section, we introduce a mixed integer linear programming (MILP) formulation to the WT-VRP. Let us define an artificial arc set $A^0 = \{(i,i) \in V\}$. For each $(i,j) \in A \cup A^0$

and $k \in T$, we define the following decision variables.

$$x_{ijk} = \begin{cases} 1, & \text{if the vehicle crosses path } (i,j) \text{ and arrives to node } j \text{ at time point } k, \\ 0, & \text{otherwise.} \end{cases}$$

We observe that the action of crossing an artificial arc $(i,i) \in A^0$ models the following action: the vehicle is located at node i without neither moving nor transferring data. Let $t = (t_{ij})$ be the weight matrix associated with A. We also define $\hat{t} = t + I$ the weight matrix associated with $A \cup A^0$. For each $(j,i) \in W$, $k \in T$ and $l \in S_k = \{1, \ldots, \bar{T} - k\}$, we also define,

$$w_{jikl} = \begin{cases} 1, & \text{if node } j \text{ is sending data to the vehicle while it is located in} \\ & \text{node } i\text{, with transmission starting at time point } k \\ & \text{and lasting } l \text{ time units,} \\ 0, & \text{otherwise.} \end{cases}$$

Linear constraints defining the MILP formulation are presented next, divided in three sets according to their modeling purposes.

1.3.1 Problem constraints

The first set, the *Routing Constraints*, characterizes the way the vehicle moves around the set of stations.

$$\sum_{s:(1,s) \in A \cup A^0} x_{1s(1+t_{1s})} = 1, \tag{1.2}$$

$$\sum_{(s,1) \in A \cup A^0} x_{s1\bar{T}} = 1, \tag{1.3}$$

$$x_{ijk} \leq \sum_{(j,p) \in A \cup A^0} x_{jp(k+t_{jp})} + \sum_{(u,j) \in W} \sum_{l \in S_k} w_{ujkl}, \qquad (i,j) \in A \cup A^0, \forall k \in T, \tag{1.4}$$

$$x_{ij(k+t_{ij})} \leq \sum_{(u,i) \in A \cup A^0} x_{uik} + \sum_{n | (n,i) \in W} \sum_{l=1}^{k-1} w_{ni(k-l)l}, \quad \forall (i,j) \in A \cup A^0, \forall k : k + t_{ij} \in T, \tag{1.5}$$

$$\sum_{l \in S_k} w_{jikl} \leq \sum_{(i,n) \in A \cup A^0} \sum_{s \in S_k} x_{in(k+s+t_{in})}, \qquad i \in V, \forall j \text{ s.t. } (j,i) \in W, \forall k \in T, \tag{1.6}$$

$$\sum_{(i,j)\in A} x_{ijk} \leq 1, \qquad \forall k \in T, \tag{1.7}$$

$$\sum_{s=k+1}^{k+t_{ij}-1} \sum_{(m,n)\in A\cup A^0} x_{mns} \leq t_{ij}(1 - x_{ij(k+t_{ij})}), \qquad (i,j) \in A,\ \forall k \in T : k + t_{ij} \in \bar{T}, \tag{1.8}$$

$$x_{ijk} \in \{0,1\}, \quad \forall (i,j) \in A, \qquad \forall k \in T, \tag{1.9}$$

$$w_{uvkl} \in \{0,1\}, \quad \forall (u,v) \in W, \qquad \forall k \in T,\ \forall l \in S_k. \tag{1.10}$$

Equations (1.2) and (1.3) ensure, respectively, that the vehicle starts and ends the routing at the base station. Inequalities (1.4) ensure that, if at a time point $k \in T$ the vehicle arrives at station $j \in V$ crossing arc $(i,j) \in A \cup A^0$, it either goes immediately to a neighboring station (crossing an appropriate path $(j,p) \in A \cup A^0$), or it stays at node j for a data transfer from another station $u \in V$. Likewise, if at a time point $k + t_{ij} \in T$ the vehicle arrives to station $j \in V$ coming from station i, inequalities (1.5) impose that either the vehicle has arrived to station i at time point k, or a data transfer from at least one another station was occurring and it has finished at time point k. Once the vehicle has arrived to station $i \in V$, at a time point $k \in T$, if data transfer happens, inequalities (1.6) force the vehicle to leave i, in a future time point $k + s$, by crossing an arc $(i,n) \in A \cup A^0$. Inequalities (1.7) and (1.8) are responsible for the elimination of simultaneous paths. On one hand, inequalities (1.7) force the vehicle always to leave a given station along a unique path. On the other hand, if the vehicle cross the arc (i,j), leaving i at time point k, an inequality in (1.8) will prevent the vehicle from moving from time point $k+1$ to $k + t_{ij} - 1$. Constraints (1.9) and (1.10) impose binary conditions on the variables defined.

Inequalities defining the second set of constraints are the *Data Transfer Constraints*.

$$\frac{1}{M} \sum_{(j,i)\in W} \sum_{l\in S_k} w_{jikl} \leq 1, \qquad \forall k \in T, \tag{1.11}$$

$$w_{jikl} \leq \sum_{(p,i)\in A} x_{pik}, \qquad \forall (j,i) \in W,\ \forall k \in T,\ \forall l \in S_k, \tag{1.12}$$

$$\sum_{(i,n)\in A} \sum_{s=0}^{l-1} x_{in(k+s+t_{in})} \leq l(1 - w_{jikl}), \qquad \forall (j,i) \in W,\ \forall k \in T,\ \forall l \in S_k. \tag{1.13}$$

Inequalities (1.11) define a bound of M simultaneous data transfers. Inequalities (1.12) impose that, at each time point $k \in T$, in order to start sending information to a station $i \in V$, the vehicle must previously arrive to this station. Inequalities (1.13) prevent the vehicle to leave station $i \in V$ while a data transfer is occurring: if w_{jikl} equals to 1, the vehicle cannot leave station i from time point k to time point $k + l - 1$. When simultaneous data transfers occurs, this set of inequalities will be in charge of defining the total duration of the simultaneous transfer.

Before presenting the last set of constraints, we need to define the continuous variables of the MILP formulation. For each $j \in V$ and $k \in T$, let q_{jk} represent the amount of data accumulated at station j (waiting for transfer out of the network) at time point k. The last set of constraints is presented next and they are the *Amount of Information Constraints.*

$$q_{j1} = C_j, \qquad \forall j \in V \setminus \{1\}, \tag{1.14}$$
$$q_{jk+1} = q_{jk}(1 - \sum_{i|(j,i)\in W} \sum_{l \in S_k} w_{jikl}) + r_j, \qquad \forall k \in T \setminus \{\bar{T}\}, \forall j \in V, \tag{1.15}$$
$$\alpha_{ji}(1 + d_{ij}^2) q_{jk} w_{jikl} \leq l, \qquad \forall (j,i) \in W,\ \forall k \in T,\ \forall l \in S_k. \tag{1.16}$$

Equations (1.14) set the initial load of each station $j \in V \setminus \{1\}$. Equations (1.15) are in charge of update the load of stations along the time horizon. The amount of data accumulated in node j, at time point $k+1$, $q_{j(k+1)}$, is set to r_j in the case a data transfer started at time point k, otherwise it equals to $q_{jk} + r_j$. Finally, inequalities (1.16) define the time necessary for transferring q_{jk} data units (as defined by (1.1)) whenever $w_{jikl} = 1$.

Constraints (1.15) and (1.16) are quadratic ones and could be linearized by applying a classical change of variables (Wolsey, 1998). An alternative way of linearizing these constraints is by replacing them with the following big-M inequalities:

$$q_{j(k+1)} \geq q_{jk} + r_j - M_{jk} \sum_{j|(j,i)\in W} \sum_{l \in S_k} w_{jikl}, \qquad \forall k \in T \setminus \{\bar{T}\}, \forall j \in V, \tag{1.17}$$
$$q_{jk} \geq r_j, \qquad \forall j \in V,\ \forall k \in T \tag{1.18}$$
$$\alpha_{ji}(1 + d_{ij}^2) q_{jk} - N_{jik}(1 - w_{jikl}) \leq l, \qquad \forall (j,i) \in W, \forall k \in T,\ \forall l \in S_k, \tag{1.19}$$

where

$$M_{jk} = r_j(k+1), \qquad j \in V,\ k \in T,$$
$$N_{jik} = \alpha_{ji}(1 + d_{ji}^2) r_j k, \qquad (j,i) \in W,\ \forall k \in T.$$

Consider a node $j \in V$ and a time point $k \in T$. If a transfer occurs from station j at time point k, the associated inequality in (1.17) becomes redundant and the associated inequality in (1.18) defines the valid lower bound $q_{jk} \geq r_j$. On the other hand, if no transfer occurs, the associated inequality in (1.18) becomes redundant and a valid lower bound $q_{j(k+1)} \geq q_{jk} + r_j$ is defined by (1.17). A minimization objective function of the WT-VRP (discussed in the next subsection) together with inequalities (1.17) and (1.18) will be in charge of appropriately setting the value of variables q_{jk}. In the same way, an inequality in (1.19) is either redundant ($w_{jikl} = 0$) or it becomes an inequality in (1.16) ($w_{jikl} = 1$).

1.3.2 Objective function

As it has been described in Section 2.2, the WT-VRP looks for an efficient way of collecting data located in the set of remote stations. We discuss in this section three different criteria to measure the efficiency of a collection planning, each one giving birth to a different linear objective function. The first one maximizes the total amount of information extracted at the end of the time horizon T; the second one maximizes the average of the information in the vehicle at each time point; while the third one maximizes the satisfaction of each station at the end of the time horizon T.

Let us analysis the first criteria. Consider first that our goal is the extraction of as much information as possible from all the stations, at the end of the finite time horizon T. Let η_j be the quantity of information extracted from a given station $j \in V \setminus \{1\}$, i.e.,

$$\eta_j = C_j + r_j(\bar{T} - 1) - q_{j\bar{T}}.$$

The amount of information extracted from all stations in $V \setminus \{1\}$ is

$$\sum_{j \in V\setminus\{1\}} \eta_j = \sum_{j \in V\setminus\{1\}} C_j + \sum_{j \in V\setminus\{1\}} r_j(\bar{T} - 1) - \sum_{j \in V\setminus\{1\}} q_{j\bar{T}}.$$

Since, in this equation, the only variables from our MILP formulation, introduced in Section 1.3, are the q_{jT} variables, for $j \in V$, we have that

$$\max \left\{ \sum_{j \in V\setminus\{1\}} \eta_j \right\} = \sum_{j \in V\setminus\{1\}} C_j + \sum_{j \in V\setminus\{1\}} r_j(\bar{T} - 1) - \min \left\{ \sum_{j \in V\setminus\{1\}} q_{j\bar{T}} \right\}.$$

The first objective function, denoted **FO1**, minimizes the remaining amount of information in the end of the time period, over the set of stations, i.e.,

$$\text{Minimize} \sum_{j \in V\setminus\{1\}} q_{j\bar{T}}. \tag{1.20}$$

For the first criterion, no assumption is made about data location security: only the amount of information collected at the end of the time horizon matters. The second efficiency criterion is motivated by environments where information is safer once they leave the station to be stored at the vehicle (Basagni et al., 2014). Among the factors behind this supposition we have failures in station equipments, attack in case of military applications and energy resources limited. Let v_k be the amount of information in the vehicle at a time point $k \in T$,

$$v_k = \sum_{j \in V\setminus\{1\}} C_j + \sum_{j \in V\setminus\{1\}} r_j(k - 1) - \sum_{j \in V\setminus\{1\}} q_{jk}.$$

We look for a solution, i.e., a routing and a collection planning, that maximizes the average over time of the amount of information in the vehicle, i.e.,

$$\begin{aligned}
\max\left\{\frac{1}{\bar{T}}\sum_{k\in T} v_k\right\} &= \max\left\{\frac{1}{\bar{T}}\sum_{k\in T}\left(\sum_{j\in V\setminus\{1\}} C_j + \sum_{j\in V\setminus\{1\}} r_j(k-1) - \sum_{j\in V\setminus\{1\}} q_{jk}\right)\right\}\\
&= \max\left\{\frac{1}{\bar{T}}\left((\sum_{j\in V\setminus\{1\}} C_j)\bar{T} + (\sum_{j\in V\setminus\{1\}} r_j)\frac{(\bar{T}-1)\bar{T}}{2} - \sum_{k\in T}\sum_{j\in V\setminus\{1\}} q_{jk}\right)\right\}\\
&= \max\left\{\left(\sum_{j\in V\setminus\{1\}} C_j + (\sum_{j\in V\setminus\{1\}} r_j)\frac{(\bar{T}-1)}{2} - \frac{1}{\bar{T}}\sum_{k\in T}\sum_{j\in V\setminus\{1\}} q_{jk}\right)\right\}\\
&= \sum_{j\in V\setminus\{1\}} C_j + (\sum_{j\in V\setminus\{1\}} r_j)\frac{(\bar{T}-1)}{2} - \frac{1}{\bar{T}}\min\left\{\sum_{k\in T}\sum_{j\in V\setminus\{1\}} q_{jk}\right\}.
\end{aligned}$$

The second objective function, denoted **FO2**, minimizes the total amount of remaining information on the set of stations, over the hole time horizon, i.e.,

$$\text{Minimize} \sum_{k\in T}\sum_{j\in V\setminus\{1\}} q_{jk}. \tag{1.21}$$

In the third criterion, the satisfaction of each station will be taken into account. Let us define the satisfaction of a station $j \in V$ as the total amount of information extracted from j over the amount of information generated at j during the time horizon considered, i.e.,

$$s_j = \frac{C_j + r_j(\bar{T}-1) - q_{j\bar{T}}}{C_j + r_j(\bar{T}-1)} = 1 - \frac{q_{j\bar{T}}}{C_j + r_j(\bar{T}-1)}.$$

The maximization of the satisfaction over the hole set of stations is

$$\begin{aligned}
\max\left\{\frac{1}{n-1}\sum_{j\in V\setminus\{1\}} s_j\right\} &= \max\left\{\frac{1}{n-1}\sum_{j\in V\setminus\{1\}}\left(1 - \frac{q_{j\bar{T}}}{C_j + r_j(\bar{T}-1)}\right)\right\}\\
&= 1 - \frac{1}{n-1}\min\left\{\sum_{j\in V\setminus\{1\}}\frac{q_{j\bar{T}}}{C_j + r_j(\bar{T}-1)}\right\}.
\end{aligned}$$

For the particular case where $C_j = r_j$, we have

$$\max\left\{\frac{1}{n-1}\sum_{j\in V\setminus\{1\}} s_j\right\} = 1 - \frac{1}{(n-1)\bar{T}}\min\left\{\sum_{j\in V\setminus\{1\}}\frac{q_{j\bar{T}}}{r_j}\right\}.$$

Finally, the third objective function, denoted **FO3**, which maximizes the satisfaction of the stations, is define as,

$$\text{Minimize} \sum_{j\in V\setminus\{1\}}\frac{q_{j\bar{T}}}{r_j}. \tag{1.22}$$

Consider the instance of the WT-VRP depicted in Figure 1.1 with five non-base stations. Figures 1.2–1.4 illustrate the solutions obtained by solving the MILP formulation, with the three different objective functions, assuming $\alpha_{ij} = 0.03$, for each $i, j \in V$, $r_{cov} = 2$, $\bar{T} = 20$ and $M = 3$. The movement of the vehicle is illustrated in each one of these figures by a time expanded network. Each set of nodes aligned horizontally is associated to a same station $i \in V$ in different time points. Each set of nodes aligned vertically represents the hole set of stations in a unique time point $k \in T$. Let us denote a node in the time expanded network by a pair ik, with $i \in V$ and $k \in T$. The scalar value inside a node ik gives us the amount of information in station i at time point k. A filled arrow from a node ik to a node jl represents a movement of the vehicle, leaving station i at time point k and arriving to station j at time point l. A dashed arrow from a node ik to a node jk indicates a data transfer occurs from station i to station j, starting in a time point after k. A weight associated with a dashed arc indicates the duration of the transfer.

$$d = \begin{bmatrix} 0 & 4 & 5 & 4 & 7 & 7 \\ 4 & 0 & 2 & 4 & 3 & 6 \\ 5 & 2 & 0 & 2 & 2 & 1 \\ 4 & 4 & 2 & 0 & 5 & 1 \\ 7 & 3 & 2 & 5 & 0 & 2 \\ 7 & 6 & 1 & 1 & 2 & 0 \end{bmatrix} \qquad \bar{t} = \begin{bmatrix} 1 & 4 & 5 & 4 & 0 & 0 \\ 4 & 1 & 2 & 0 & 3 & 0 \\ 5 & 2 & 1 & 2 & 2 & 1 \\ 4 & 0 & 2 & 1 & 0 & 1 \\ 0 & 3 & 2 & 0 & 1 & 2 \\ 0 & 0 & 1 & 1 & 2 & 1 \end{bmatrix}$$

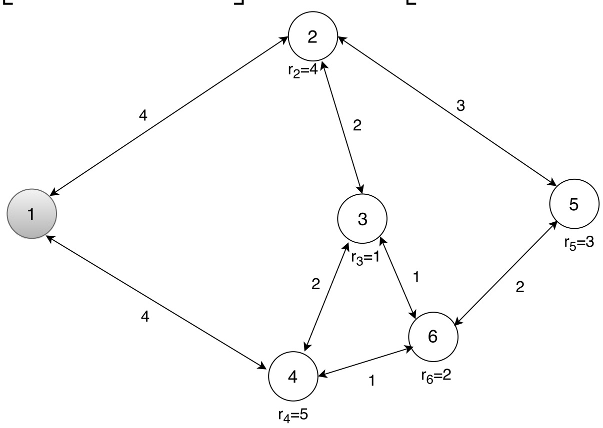


***Figure 1.1** – Directed (symmetric) graph describing an instance with 6 stations, data rates $r = (0, 4, 1, 5, 3, 2)$, time matrix $\bar{t}$ and distance matrix d.*

The vehicle routing depicted in Figure 1.2 is $1 \to 2 \to 3 \to 6 \to 4 \to 1$. The vehicle leaves the base station 1 at time point 1 and arrives to station 2 at time point 5 with transmission from stations 2 and 3 starting immediately, each lasting 1 time unit. Then, the vehicle leaves station 2 at time point 6 arriving at station 6 at time point 9 for data transfer from stations 4, 5 and 6; data transfer at station 6 has lasts $\max\{5, 3, 3\}$ time units. The vehicle leaves station 6 at time point 14 and continues its routing and data transfers until returning at time point 20 to the base station 1. The total amount of information generated during the hole time horizon at each station, respectively from station 2 to 6, is $80, 20, 100, 60, 40$. For the solution depicted in 1.2, the remaining information at each station is, respectively, $60, 15, 25, 33, 10$ which means, the

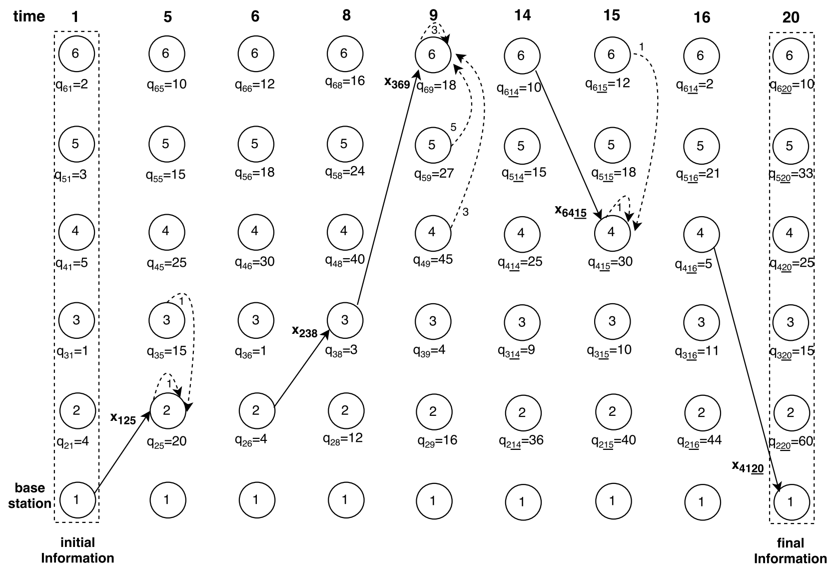


Figure 1.2 – Solution obtained for the instance depicted in Figure 1.1 by the MILP formulation with objective function FO1 defined in (1.20).

total amount of information collected from each station is, respectively, 20, 5, 75, 26, 30. For this solution, the value of the three different objective functions are: 143 for **FO1**, 1575 for **FO2** and 51 for **FO3**.

The vehicle routing depicted in Figure 1.3 choses for not visiting stations 2 and 5; wireless transmissions are used to collected information from these stations. For this solution, the value of the three different objective functions are: 162 for **FO1**, 1536 for **FO2** and 48 for **FO3**. The vehicle routing depicted in Figure 1.4 choses for visiting each station in the network. This improves the satisfaction over the hole set of stations though it increases the amount of remaining information in the network. For this solution, the value of the three different objective functions are: 145 for **FO1**, 1598 for **FO2** and 45 for **FO3**.

1.4 Computational experiments

In this section, we report computational experiments carried out with the formulation presented in Section 1.3 comparing the solutions obtained with the different objective functions proposed. The MILP problems were solved by IBM CPLEX Optimizer 12.6.1.0 on a server with a 16 processor Intel®Xeon® Processor E5640 12M Cache, 2.67 GHz and 32 GB of RAM memory. The CPU time limit was set to 1h for all instances. Before presenting the obtained results, we briefly describe the set of instances used in our experiments.

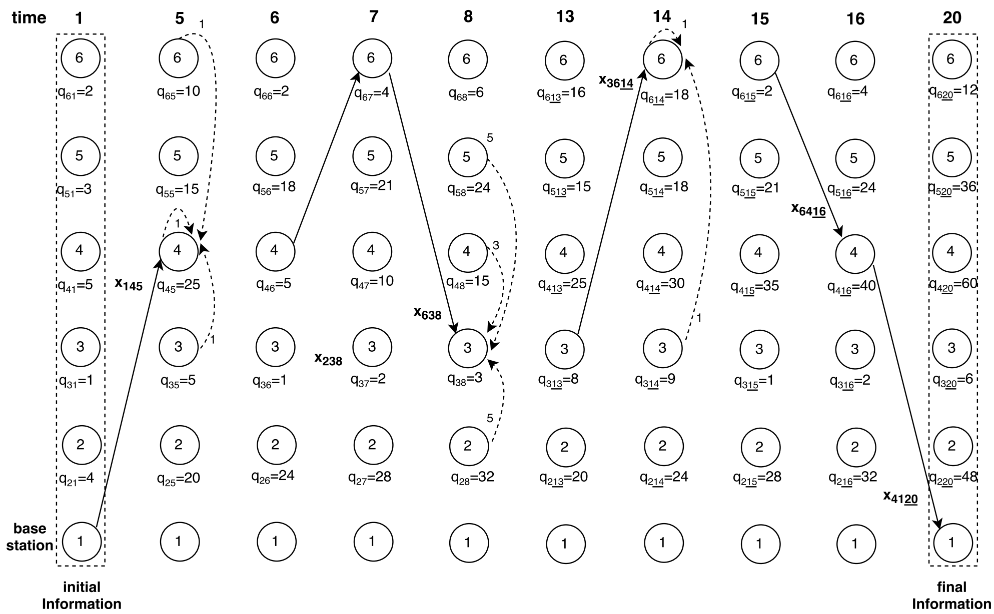


***Figure 1.3** – Solution obtained for the instance depicted in Figure 1.1 by the MILP formulation with objective function FO2 defined in (1.21).*

Instances

We evaluate the MILP formulation on a set of 80 instances. Each instance is defined by a random graph $D = (V, A)$ and a given value $\bar{T}$. A set of 40 random graphs were generated as follows.

Let n and $\bar{\Delta}$ be, respectively, the total number of vertices and a border on the density of the graph to be generated. On a square of length B, the base station is located in the bottom-left vertex and the other $n-1$ stations are placed randomly on the square of length $B-2$ in the upper-right (as shown in the figure (1.5)). We define A as the adjacency matrix associated with the complete graph defined by the n stations. Let $\Delta(D)$ denote the density of the graph $D = (V, A)$. We randomly select an arc $(i, j) \in A$ such that $D = (V, A \setminus (i, j))$ is a connected graph. We define $A = A \setminus (i, j)$. In case $\bar{\Delta} - \Delta(D) > 0.01$ we proceed with the random elimination of arcs; otherwise we stop and return the graph $D = (V, A)$. Finally, a rate r_j is randomly generated for each $j \in V \setminus \{1\}$. The distance matrix $\mathbf{d}$ is defined by using the euclidean distance between each pair of vertices located in the square of side B. The time matrix $\mathbf{t}$ is defined according to the adjacent matrix considering a vehicle of speed equal to 1 which means, $t_{ij} = d_{ij}$ if $(i, j) \in A$, $t_{ij} = 0$ otherwise. We generate graphs with number of vertices $n \in \{6, 8, 10, 12\}$ and upper bounds on the graph density $\bar{\Delta} \in \{0.5, 0.7\}$. For each combination of n and $\bar{\Delta}$, five random graphs are generated according to the procedure just described. In our experiments we considered the set of graphs just described together with $\bar{t} \in \{24, 28\}$.

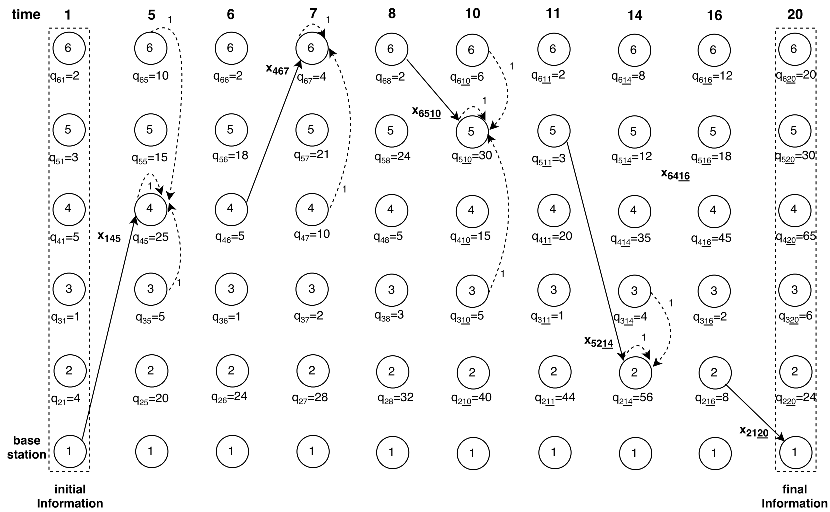


Figure 1.4 *– Solution obtained for the instance depicted in Figure 1.1 by the MILP formulation with objective function FO3 defined in (1.22).*

1.4.1 Objective functions comparison

Tables 1.1 and 1.2 exhibit the results obtained with the three objective functions on the set of 80 random instances. In both tables, the first multicolumn exhibits information about the instances: $|V|$ is the number of stations, d is the graph density, $\bar{T}$ is the size of the time horizon considered. The fourth column informs the total amount of information generated in the network during the hole time horizon. The next five multicolumns display information about the solutions obtained with each different objective function: "$CpuTime$" is the time, in seconds, spent to solve the instance to optimality ("$-$" means the instance was not solved in the time limit); Gap is the MILP gap calculated between the best integer solution found and the final lower bound (a $Gap = 0$ means the solution was solved to optimality in the time limit); "$Inf.collected$" display the total amount of information collected by the vehicle during the hole time horizon; "$Av.Vehicle$" display the average amount of information in the vehicle over time; "$Av.Satisfaction$" display the average satisfaction over the set of stations.

Table 1.1 presents results obtained on instances with 6 and 8 stations and show, as we expected, the sensibility of the ILP formulation to the total number of stations in the network. In general, instances with more than eight stations cannot be solved in one hour of computation. Looking the optimal solutions obtained, data transmissions took no more than three time units. Thus, we solved a set of instances with 8,10 and 12 stations by limiting data transmissions to four time units which reduces the number of binary variables in the formulation: variables w_{jikl} are defined for $l \leq 4$. Table 1.2 presents the results obtained on instances with 10 and 12 stations.

From results on these two tables, we conclude that the WT-VRP problem became more difficult as the number of stations increases as well as the time horizon increases. From Table 1.2, a total of 2 and 17 MILP problems were not solved to optimality, re-

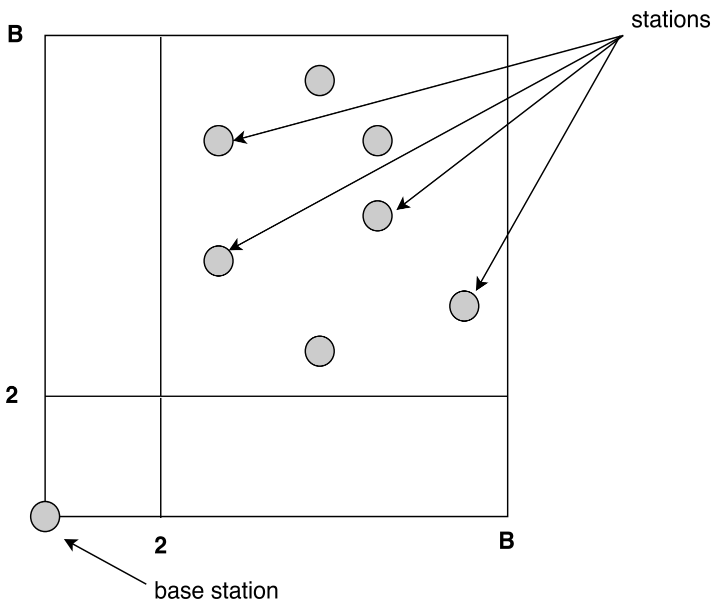


Figure 1.5 *– Schema used for random graph generation: the base station is located in the bottom-left vertex; the other stations are randomly placed in the square located in the upper-right.*

spectively, for $T = 24$ and $T = 28$ with gaps arriving to 30% when $T = 28$. From Table 1.1, a total of 7 and 29 MILP problems were not solved to optimality, respectively, for $T = 24$ and $T = 28$ with gaps arriving to 69% when $T = 28$. The graphics in Figure 1.6 displays the results obtained on the 47 instances solved to optimality by all three objective functions. In each graphic, the instances are ordered in increasing order of the average of the time spent to solve the three ILP problems. The three graphics from the top compare the optimal solutions obtained according to the value of "$Inf.Collected$", "$Av.Vehicle$" and "$Av.Satisfaction$". We can observe that, in fact, the WT-VRP problem became more difficult as the total amount of information collected from the network increases. From our results, we conclude the MILP formulation defined with objective function **FO2** is computationally easier: less instances not solved to optimality and, in average, less time spending and smaller gaps. From both tables, the total number of MILP not solved to optimality in the time limit is equal to 22, 11 and 22, respectively, for objective functions **FO1**, **FO2** and **FO3**. We can also observe that, on one hand, the solutions maximizing the average satisfaction of the network impacts almost equally the average amount of information in the vehicle and the total amount of information collected. On the other hand, the solutions maximizing the average amount of information in the vehicle over time, impacts slightly more the average satisfaction of the network than the solutions maximizing the total amount of information collected.

***Table 1.1** – Results obtained on the set of instances with $n = 6, 8$ and $T = 24, 28$.*

|V|	d	T̄	$\sum r_j T$	Cpu Time			Gap			Inf. collected			Av. Vehicle			Av. Satisfaction		
				FO1	FO2	FO3	FO1	FO2	FO3	FO1	FO2	FO3	FO1	FO2	FO3	FO1	FO2	FO3
6	0,5	24	312	1,64	1,77	1,84	0	0	0	62	50	44	28,33	29,16	16,41	0,24	0,16	0,25
			264	3,21	6,8	3,26	0	0	0	129	129	129	65,45	65,45	65,45	0,5	0,5	0,5
			408	2,83	4,14	2,89	0	0	0	111	111	111	57,45	59,45	59,45	0,32	0,32	0,32
			432	26,75	9,37	5,87	0	0	0	98	94	94	47,08	52,58	52,58	0,28	0,29	0,29
			312	3,39	3,91	3,9	0	0	0	140	130	96	68,33	71,08	50,75	0,35	0,34	0,4
		28	364	4,01	3,6	3,5	0	0	0	134	134	132	76,21	76,21	71,5	0,27	0,27	0,39
			308	34	67,48	49,51	0	0	0	161	157	161	80,14	83,64	82,71	0,55	0,52	0,55
			476	78,42	55,88	76,95	0	0	0	174	166	170	91,57	93,78	91,35	0,4	0,37	0,42
			504	187,75	77,63	125,07	0	0	0	176	176	172	93,14	93,14	80,71	0,28	0,28	0,36
			364	59,65	72,37	80,29	0	0	0	193	193	184	100,78	100,78	92,71	0,46	0,46	0,5
	0,72	24	384	3,55	3,18	3,49	0	0	0	110	110	96	64,16	64,16	54	0,25	0,25	0,27
			408	2,79	2,45	2,59	0	0	0	104	104	104	56,66	56,66	56,66	0,31	0,31	0,31
			312	2,94	2,33	2,6	0	0	0	67	56	48	27,79	30	21	0,23	0,11	0,25
			240	57,61	19,91	68,77	0	0	0	141	141	136	65	65	62,7	0,57	0,57	0,58
			336	20,34	14,48	40,29	0	0	0	178	176	178	90,16	95,5	90,16	0,48	0,4	0,48
		28	448	51,46	10,92	11,23	0	0	0	164	164	164	88,28	88,28	88,28	0,4	0,4	0,4
			476	6,78	7,69	6,46	0	0	0	172	172	172	87,57	87,57	87,57	0,38	0,38	0,38
			364	13,22	11,14	11,8	0	0	0	105	105	98	52,17	53,46	42,78	0,2	0,21	0,35
			280	—	898	—	4,25	0	16,99	186	186	174	89,89	89,89	78,53	0,67	0,67	0,64
			392	2988,72	332,88	—	0	0	22,95	245	236	239	123,89	127,96	120,89	0,56	0,48	0,56
			369,20	186,79	80,30	27,80	4,25	0	19,97	142,5	139,5	135,1	72,70	74,19	68,31	0,39	0,36	0,41
8	0,53	24	672	7,97	7,48	10,26	0	0	0	144	144	142	85,16	85,16	77,16	0,19	0,19	0,22
			480	28,63	21	60,62	0	0	0	157	157	142	82,29	82,29	70,79	0,3	0,3	0,3
			528	296,01	83,89	202,91	0	0	0	184	174	158	89	92	77,08	0,35	0,32	0,36
			456	5,68	5,31	4,92	0	0	0	120	120	117	70	70	64,12	0,17	0,17	0,23
			480	32,85	20,98	27,27	0	0	0	179	167	179	88,54	90,79	88,54	0,35	0,29	0,35
		28	784	177,97	177,79	181,17	0	0	0	220	212	196	119	121,85	104,25	0,28	0,24	0,29
			560	1667,62	422,09	532,58	0	0	0	244	221	244	118,14	120,1	118,14	0,41	0,35	0,41
			616	—	—	—	23,42	15,27	13,39	236	236	212	118,67	118,67	105	0,45	0,45	0,42
			532	139,86	85,39	305,35	0	0	0	212	199	212	103,78	108,25	103,78	0,33	0,31	0,33
			560	—	1160,55	2225,25	30,89	0	0	259	245	259	124,32	130,28	124,32	0,44	0,38	0,44
	0,71	24	528	—	1102,72	—	17,53	0	14,03	220	214	208	97,5	114,41	105,91	0,35	0,41	0,41
			552	9,4	17,21	9,58	0	0	0	132	116	123	70,25	71,66	51,62	0,24	0,22	0,24
			504	44,75	42,79	50,36	0	0	0	169	152	169	83,04	83,33	83,04	0,36	0,33	0,36
			408	11,51	11,54	24,65	0	0	0	121	114	118	57,91	63,5	52,5	0,3	0,29	0,3
			480	137,78	91,97	339,78	0	0	0	169	169	168	81,04	85,04	72,91	0,35	0,35	0,35
		28	616	—	—	—	17,11	32,65	29,77	300	294	254	123	146,89	123,85	0,43	0,4	0,43
			644	—	933	—	16,36	0	29,54	224	224	224	114,85	114,85	114,85	0,32	0,32	0,32
			588	—	1889,39	—	14,13	0	16,1	213	199	213	105,89	108,5	105,89	0,39	0,33	0,39
			476	—	602,05	1458,9	26,25	0	0	166	161	166	72,07	86,89	73,57	0,37	0,36	0,37
			560	—	3124,41	—	17,47	0	5,83	217	205	215	98,96	107,32	93,17	0,39	0,36	0,38
			551,20	213,34	544,42	388,11	20,40	23,96	18,11	194,3	186,15	185,95	95,17	100,09	90,52	0,34	0,32	0,35

Table 1.2 *– Results obtained on the set of instances with* $n = 8, 10, 12$ *and* $T = 24, 28$.

|V|	d	T̄	$\sum r_j T$	Cpu Time			Gap			Inf. collected			Av. Vehicle			Av. Satisfaction		
				FO1	FO2	FO3	FO1	FO2	FO3	FO1	FO2	FO3	FO1	FO2	FO3	FO1	FO2	FO3
8	0,53	24	672	3,15	3,4	3,08	0	0	0	126	117	90	59,16	61,12	46,25	0,16	0,15	0,17
			480	13,66	4,44	13,96	0	0	0	157	157	157	82,29	82,29	82,29	0,3	0,3	0,3
			528	59,3	13,49	41,53	0	0	0	184	174	158	89	92	77,08	0,35	0,32	0,36
			456	2,55	3,12	2,69	0	0	0	98	98	90	50,45	52,5	47,25	0,16	0,16	0,17
			480	9,02	11,01	18,89	0	0	0	179	167	143	88,54	90,79	66,04	0,35	0,29	0,35
		28	784	26,68	13	74	0	0	0	179	179	179	95,89	95,89	94,6	0,26	0,26	0,26
			560	532,59	136,04	1128,2	0	0	0	244	221	244	118,14	120,1	118,14	0,41	0,35	0,41
			616	—	432,9	—	8,33	0	17,09	232	232	210	120,57	120,57	102,28	0,38	0,38	0,4
			532	51,35	10,93	55,54	0	0	0	212	199	212	103,78	108,25	103,78	0,33	0,31	0,33
			560	2233,91	170,94	1300,59	0	0	0	259	256	259	124,32	129,28	124,32	0,44	0,43	0,44
			566,8	325,80	79,93	293,16	8,33	0	17,09	187	180	174,20	93,21	95,28	86,20	0,31	0,30	0,32
10	0,52	24	576	6,49	8,02	10,07	0	0	0	140	121	124	69,16	69,54	59,16	0,2	0,15	0,21
			576	5,44	7,39	19,5	0	0	0	164	164	104	79,66	79,66	45,04	0,25	0,25	0,25
			480	31,04	21,45	103,6	0	0	0	123	123	123	62,54	62,54	62,54	0,26	0,26	0,26
			576	113,72	69,44	265,94	0	0	0	152	147	143	71,83	75	67,54	0,28	0,26	0,32
			720	5,11	7,67	11,46	0	0	0	168	164	168	85	87,5	85	0,19	0,18	0,19
		28	672	268,62	194,93	499,66	0	0	0	208	208	135	101,71	101,71	73,96	0,26	0,26	0,27
			672	499,44	342,44	532,86	0	0	0	232	216	194	110,71	115,28	97,5	0,28	0,26	0,3
			560	1607,21	766,31	—	0	0	24,33	220	218	220	107,28	111,28	107,28	0,38	0,36	0,38
			672	3103,83	462,65	—	0	0	22,21	224	224	212	117	117	95,42	0,32	0,32	0,39
			840	1210,58	348,76	1393,03	0	0	0	228	228	205	115,28	117,28	104,39	0,22	0,22	0,25
10	0,66	24	696	414,41	147,61	841,84	0	0	0	207	193	207	100,75	101,87	100,75	0,32	0,3	0,32
			768	15,88	19,67	28,5	0	0	0	160	160	152	75,83	75,83	68,16	0,24	0,24	0,26
			648	596,51	207,01	285,6	0	0	0	167	167	157	79,12	80,16	65,45	0,29	0,29	0,3
			696	82,49	32,37	186,65	0	0	0	172	172	132	86,66	86,66	68,33	0,23	0,23	0,24
			576	1550,84	3583,52	1640,02	0	0	0	200	184	130	97,33	97,83	59,58	0,35	0,31	0,36
		28	812	—	—	—	41,54	31,11	23,07	257	251	268	129,82	135,1	127,42	0,34	0,33	0,38
			896	—	651,33	2041,84	3,8	0	0	257	257	230	130,82	130,82	117,46	0,28	0,28	0,31
			756	—	—	—	42,37	25,06	52,99	218	210	219	93,42	104,46	90,03	0,3	0,31	0,32
			812	—	2904,71	—	42,98	0	46,16	270	270	237	136,42	136,42	120,67	0,29	0,29	0,34
			672	—	—	—	52,96	61,34	47,37	234	248	195	113,92	118,57	95,96	0,34	0,34	0,42
			683,8	634,11	575,02	561,47	36,73	39,17	36,02	200,05	196,25	177,75	98,21	100,23	85,58	0,28	0,27	0,30
12	0,51	24	840	476,05	178,77	407,48	0	0	0	190	175	190	89,33	95,79	89,33	0,22	0,2	0,22
			648	488,24	551,27	1742,87	0	0	0	238	228	207	116,83	122,25	106,29	0,24	0,25	0,26
			960	266,89	189,44	738,49	0	0	0	218	218	208	107,91	111,25	101,66	0,23	0,23	0,23
			936	—	3339,79	—	47,44	0	6,04	208	242	228	102,91	116,91	102,41	0,28	0,32	0,31
			744	—	—	—	54,07	32,36	24,69	235	245	269	108,2	126,41	116,2	0,32	0,34	0,38
		28	980	—	—	—	62,1	57,17	52,24	231	221	217	103	116,96	95,46	0,19	0,21	0,26
			756	—	—	—	51,36	58,88	57,01	275	274	240	131,6	140,25	108,14	0,24	0,24	0,25
			1120	—	—	—	47,3	19,98	56,54	323	310	263	159,32	159,14	127,21	0,26	0,25	0,25
			1092	—	—	—	63,54	65,23	62,81	290	283	274	136,5	150,57	105,21	0,33	0,24	0,31
			868	—	—	—	64,99	69,61	59,71	276	302	296	124,32	147,96	126,21	0,33	0,35	0,38
			894,4	410,39	1064,82	962,95	55,83	50,54	45,58	248,40	249,80	239,20	117,99	128,75	107,81	0,26	0,26	0,29

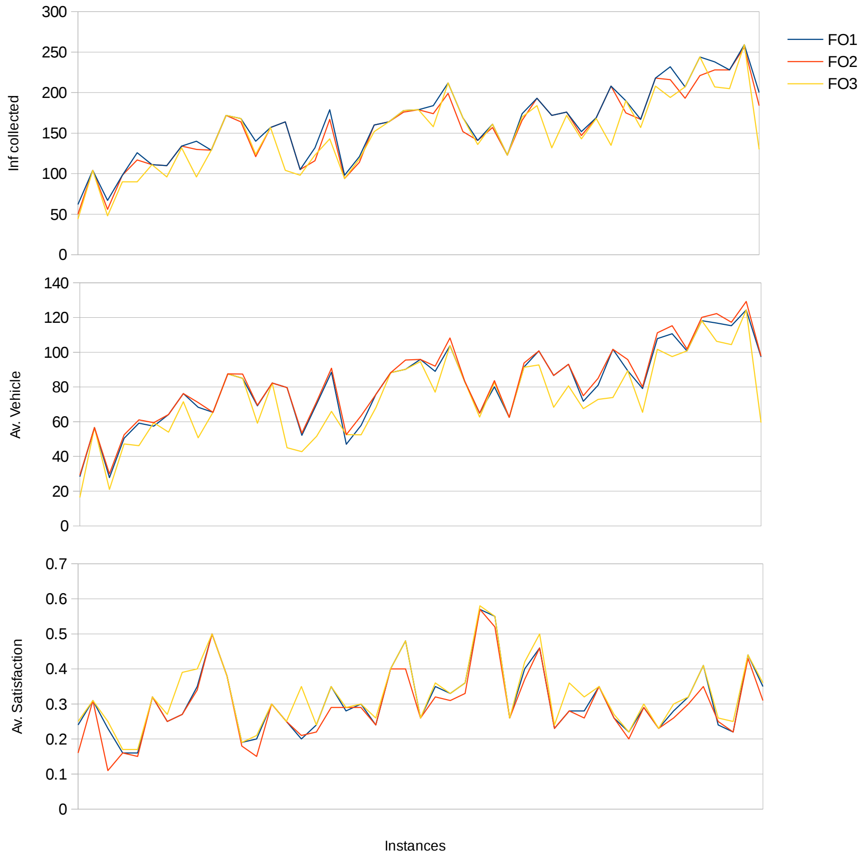


Figure 1.6 *– Results obtained on the set of 47 instances solved to optimality with all objective functions.*

Periodicity

We compare now the optimal solutions obtained with each objective function with respect to the information left in the network at the end of a time period T. Let $C^0 = [C^0_0, C^0_1, \ldots, C^0_n]$ be a vector with the amount of information stored in each station $i \in V$. By considering the vector C^0 as the initial conditions in the MILP formulation, we obtain a routing for the vehicle, a collection planning and the amount of remaining information in each station, denoted here as $C^1 = [C^1_0, C^1_1, \ldots, C^1_n]$. In that way, the MILP formulation associated with an instance can be seen as a function $F : \mathbb{R}^n \longrightarrow \mathbb{R}^n$ such that $F(C^0) = C^1$. In order to study the dynamics of function F, let us define $C^m = F^m(C^0)$ where $F^k(C) = F(F^{k-1}(C))$. We experimentally investigate the existence of values $k, \tau \in \mathbb{N}$ such that $F^\tau(C^k) = C^k$ starting with the initial conditions $C^0 = \mathbf{0}$. When such values exist, we say that the function F is periodic.

For this experiment, we use all the instances solved in less than 300s by our MILP formulation (with all three different objective functions). Consider an instance of the problem and a MILP formulation, i.e. a function $F : \mathbb{R}^n \longrightarrow \mathbb{R}^n$. Starting with $C^0 = \mathbf{0}$, we solve this instance and obtain $F(C^0) = C^1$. Then, we continue to iterate until we discover that the function is periodic or we have solved the MILP formulation 30 times.

Table 1.3 displays the results obtained. The first multicolumn exhibits information about the instances used in this experiment as defined for the previous tables. The other three multicolumns inform us the results obtained for each objective function: the values obtained for k, τ and the number of isolated stations #isol. An isolated station i is a station such that, in the successive applications of function F, we arrive to a given $\bar{k}$, such that $C_i^{k'} > C_i^{\bar{k}}$, for each $k' > \bar{k}$. That means, the amount of information accumulated at station i at iteration $\bar{k}$ arrives to a value that the optimal solution of the MILP formulation (due to the size of the time period $\bar{T}$ and the assumption that once a transmission starts all the information in the station must be transmitted) is not able to reduce it. Thus, station i became isolated from outside of the network. The entry "$-$" for k and τ means the solution became periodic for a subset of stations but with the presence of isolated stations. Likewise, the entry "$*$" for k and τ means after the limit of 30 iterations no periodicity was achieved. From the results in Table 1.3, we can conclude that the formulation maximizing the average satisfaction of stations is more suitable to achieve periodicity and with smaller values of k.

***Table 1.3** – Results obtained when the periodicity of the solutions obtained by the ILP formulations is studied.*

|V|	**d**	$\bar{\mathbf{T}}$	**FO1**			**FO2**			**FO3**		
			k	τ	#isol	k	τ	#isol	k	τ	#isol
6	0.50	28	–	–	1	–	–	1	–	–	1
			–	–	1	–	–	1	–	–	1
			7	5	0	7	2	0	5	3	0
			–	–	1	–	–	1	–	–	1
			4	2	0	3	2	0	3	1	0
6	0.5	30	2	3	0	3	3	0	2	3	0
			–	–	1	–	–	1	–	–	1
			2	1	0	5	1	0	2	5	0
			6	4	0	–	–	1	6	3	0
			4	1	0	3	2	0	5	1	0
8	0.53	32	–	–	1	–	–	1	–	–	1
			*	*	0	*	*	0	5	3	1
			*	*	0	*	*	0	*	*	0
			3	2	0	7	4	0	5	2	0
			12	4	0	*	*	0	1	2	0

Chapter 2

MIP models for a vehicle routing problem with information collection in wireless networks

Contents

2.1 Introduction

In this chapter we study a routing-collecting problem where a system of n stations is considered. A vehicle is responsible for collecting information generated continuously in the stations and to deliver it to the base station. The objective is to determine the vehicle route and the collection operations, both physical and wireless, in order to maximize the amount of information collected during a time horizon.

In the chapter 1 it is assumed that, once a station starts a transmission to the vehicle, all the current information accumulated in this stations at that moment must be transmitted. This assumption is justified in a scenario where information is safer once it leaves the stations. However, we observed that this imposition contributes for the disconnection of some stations. In this chapter, we do not make this assumption and we set as a strategy for the vehicle routing problem the maximization of the total amount of information extracted at the end of the time horizon T. Adding new aspects to a problem can impact the mathematical modeling choices (Cancela et al., 2015) and our assumptions allowed us to develop differents MILP formulations to the problem.

We develop three different MILP formulations to the problem, one based on a time discretization, where each decision is a multiple of the time unity, and two using continuous time and based on events. The two formulations based on continuous time differ in the type of events considered. One formulation assumes the events are visits to stations and transfer operations, and the other assumes that events are the vehicle stops. We will see that each such model proposed in this chapter have pros and cons making it more suitable for a particular instance. To the best of our knowledge these models have never been introduced for a VRP with information collection in wireless networks. The most related work is the comparison of discrete time models with continuous time models presented in (Agra et al., 2017) for a maritime inventory routing problem; although the conclusions are not coincident to the ones in this paper as the problems are different.

The problem treated in this chapter is very close to the problem defined in chapter 1. In order to make clear the differences, In the next section we give the complete problem definition of the problem considere in this chapter.

2.2 Description of the problem

The wireless network is modeled by a directed graph $D = (V, A)$. The node set $V = \{1, \ldots, n\}$ represents the n stations of the network and the arc set A represents the directed paths connecting pairs of stations in V. The base station is regarded as node 1. Weights t_{ij} and d_{ij} are associated to each arc (path) $(i, j) \in A$ representing, respectively, the time it takes to travel from node (station) i to node (station) j and the distance among these nodes (stations). Let $T = \{1, 2, \ldots, \bar{T}\}$ be the time horizon considered divided in m time periods. At the beginning of the time horizon, each node $j \in V \setminus \{1\}$ contains an amount C_j of data. For each node $i \in V \setminus \{1\}$ data is generated at a rate of r_j units per time period in T. Thus, the amount of information at node j at each time period $k \in T$, denoted by q_{jk}, is proportional to the elapsed time from the last extraction (either physically or by radio), i.e.,

$$q_{jk} = \begin{cases} C_j + kr_j, & \text{if node } j \text{ has not been visited before time period } k, \\ (k - t_{last})r_j, & \text{otherwise, where } t_{last} \text{ is the time of the last extraction.} \end{cases}$$

Only the base node is properly equipped to send information outside the network. A unique vehicle is in charge of collecting data from all the stations in $V \setminus \{1\}$ and of transporting it to the base node. There is no capacity limit associated to the vehicle. At the beginning of the time horizon, the vehicle is located at the base node and at the end of the time horizon, it must return to the base node. Multiple visits are allowed to each node in V. Data can only be transferred to the vehicle once it is located in one of the stations in V, i.e., no data transfer is allowed while the vehicle is moving on an arc $(i, j) \in A$.

Wireless transmission is used to transfer data from a node $j \in V$ to the vehicle located in a node $i \in V$. Wireless transmission is only possible for close enough nodes. Let r_{cov} be a maximum distance allowing wireless transmission. A node j can wireless transfer its data to (the vehicle in) node i whenever $d_{ij} \leq r_{cov}$. We define the set of nodes that can send information to node i as $range(i) = \{j \in V : d_{ji} \leq r_{cov}\}$. We make the same physical and technical assumptions as in chapter 1. Thus, we assume transmission speed inversely proportional to the square of the distance between nodes depending on two additional factors: the amount of information transmitted and physical factors (as equipments used or obstacles between nodes). Let α_{ij} be a parameter representing the physical limitations of sending information among nodes i and j. The amount of information that can be sent per time unit from node j to node i is $\dfrac{1}{\alpha_{ji}(1 + d_{ji}^2)}$. As we have mentioned in the introduction, different from chapter 1, we assume nodes are free to transfer only part of their information to the vehicle.

Simultaneous transmissions are possible. Parameter M denotes the maximum number of nodes that can transfer information simultaneously to the vehicle in each time period and R denotes the maximum amount of information that can be transferred in each time period. In this case, the simultaneous data transfer finishes only when each individual wireless transmission finishes. As a consequence, the time of a simultaneous transmission corresponds to the highest time among individual wireless transmissions.

The version of the VRP treated in this chapter, called Wireless Transmission VRP (WT-VRP), consists of finding a feasible routing for the vehicle (i.e., a routing leaving at the beginning and returning at the end to the base node) and an efficient planning for collecting data from nodes $V \setminus \{1\}$. The criteria for measuring the efficiency of a collect planning is the total amount collected.

2.3 Mathematical models

Next, we introduce three MILP formulations to the WT-VRP. First, a time discrete model (Section 2.3.1) is developed where each decision is a multiple of the time unity. Second, an event model is proposed (Section 2.3.2) where visits to stations and transfer operations are considered as events. Finally, another event model is presented (Section 2.3.3) in which the considered events are the vehicle stops.

2.3.1 Discrete time model

Discrete time models have been used for related problems as maritime inventory routing problems (see (Agra et al., 2013a,b)). In this model the time horizon is discretized in a number of time periods $T = \{1, 2, \ldots, \bar{T}\}$. We assume that, at each time period in T, the vehicle is either traveling or waiting at a node and this behavior is modeled by the following two sets of binary variables.

For each $(i, j) \in A$ and $k \in T$, let

$$x_{ijk} = \begin{cases} 1 & \text{if the vehicle crosses the arc } (i,j) \text{ (going directly from node } i \text{ to node } j\text{) and arrives} \\ & \text{at the end of time } k, \\ 0 & \text{otherwise.} \end{cases}$$

For each $j \in V$ and $k \in T$, let

$$z_{jk} = \begin{cases} 1 & \text{if the vehicle is waiting at node } j \text{ during time period } k, \\ 0 & \text{otherwise.} \end{cases}$$

A third set of binary variables controls the wireless transmissions occurring at each time unit. For each pair of vertices $j \in V$, $i \in range(j)$ and $k \in T$, let

$$\theta_{jik} = \begin{cases} 1 & \text{if node } j \text{ sends information to node } i \text{ during time period } k, \\ 0 & \text{otherwise.} \end{cases}$$

Finally, continuous variables are used to describe the amounts of information at the nodes and the amounts of information transmitted. For each $j \in V$ and $k \in T$, let q_{jk} be the amount of information in node j at the end of time period k. For each $j \in V$, $i \in range(j)$ and $k \in T$, let f_{jik} be the amount of information transmitted from node j to node i during time period k.

The Discrete Time (DT) model follows.

$$\textit{Minimize} \qquad \sum_{j \in V} q_{jm} \tag{2.1}$$

$$\text{s.t.} \sum_{(1,j)\in A} x_{1jt_{1j}} = 1, \tag{2.2}$$

$$\sum_{(j,1)\in A} x_{j1\bar{T}} = 1, \tag{2.3}$$

$$\sum_{j\in V} z_{jk} + \sum_{(i,j)\in A} x_{ijk} \leq 1, \qquad \forall j \in V,\ \forall k \in T, \tag{2.4}$$

$$z_{jk} + \sum_{(i,j)\in A} x_{ijk} = \sum_{(j,p)\in A} x_{jp(k+t_{jp})} + z_{j(k+1)}, \qquad \forall j \in V, \forall k \in T, \tag{2.5}$$

$$\sum_{j\in range(i)} \theta_{jik} \leq M z_{ik}, \qquad \forall i \in V, k \in T, \tag{2.6}$$

$$f_{jik} \leq \frac{\theta_{jik}}{\alpha_{ji}(1+d_{ji}^2)}, \qquad \forall j \in V\ , \forall i \in range(j)\ ,\ \forall k \in T, \tag{2.7}$$

$$\sum_{j\in range(i)} f_{jik} \leq R, \qquad \forall i \in V, k \in T, \tag{2.8}$$

$$q_{jk} = q_{j,k-1} + r_j - \sum_{i\in range(j)} f_{jik}, \qquad \forall j \in V,\ k \in T | k > 1, \tag{2.9}$$

$$q_{j0} = C_j, \qquad \forall j \in V, \tag{2.10}$$

$$q_{jk} \geq 0, \qquad \forall j \in V,\ k \in T, \tag{2.11}$$

$$f_{jik} \geq 0, \qquad \forall j \in V,\ i \in range(j),\ k \in T, \tag{2.12}$$

$$\theta_{jik} \in \{0,1\} \qquad \forall j \in V,\ i \in range(j),\ k \in T, \tag{2.13}$$

$$z_{jk} \in \{0,1\}, \qquad \forall j \in V,\ k \in T, \tag{2.14}$$

$$x_{ijk} \in \{0,1\}, \qquad \forall (i,j) \in A, k \in T. \tag{2.15}$$

The objective function (2.1) minimizes the total amount of information remaining at the nodes at the end of the time horizon T, i.e., at time period $\bar{T}$. Constrains (2.2)–(2.5) are the *Routing Constraints*. Equations (2.2) and (2.3) ensure that the vehicle starts and ends its route at the base node. Inequality (2.4) ensures that at most one of the following cases can occur at time period k: the vehicle arrives at a node or the vehicle is waiting at a node to receive information. Equations (2.5) ensure that if either the vehicle arrives at node j or it is waiting at this node at time period k, then, at the next time period $k+1$, either it travels to a neighbor node or it keeps waiting at node j (see Figure 2.1). Inequalities (2.6) and (2.8) are variable upper bound constraints imposing the *Transfer Constraints*. Constraints (2.6) guarantees that at most M nodes send information to the vehicle simultaneously. Also, this inequality ensures that, whenever variable θ_{jik} is positive, for some $j \in V$, $i \in range(j)$ and $k \in K$ (i.e. at time period k, a node j sends information to the vehicle at node i) then z_{ik} must be one (i.e. the vehicle must be located at i at time period k). Similarly, inequalities (2.7) ensure that the maximum amount of information sent from node j to node i, at a time period k, is obeyed. Additionally, this set of inequalities ensure that if f_{jik} is positive, then the binary variable θ_{jik} must be one. Equations (2.8) ensures that during each time period the maximum amount of information that can be transferred to a node i cannot exceed R. Constraints (2.9) and (2.10) are the *Amount of information Constraints*. Equations (2.10) set the initial amount of information at each node. Equations (2.9) are the equilibrium constraints for the amount of information at each node. They impose that the amount of information at a node in time period k is equal to the amount information in the time $k-1$ plus the rate of that node minus the amount information extracted in the previous time period. Finally, inequalities (2.11)–(2.15) establish the domain of the variables.

Example 2.3.1 *Consider an example with six nodes, where node* 1 *is the base station, and with* $\bar{T} = 30$ *time periods. The data generation rates for nodes* 2 *to* 6 *are given respectively by* $3, 4, 2, 2$ *and* 4*. The parameters R and M are set to 20 and 3, respectively. We define* $\alpha_{ji} = 1/20$ *for* $j = i$ *and* $\alpha_{ji} = 1/6$*, otherwise. The initial amount of information at each node is zero. The following matrices are considered.*

$$d = \begin{bmatrix} 0 & 4 & 5 & 4 & 7 & 7 \\ 4 & 0 & 2 & 4 & 3 & 6 \\ 5 & 2 & 0 & 2 & 2 & 1 \\ 4 & 4 & 2 & 0 & 5 & 1 \\ 7 & 3 & 2 & 5 & 0 & 2 \\ 7 & 6 & 1 & 1 & 2 & 0 \end{bmatrix} \quad t = \begin{bmatrix} 0 & 4 & \infty & 4 & \infty & \infty \\ 4 & 0 & 2 & \infty & 3 & \infty \\ \infty & 2 & 0 & 2 & \infty & 1 \\ 4 & \infty & 2 & 0 & \infty & 1 \\ \infty & 3 & \infty & \infty & 0 & 2 \\ \infty & \infty & 1 & 1 & 2 & 0 \end{bmatrix}$$

$$W = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

where d is the distance matrix, t is the travel times matrix, and W is the matrix indicating whether $j \in range(i)$ $(W_{ji} = 1)$ *or* $j \notin range(i)$ $(W_{ji} = 0)$*. For example, node 2 can receive information from nodes 2, 3 and 5, while node 3 can receive information from nodes 2, 3,4, 5, 6.*

The optimal solution obtained with the DT model is depicted in Figure 2.1. The vehicle leaves the base station (node 1) at the beginning of the time horizon and arrives at node 2 at the end of period 4. It stays in node 2 during time periods 5 and 6, receiving information from nodes 2, 3 and 5. Next, the vehicle moves to node 3 where it spends one time period to receive information from nodes 2, 3 and 6. Then it moves to node 6 to receive information from nodes 3, 4 and 6. At the end of time period 12 the vehicle moves to node 5 where it stays for four periods. It receives information from nodes 3 and 5 during four time periods and from nodes 2 and 6 during two time periods. Then the vehicle moves again to node 6 where it stays for two periods, receiving information from nodes 3, 5 and 6. At the end of time period 22 it moves to node 4 to receive information from nodes 3, 4 and 6 during three time periods. Finally, the vehicle returns to the base station.

Strengthening the model Here we discuss several enhancements that allow to tighten the model, that is, to derive a new model whose linear relaxation value is closer to the optimum value. Exact methods based on the linear relaxation such as branch-and-bound and branch-and-cut will haver smaller enumeration trees when the model is tight (Wolsey, 1998).

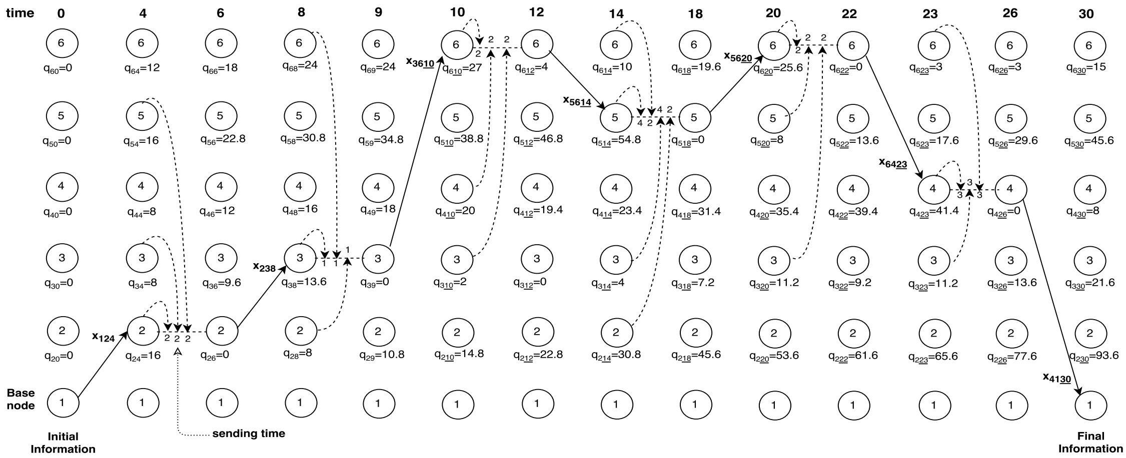


Figure 2.1 *– Optimal solution of the instance in Example 2.3.1 obtained with the Discrete Time model for* $\bar{T} = 30$*. Objective function value* $= 183.8$*.*

The first enhancement is to disaggregate inequalities (2.6) as follows:

$$\theta_{jik} \leq z_{ik}, \qquad \forall i \in V,\ j \in range(i),\ k \in T. \tag{2.16}$$

Although inequalities (2.16) are more in number, they are tighter than (2.6).

Another improvement is to replace inequalities (2.8) by the following variable upper bound constraints.

$$\sum_{j \in \text{range}(i)} f_{jik} \leq R\, z_{ik}, \qquad \forall i \in V,\ k \in T. \tag{2.17}$$

Next we define a set of valid inequalities that impose a limit on the amount transferred, for a subset of time periods. Let

$$t^* = \min_{(i,j) \in A} \{t_{ij}\}$$

denote the minimum traveling time between nodes. The following proposition establishes an inequality based on the fact that, during the subset of time periods $\ell \leq t^* + 1$, only one node can be visited.

Proposition 2.3.1 *For* $l \leq t^* + 1$*, the following inequality is satisfied by each feasible solution of the DT model.*

$$\sum_{s=1}^{l+1} \sum_{j \in V} \sum_{i \in range(j)} f_{jis} \leq \sum_{j \in V} C_j + \max_{\chi \in V}\{r_\chi\} l \tag{2.18}$$

Proof The proof is straightforward.

The DT model has $\mathcal{O}(\kappa \bar{T})$ variables and constraints, where $\kappa = \mid \sum_{i \in V} range(i) \mid$. When r_{cov} is larger than the greatest distance between two nodes, the size becomes

$\mathcal{O}(\mid V \mid^2 \bar{T})$, while in the opposite case, when r_{cov} is smaller than the minimum distance, it becomes $\mathcal{O}(\mid V \mid \bar{T})$. Hence, the main components with impact on the number of variables and constraints are the number of nodes (stations), the number of time periods considered, and the number of possible pairs of nodes for wireless transfer.

The DT model provides detailed information on the visits, normally leading to tight models (Agra et al., 2017). However, as the model depends on the time discretization, and since a fine discretization may be required to model the traveling times and the transfer operations, it tends to increase with the increase of the time horizon.

2.3.2 Node event model

The DT model includes many variables that are null in each possible solution since some nodes are not visited, and those that are visited receive few visits during the time horizon $T = [0, \bar{T}]$. In order to avoid the use of so many null variables, we introduce a flow model where only *events* are modeled, see (Agra et al., 2017). Two types of events are considered. A first set of events is denoted by Δ^r and includes all the possible physical node visits, which are defined by a pair (i, n) that represents the n^{th} visit of the vehicle to the node i. The second set of events, denoted by Δ^w, include events (j, k) representing the k^{th} time information is sent from node j. A feasible vehicle route is defined by a combination of events from Δ^r and Δ^w.

Next, we define the set of binary variables to be used in this section. For each pair $(i, n), (j, m) \in \Delta^r$, with $(i, j) \in A$,

$$x_{injm} = \begin{cases} 1 & \text{if the vehicle goes directly from node visit } (i,n) \text{ to node visit } (j,m), \\ 0 & \text{otherwise.} \end{cases}$$

For each $(j, m) \in \Delta^r$,

$$w_{jm} = \begin{cases} 1 & \text{if the node visit } (j,m) \text{ belongs to the vehicle route,} \\ 0 & \text{otherwise.} \end{cases}$$

For each $(i, k) \in \Delta^w$,

$$z_{ik} = \begin{cases} 1 & \text{if at least } k \text{ transfers occur from node } i, \\ 0 & \text{otherwise.} \end{cases}$$

For each $(j, k) \in \Delta^w$ and $(i, n) \in \Delta^r$,

$$\theta_{jkin} = \begin{cases} 1 & \text{if the } k^{th} \text{ transfer from node } j \text{ to the vehicle occurs at its } n^{th} \text{ stop at node } i, \\ 0 & \text{otherwise.} \end{cases}$$

The following set of continuous and integer variables will be also necessary. For

$(j,k) \in \Delta^w$ and $(i,n) \in \Delta^r$,

f_{jkin} : amount of information sent during the k^{th} transfer from node j to the vehicle, occurred at its n^{th} stop at node i.

ξ_{jkin} : duration, in number of time periods, of the k^{th} transfer from node j to the vehicle, occurred at its n^{th} stop at node i.

For $(i,n) \in \Delta^r$,

γ_{in} : duration, in number of time periods, of the n^{th} visit to node i.

t^r_{in} : time period at which the node visit (i,n) starts.

For $(j,k) \in \Delta^w$,

q_{jk} : amount of information in node j at the beginning of the k^{th} information transfer.

t^w_{ik} : time period at which the the k^{th} transfer from node i starts.

The Node Event (NE) model is described in the follows.

$$minimize \left\{ \sum_{i \in V} C_i + \bar{T} \sum_{i \in V} r_i - \sum_{(j,m) \in \Delta^w} \sum_{(i,n) \in \Delta^r} f_{jmin} \right\} \tag{2.19}$$

$$\sum_{j \in V \setminus \{1\}} x_{11j1} = 1, \tag{2.20}$$

$$\sum_{(j,m) \in \Delta^r} x_{jm12} = 1, \tag{2.21}$$

$$\sum_{(j,n) \in \Delta^r | (i,j) \in A} x_{imjn} = w_{im}, \qquad \forall (i,m) \in \Delta^r, \tag{2.22}$$

$$\sum_{(j,n) \in \Delta^r | (j,i) \in A} x_{jnim} = w_{im}, \qquad \forall (i,m) \in \Delta^r, \tag{2.23}$$

$$w_{im} \leq w_{i,m-1}, \qquad \forall (i,m) \in \Delta^r, m > 1, \tag{2.24}$$

$$\sum_{(j,k) \in V^w | j \in range(i)} f_{jkin} \leq R\xi_{in}, \qquad \forall (i,n) \in \Delta^r, \tag{2.25}$$

$$f_{jkin} \leq q_{jk} + r_j(\xi_{jkin} - 1), \qquad \forall (j,k) \in \Delta^w, (i,n) \in \Delta^r, j \in range(i), \tag{2.26}$$

$$f_{jkin} \leq \frac{\xi_{jkin}}{\alpha_{ji}(1 + d^2_{ji})}, \qquad \forall (i,n) \in \Delta^r, (j,k) \in \Delta^w, j \in range(i), \tag{2.27}$$

$$\sum_{(j,k)\in\Delta^w | j\in range(i)} \xi_{jkin} \leq M\, \gamma_{in}, \qquad \forall (i,n) \in \Delta^r, \tag{2.28}$$

$$\xi_{jkin} \leq \bar{T}\theta_{jkin}, \qquad \forall (j,k) \in \Delta^w, (i,n) \in \Delta^r, j \in range(i), \tag{2.29}$$

$$\sum_{k|(j,k)\in\Delta^w} \theta_{jkin} \leq w_{in}, \qquad \forall (i,n) \in \Delta^r,\ j \in range(i), \tag{2.30}$$

$$z_{jk} \leq z_{i,k-1}, \qquad \forall (j,k) \in \Delta^w, k > 1, \tag{2.31}$$

$$z_{jk} = \sum_{(i,m)\in\Delta^r} \theta_{jkim}, \qquad \forall (j,k) \in \Delta^w, \tag{2.32}$$

$$q_{jk} = C_j + r_j t^w_{jk} - \sum_{(i,n)\in\Delta^r | i\in range(j)} \sum_{\ell=1}^{k-1} f_{j\ell in}, \qquad \forall (j,k) \in \Delta^w, \tag{2.33}$$

$$t^r_{jm} \geq t^r_{in} + \gamma_{in} + t_{ij} - (\bar{T} + t_{ij})(1 - x_{injm}), \qquad \forall (i,n),\ (j,m) \in \Delta^r, (j,i) \in A \tag{2.34}$$

$$t^r_{i1} \geq \sum_{i|(i,1)\in\Delta^r} t_{i1} x_{11i1}, \qquad \forall (i,1) \in \Delta^r, \tag{2.35}$$

$$t^r_{jm} + \gamma_{jm} + t_{j1} x_{jm12} \leq \bar{T}, \qquad \forall (j,m) \in \Delta^r, \tag{2.36}$$

$$t^w_{jk} \geq t^r_{in} - \bar{T}(1 - \theta_{jkin}), \qquad \forall (i,n) \in \Delta^r,\ (j,k) \in \Delta^w, \tag{2.37}$$

$$t^w_{jk} + \xi_{jkin} \leq t^r_{in} + \gamma_{in} + \bar{T}(1 - \theta_{jkin}), \qquad \forall (i,n) \in \Delta^r, (j,k) \in \Delta^w, \tag{2.38}$$

$$x_{jmin} \in \{0,1\}, \qquad \forall (j,m), (i,n) \in \Delta^r, (j,i) \in A, \tag{2.39}$$

$$\theta_{jkin} \in \{0,1\}, \qquad \forall (j,k), \in \Delta^w, (i,n) \in \Delta^r, \tag{2.40}$$

$$q_{jk} \in \mathbb{R}^+, \qquad \forall (j,k) \in \Delta^w, \tag{2.41}$$

$$f_{jkin} \in \mathbb{R}^+, \qquad \forall (j,k), \in \Delta^w, (i,n) \in \Delta^r, \tag{2.42}$$

$$\gamma_{jk} \in \mathbb{Z}^+, \qquad \forall (j,k) \in \Delta^w, \tag{2.43}$$

$$\xi_{jkin} \in \mathbb{Z}^+, \qquad \forall (j,k), \in \Delta^w, (i,n) \in \Delta^r, \tag{2.44}$$

$$t^r_{jk} \in \mathbb{Z}^+, \qquad \forall (i,n) \in \Delta^r, \tag{2.45}$$

$$t^w_{in} \in \mathbb{Z}^+, \qquad \forall (j,k) \in \Delta^w. \tag{2.46}$$

The objective function (2.19) minimizes the amount of information kept in the nodes at the end of the time horizon. This amount is computed by removing the extracted information from the total information generated through the entire time horizon. Constraints (2.20)–(2.24) are the *Routing Constraints*. Equations (2.20) and (2.21) ensure, respectively, the vehicle leaves and ends its route in the node base 1. Equations (2.22) and (2.23) ensure the flow conservation, stating that if the m^{th} visit to node i occurs, then there must exist an arc entering and leaving that node. Constraints (2.24) state that if the m^{th} visit to node i occurs, so the previous $m-1^{th}$ visit must have occurred. The set of *Information Transfer Constraints* is defined by constraints (2.25)–(2.33). Constraints (2.25) limit the transfer amount considering the maximum transfer quantity per period. Constraints (2.26) ensure that the amount that can be transferred cannot exceed the information available at the corresponding node. Constraints (2.27) limit the transfer amount taking into account the transfer rate. Constraints (2.28) ensure that during the k^{th} visit of node i, the total duration of all transfers to node i cannot exceed the maximum allowed number of transfers per period, M, times the duration of the visit. Constraints (2.29) link the variables indicating the duration of the transfer operations (variables ξ_{jkin}) to the binary variables θ_{jkin} indicating whether a transfer occurs. Constraints (2.30) ensure that an information transfer occurs only if a visit occurs. Constraints (2.31) state that if the m^{th} transfer occurs so the previous $m-1^{th}$ must have occurred. Constraints (2.32) relate the binary transfer variables. Constraints (2.33) define the amount of information at each node at the beginning of each information transfer. Constraints (2.34)–(2.38) are the *Time Constraints*. The start time of each k^{th} visit is defined by constraints (2.34) and (2.35). Constraints (2.34) takes into account the start time of the previous visit plus the traveling time between the two locations and the time spent on the last visit. Notice that this inequality is redundant whenever $x_{injm} = 0$. Constraints (2.35) restrict the start time of the first visit. Constraints (2.36) force all the visits to end early enough so the vehicle can return to the base station before the end of the time horizon. Constraints (2.37) and (2.38) relate the start and end times of an information transfer from node j to node i, with the start and end times of the visit to node i. Notice that these inequalities are redundant whenever $\theta_{jkin} = 0$. Finally, constraints (2.39) - (2.46) define the variables domain.

Example 2.3.2 *Figure 2.2 depicts the solution of the instance used in Example 2.3.1, with the solution representation of the NE model. In this figure, two different set of nodes represent the two different types of events: circles represent events in Δ^r while squares represent events in Δ^w. For instance, we can see that station 6 is visited twice while the other stations are visited once; during the event $(6,2) \in \Delta^r$ (second visit to station 6), occur the events $(3,5),(5,2),(6,4) \in \Delta^w$ corresponding to transfer operations.*

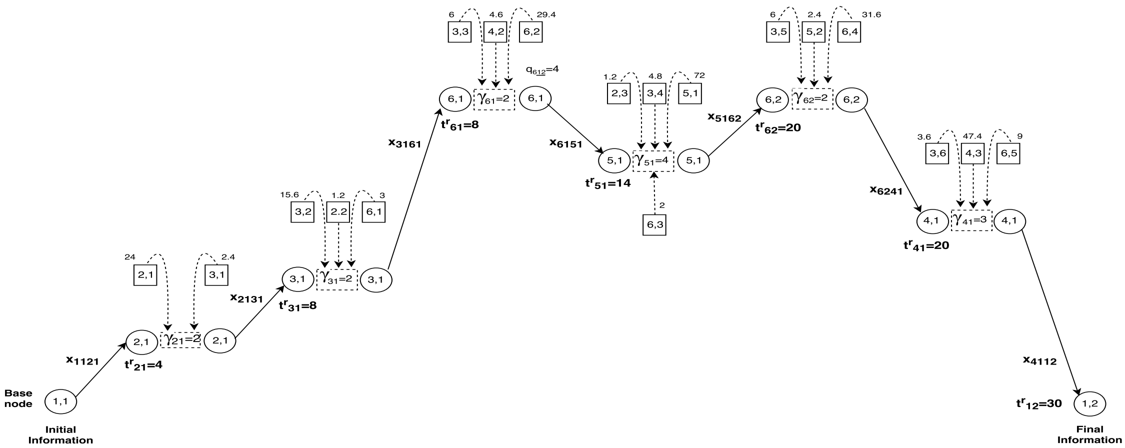


***Figure 2.2** – Optimal solution of the instance in Example 2.3.1 obtained with the Node Event model.*

Strengthening the model. The NE model can be strengthened with a valid inequality that bounds the transfer amount $\sum f_{jmin}$ with the maximum transfer rate R times the number of periods the vehicle can receive information, i.e., the number of periods the vehicle is not traveling.

Proposition 2.3.2 *The following inequality is satisfied by each feasible solution of the NE model.*

$$\sum_{(j,k)\in\Delta^w} \sum_{(i,n)\in\Delta^r : j \in range(i)} f_{jmin} \leq R(\bar{T} - \sum_{(i,n)\in\Delta^r} \sum_{(j,m)\in\Delta^r:(i,j)\in A} t_{ij} x_{injm}). \tag{2.47}$$

Proof The proof is straightforward.

Model NE has $\mathcal{O}(\sum_{i\in V} \sum_{j\in range(i)} r_i w_j + \sum_{(i,j)\in A} r_i r_j)$ variables and constraints, where the values r_i, w_j represent the maximum number of allowed visits to node i and the maximum number of allowed transfers from station j, respectively.

2.3.3 Vehicle event model

Typically, a vehicle route includes only a small number of nodes visited. Since the events in the NE model were defined on the set of nodes, most of the variables in this model will have null value. In this section we define a set of events associated with the vehicle: each event is a vehicle stop. This formulation resembles the layered formulation for the vehicle routing problems, see (Agra et al., 2012) and the references therein. Let $N = \{1, \ldots, \hat{N}\}$ denote the set of possible events where $\hat{N}$ is an upper bound on the number of events. The new routing variables indicate the node visited at the k^{th} vehicle stop, indexed by the event $k \in N$.

Next we define the new set of binary variables. For each $i \in V$, $k \in N$,

$$x_{ik} = \begin{cases} 1 & \text{if the } k^{th} \text{ vehicle event occurs at node } i, \\ 0 & \text{otherwise.} \end{cases}$$

The following continuous and integer variables are also defined. For each $k \in N$,

t_k : time period at which the k^{th} event occurs,

γ_k : time (in time periods) spent at the k^{th} event.

For each $j \in V$, $k \in N$,

q_{jk} : amount of information in node j at the beginning of event k.

For each $i, j \in V, k \in N$,

ξ_{jik} : duration (in time periods) of the information transfer from node j to node i at event k,
f_{jik} : amount of information transmitted from node j to node i during event k.

The Vehicle Event (VE) model is as follows.

$$Minimize \left\{ \sum_{i\in V}(Q_i + \bar{T}r_i) - \sum_{i\in V, j\in range(i), k\in N} f_{jik} \right\} \tag{2.48}$$

$$\sum_{j:(1,j)\in A} x_{j1} = 1, \tag{2.49}$$

$$\sum_{n\in N} x_{1n} = 1, \tag{2.50}$$

$$\sum_{j\in V} x_{jn} \le 1, \qquad \forall n \in N, \tag{2.51}$$

$$x_{jk} \le \sum_{i:(i,j)\in A} x_{i,k-1}, \qquad \forall j \in V, k \in N, \tag{2.52}$$

$$x_{jk} \le 1 - \sum_{\ell=1}^{k-1} x_{1\ell}, \qquad \forall j \in V \setminus \{1\}, k \in N, \tag{2.53}$$

$$t_k \ge t_{k-1} + \gamma_{k-1} + t_{ij}(x_{i,k-1} + x_{jk} - 1), \qquad \forall (i,j) \in A,\ k \in N, \tag{2.54}$$

$$t_1 \ge \sum_{j:(1,j)\in A} t_{1j}x_{1j}, \tag{2.55}$$

$$t_k \le \bar{T}, \qquad \forall k \in N, \tag{2.56}$$

$$q_{jk} = Q_j + r_j t_k - \sum_{\ell=1}^{k-1} \sum_{i\in range(j)} f_{ji\ell}, \qquad \forall j \in V,\ k \in N, \tag{2.57}$$

$$f_{jik} \le q_{jk} + r_j(\xi_{jik} - 1), \qquad \forall j \in V,\ i \in \text{range}(j),\ k \in N, \tag{2.58}$$

$$f_{jik} \le \frac{\xi_{jik}}{\alpha_{ji}(1 + d_{ji}^2)}, \qquad \forall j \in V,\ i \in \text{range}(j),\ k \in N, \tag{2.59}$$

$$\sum_{j\in range(i)} f_{jik} \le R\gamma_k, \qquad \forall i \in V,\ k \in N, \tag{2.60}$$

$$\sum_{i\in range(j)} \xi_{jik} \le \gamma_k, \qquad \forall j \in V,\ k \in N, \tag{2.61}$$

$$\sum_{j\in V, i\in range(j)} \xi_{jik} \le M\gamma_k, \qquad \forall k \in N, \tag{2.62}$$

$$\xi_{jik} \le \bar{T}x_{ik}, \qquad \forall i \in V,\ j \in range(i),\ k \in N, \tag{2.63}$$

$$f_{jik} \geq 0, \qquad \forall i \in V, j \in range(i), k \in N, \tag{2.64}$$
$$q_{jk} \geq 0, \qquad \forall j \in V, k \in N, \tag{2.65}$$
$$t_k \in \mathbb{Z}^+ \qquad \forall k \in N, \tag{2.66}$$
$$\gamma_k \in \mathbb{Z}^+, \qquad \forall k \in N, \tag{2.67}$$
$$\xi_{jik} \in \mathbb{Z}^+, \qquad \forall i \in V, j \in range(i), k \in N, \tag{2.68}$$
$$x_{ik} \in \{0,1\}, \qquad \forall i \in V, k \in N. \tag{2.69}$$

The objective function (2.48) minimizes the amount of information kept in the nodes at the end of the time horizon. Constraints (2.49)–(2.53) are the *Routing Constraints*. Equations (2.49) and (2.50) ensure, respectively, the vehicle leaves and ends its route at the base node. Inequalities (2.51) state that at most one visit labeled n is made. Inequalities (2.52) ensure that, if the k^{th} visit is made to node j, then the $k-1^{th}$ visit occurred in one of the predecessors of node j. Constraints (2.53) ensure that all the routing variables are null after the vehicle has returned to the base node. Constrains (2.54)–(2.56) are the *Time Constraints*. Inequalities (2.54) impose that the start time of the k^{th} visit takes into account the start time of the previous visit, the time spent on the last visit and the traveling time between the two locations visited. Constraints (2.55) restrict the start time of the first visit while constraints (2.56) force all the visits to start during the time horizon (this includes the last visit which is the return to the base station). The *Information Transfer Constraints* are constraints (2.57)–(2.63). Constraints (2.57) define the amount of information at each node at the beginning of each visit. Constraints (2.58) ensure that the amount that can be transferred cannot exceed the information available at the corresponding node. Constraints (2.59) limit the transfer amount taking into account the transfer rate, while constraints (2.60) limit the transfer amount considering the maximum transfer quantity per period. Constraints (2.61) ensure that, during a visit to node i, the time used to transfer information from each node j to node i, cannot exceed the time the vehicle has spent at node i. Constraints (2.62) ensure that during each visit, the total transfer time to node i cannot exceed the maximum number of transfers per period, M, times the duration of the visit. Constraints (2.63) link the transfer variables to the routing variables, ensuring that a node j can transfer information to a node i during the k^{th} visit if the k^{th} visit occurred at node i. Finally, Constraints (2.64)–(2.69) define the variables domain.

Example 2.3.3 *Figure 2.3 depicts the solution of the instance used in Example 2.3.1, with the solution representation of the VE model. A node (i,k) in this network representation of a solution is associated with the event $k \in N$ occurring at node $i \in V$. A dotted line from a node (j,k) to a node (i,k) represents an information transfer occurring from j to i during the k^{th} visit. For instance, we can see that the 5^{th} visit of the vehicle occurs at node 6 to receive information from nodes 3, 5 and 6.*

Model VE has $\mathcal{O}(\kappa\hat{N})$ variables and constraints, where $\kappa = \left| \sum_{i \in V} range(i) \right|$. As for the NE model, the size of the VE model depends on the size of the event set, i.e., from $\hat{N}$. Let $\varphi(\hat{N})$ denote the value of the objective function of the vehicle event model considering a maximum of $\hat{N}$events occur. The following statements hold true.

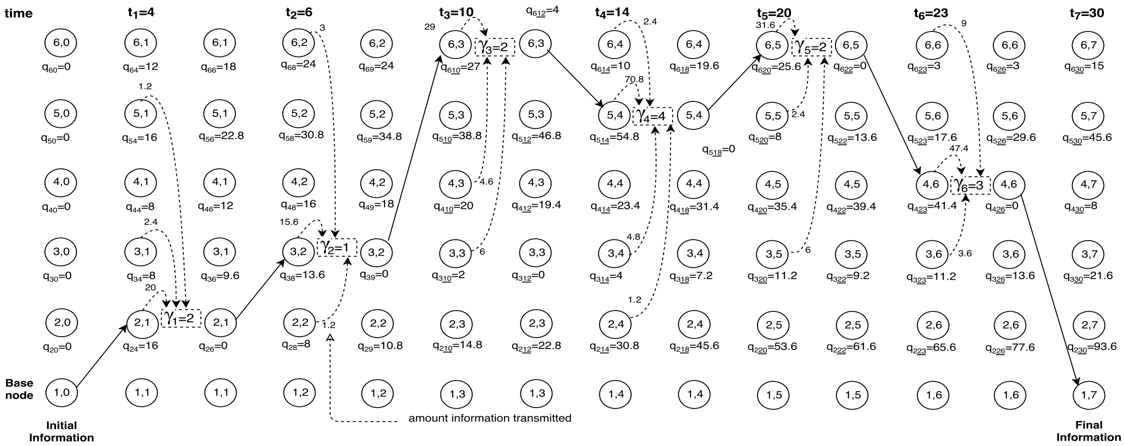


Figure 2.3 *– Optimal solution of the instance in Example 2.3.1 obtained with the Vehicle Event model.*

(i) φ is non increasing in the maximum number of events, that is, $N_1 > N_2$ then $\varphi(N_1) \leq \varphi(N_2)$.

(ii) $\exists N \in \mathbb{N}$ such that $\varphi(N_1) = \varphi(N), \forall N_1 > N$.

(iii) Let N^* denote the lowest N satisfying statement (ii), that is, the lowest possible value for the number of visits that gives the optimal solution. If $N' < N^*$, then $\varphi(N')$ gives an upper bound on the optimal value $\varphi(N^*)$.

Clearly, N^* is not known. By underestimating it the model becomes easier to solve but the solution cost will increase. Overestimating N^* may lead to longer running times.

2.4 Computational experiments

In this section, we report the computational tests conducted to evaluate the three models presented in the previous section. All the results were performed using a server with 15 CPU's Intel ®Xeon (R) E5540@ 2.53Ghz X4, with 16 GB of RAM. To solve the several MILP models, the IBM CPLEX Optimizer 12.6.1.0 solver was used with a time limit equal to 3600 seconds.

A set of instances was randomly generated as described in chapter 1. The vertices in V are located on a square grid of length $\ell = 8$. The base station is located on the bottom left vertex and the remaining stations are placed randomly on a square of length $\ell' = 6$ in the upper right of the grid. The distance matrix is given by the euclidean distance between the stations. The graph edges are selected randomly. In order to obtain a certain graph density d, starting from a complete graph, edges are removed randomly, while ensuring connectivity, until the desired graph density is obtained. In this work, we generate instances varying $|V|$ in $\{8, 10, 12, 20\}$ and with $d = 0.4$. We considered the values of $\bar{T} \in \{72, 120, 240\}$ and the data generation rates r_j are randomly generated in

$\|V\|$	$\bar{T}$		DT Model	NE Model				VE Model		
				$r=1,w=2$	$r=1,w=3$	$r=2,w=2$	$r=2,w=3$	$\hat{N}=7$	$\hat{N}=8$	$\hat{N}=9$
10	50	variables	7101	374	524	765	1038	743	849	955
		constraints	4142	597	838	1236	1682	1678	1927	2176
	100	variables	14201	374	524	765	1038	743	849	955
		constraints	8292	597	838	1236	1682	1678	1927	2176
20	50	variables	33701	1769	2543	3709	5200	3655	4177	4699
		constraints	16032	3002	4273	6230	8696	8168	9365	10562
	100	variables	67401	1769	2543	3709	5200	3655	4177	4699
		constraints	32082	3002	4273	6230	8696	8168	9365	10562

***Table 2.1** – Number of variables and constraints used in each model for different parameters.*

the interval $[1,5]$. The values of α_{ij} were randomly generated in $\{1/12, 1/13, 1/14\}$ if $i = j$ and in $\{1/5, 1/6, 1/7\}$ otherwise. The following values parameters were set: $r_{cov} = 4$, $R = 20$ and $M = 3$.

First, in Table 2.1, we compare the size of the three models for the combination of the parameters defining the instances and the models used in our experiments. For model NE, the values r, w in the top of the four columns represent, respectively, the maximum number of allowed visits to and the maximum number of allowed transfers from each station. We can observe that DT model is the largest model, while NE model is the smallest one. The VE model is an intermediate model in terms of size. Next, we will describe the results obtained with each one of the three models.

In Table 2.2, we report the computational results obtained with the DT model. The table is split into two parts accordingly to the time horizons. For each part, the first three columns give the number of stations ($|V|$), the density of the graph, d (since the data generation process may generate graphs with density slightly different from 0.4), and the size of the time horizon $\bar{T}$. The fourth column gives the total amount of information generated during the time horizon. The fifth column gives the value of the best feasible solution found, that is, the amount of information remaining in the nodes at the end of time horizon T. The following three columns give: (Cpu) is the running time (in seconds); (Gap) is the final integrality gap reported by the solver at the end of running time; and (Nodes) is the number of nodes of the branch-and-bound algorithm. An instance with $***$ in the (Cpu) and with (Gap) superior to zero is not solved to optimality in the time limit.

The results show that for $\bar{T} = 72$ all except one instance with $|V| \in \{10, 15, 20\}$ are solved to optimality within the one hour time limit. When $\bar{T}$ is increased, the number of solved instances decreases. For $\bar{T} = 240$, even when $|V| = 8$, no instance is solved to optimality with final gap arriving to 28%.

Next, in Table 2.3 we report the results obtained with the NE model. The four first columns are similar to the ones of Table 2.2. The following four sets of columns give the same information as the corresponding ones in Table 2.2; namely the best feasible solution value (best sol), the running time (Cpu), the final gap (Gap) and the number of nodes (Nodes). Again, the values r, w in the top of the four multicolumns represent, respectively, the maximum number of allowed visits to and the maximum number of allowed transfers from each station.

***Table 2.2** – Results obtained with the Discrete Time model.*

$\|V\|$	d	$\bar{T}$	Total	best sol.	Cpu	Gap	Nodes	$\|V\|$	d	m	Total	best sol.	Cpu	Gap	Nodes
10	0.42	72	2392	1676.64	63.69	0	6809	10	0.42	120	3832	2453.21	443.63	0	24423
			2220	1538.64	81.23	0	8589				3564	2216.67	515.01	0	28669
			1744	1128.11	57.55	0	8685				2752	1550.53	748.32	0	206814
			2006	1337.79	96.86	0	11171				3206	1876.88	***	3.64	261685
			1797	1127.11	146.49	0	18012				2853	1581.25	***	0.28	254528
			2073	1413.24	85.64	0	9085				3321	2063.8	515.77	0	23966
			2286	1620.57	106.2	0	11851				3678	2407.19	505.97	0	29222
			1952	1250.58	107.18	0	32751				3104	1726.78	636.26	0	24311
			2339	1671.6	102.15	0	9232				3731	2423.72	483.33	0	25211
			1734	1083.79	99.88	0	11483				2742	1422.04	752.43	0	42202
15	0.4	72	3808	3059.5	271.19	0	8477	15	0.4	120	5477	4086.59	***	2.47	21649
			3205	2468.17	487.51	0	27490				5638	4280.33	***	0.7	25301
			2753	2051.69	277.88	0	12014				4915	3602.7	***	0.7	31131
			3112	2390.6	203.24	0	10785				5140	3758.86	***	3.85	22132
			2738	2059.6	248.73	0	11993				4899	3637.02	***	1.84	22524
			2876	2173.72	***	2.18	360579				5068	3633.39	***	0.45	29757
			3213	2473.69	139.26	0	6694				5616	4138.39	***	0.23	24494
			3669	2912.56	212.09	0	9233				5626	4258.6	***	0.6	28796
			3112	2324.19	259.58	0	6046				6224	4791.89	***	1.76	29075
			2797	2123.41	192.35	0	11900				6234	4811.39	3382.54	0	45284
20	0.4	72	5149	4301.5	1001.32	0	151612	8	0.42	240	5216	2385.33	***	6.39	97091
			4574	3838.6	1033.39	0	16072				4976	2037.34	***	10.34	87142
			4239	3528.7	507.27	0	11944				5700	2895.91	***	4.7	100844
			4924	4167.57	483.79	0	9344				3764	1242.85	***	28.32	74207
			4657	3904.39	963.63	0	10171				4960	2297.1	***	7.09	71848
			5097	4315.39	1094.51	0	21540				4965	2190.77	***	8.42	55845
			4928	4180.5	905.1	0	26529				4977	2127.55	***	15.57	46945
			4175	3448.59	587.1	0	12830				5910	2896.44	***	3.71	110735
			5650	4844.29	410.95	0	8962				5223	2572.06	***	7.5	87638
			5007	4239.89	451.94	0	10770				5448	2700.38	***	5.65	114550

**** time limit: 3600 sec.*

Table 2.3 – Results obtained with the Node Event model.

$\|V\|$	d	$\bar{T}$	Total	$r=1$ $w=2$				$r=1$ $w=3$				$r=2$ $w=2$				$r=2$ $w=3$			
				best sol.	Cpu	Gap	Nodes	best sol.	Cpu	Gap	Nodes	best sol.	Cpu	Gap	Nodes	best sol.	Cpu	Gap	Nodes
10	0.42	72	2392	1724.19	8.54	0	31450	1721.78	28.54	0	172659	1697.38	***	295.49	1149888	1758.86	***	290.6	929083
			2220	1966.41	0.85	0	646	1966.41	1.83	0	768	1601.85	***	272.52	1194195	1581.05	***	303.23	789375
			1744	1159.59	4.76	0	18794	1155.89	20.81	0	95069	1149.74	***	287.46	2025411	1166.71	***	322.3	1208221
			2006	1341.99	13.44	0	39759	1337.8	31.84	0	116451	1366.89	***	310.23	1298188	1357.49	***	332.79	717590
			1797	1168.92	10.01	0	41684	1160.55	39.14	0	283311	1168.92	***	310.43	2049193	1203.48	***	324.45	1101298
			2073	1424.42	10.49	0	30478	1413.7	32.45	0	153478	1426.85	***	277.62	1513668	1437.75	***	306.21	581914
			2286	1656.6	5.59	0	20531	1656.07	18.39	0	94219	1665.86	***	265.26	1800753	1661.51	***	293.46	794190
			1952	1262.28	10.41	0	30769	1250.58	22.4	0	87726	1284.89	***	271.44	1697986	1250.58	***	306.25	1035625
			2339	1680.4	11.02	0	34742	1671.6	34.81	0	139798	1680.4	***	274.55	1268322	1721.29	***	299.58	677665
			1734	1085.61	7.58	0	33221	1083.8	52.09	0	537229	1085.61	***	317.49	1350680	1097.39	***	331.25	1029658
15	0.4	72	3808	3116.19	**	156.49	1575724	3102	***	189.56	1229423	3116.32	***	267.5	723499	3083.19	***	297.82	316754
			3205	2471.59	**	126.6	1791434	2468.19	***	0.2	2588204	2530	***	282.42	753328	2523.38	***	299.28	362158
			2753	2051.69	**	179.04	1601614	2073.99	***	256.7	772934	2110.82	***	278.47	1377724	2200.1	***	296.77	364035
			3112	2394.47	**	142.19	1704099	2446.69	***	225.83	900202	2110.82	***	278.47	1377724	2457.59	***	298.95	580941
			2738	2061.09	**	106.7	2133380	2071.79	***	228.2	986031	2110.82	***	278.47	1377724	2089.71	***	308.15	615913
			2876	2190.5	1313.04	0	1254393	2211.21	***	166.47	1146607	2280.23	***	269.12	1081001	2282.35	***	298.27	520333
			3213	2473.69	***	91.44	1545789	2533.69	***	139.5	1114749	2509.69	***	288.28	1124570	2678.04	***	295.55	350887
			3669	3007.84	***	154.48	2402603	3006.35	1857.92	0	2977964	2981.45	***	276.06	1181722	3100.4	***	285.7	438971
			3112	2329.7	***	92.23	2044297	2324.4	***	131.22	1256214	2487.69	***	267.66	1100911	2468.39	***	294.15	399204
			2797	2123.41	1990.16	0	2049539	2127.94	**	108.63	1840108	2216.57	***	269.82	1509651	2239.27	***	297.04	509990
10	0.42	120	3832	2499.63	17.59	0	110862	2482.61	344.1	0	4142786	2534.36	***	284.97	1956508	2542.55	***	331.34	804639
			3564	3137.7	0.95	0	654	3137.7	1.47	0	480	2269.92	***	288.32	1328581	2286.49	***	343.7	812161
			2752	1564.61	31.7	0	498781	1555.61	57.25	0	593308	1612.34	***	338.78	1646827	1613.97	***	375.98	1267842
			3206	1884.73	19.03	0	96745	1876.88	95.86	0	694556	1939.38	***	346.34	1151266	1923.56	***	379.44	750328
			2853	1590.94	9.26	0	49145	1580.01	76.96	0	694533	1616.08	***	340.98	2382438	1617.72	***	375.25	1129317
			3321	2079.21	22.02	0	172610	2070.82	180.89	0	1977811	2079.21	***	310.36	1628539	2138.64	***	327.01	627780
			3678	2433.1	20.54	0	224393	2422.67	84.69	0	792202	2444.45	***	306.22	1148285	2479.52	***	316.8	853670
			3104	1763.53	17.09	0	89928	1738.93	78.33	0	589494	1763.53	***	310.9	2057964	1752.72	***	369.19	880102
			3731	2434.51	29.58	0	339080	2423.72	232.72	0	3082140	2434.51	***	289.3	1658307	2495.78	***	336.14	654701
			2742	1474	545.26	0	12937921	1448.57	***	0.8	22348921	1470.29	***	390.93	1095967	1480.79	***	392.28	842943
8	0.42	240	5216	2579.93	2.06	0	14880	2568.52	4.02	0	37643	2457.39	1677.24	0	10470801	2365.66	***	32.95	8250829
			4976	2298.28	2.6	0	25218	2231.32	27.81	0	420200	2171.69	1891.87	0	16504873	2107.25	***	46.09	6450978
			5700	3122.77	1.93	0	14116	3104.08	3.61	0	26239	2982.94	1070.25	0	3981748	2950.09	***	28.81	3987428
			3764	1444.09	2.35	0	15559	1426.74	13.28	0	150225	1355.23	1818.86	0	13223040	1273.45	***	71.07	3283058
			4960	2495.54	1.97	0	15098	2460.07	4.38	0	44873	2371.44	2113.4	0	10704389	2338.73	***	39.28	8278251
			4965	3784.35	0.27	0	0	3784.64	0.27	0	0	2285.32	***	270.53	2539042	2242.18	***	316.28	1944104
			4977	2398.77	2.05	0	12741	2357.5	7.46	0	77238	2198.97	557.91	0	3896604	2164.77	***	37.31	7336873
			5910	4704.82	0.26	0	0	4704.82	0.49	0	0	2956.79	2550.54	0	7059293	2907.28	***	27.42	6743235
			5223	2661.92	2.78	0	22127	2624.35	13.02	0	178347	2640.83	***	14.11	9554160	2619.68	***	229.23	2290740
			5448	2765.27	3	0	23710	2730.17	21.31	0	284750	2765.37	***	1.76	22104402	2711.08	***	36.26	6839767

**** time limit : 3600 sec.*

The results in Table 2.3 show that the model can only be solved to optimality for very small values of r and w. The average values of the best solutions obtained with the NE Model on the set of instances with $|V| = 8$, $\bar{T} = 240$ are compared in Figure 2.4 with the average value of the best solutions found by the VE Model.

Finally, in Table 2.4, we report the results obtained using the VE Model. The meaning of the columns is the same as for the two previous tables. The last four sets of columns are grouped accordingly to the maximum number of vehicle events $\hat{N}$. The bold numbers mean that the corresponding value N is probably N^*: the objective value did not change for $N > \hat{N}$.

As depicted in Figure 2.5, for the particular case with $|V| = 15$ and $\bar{T} = 120$, running times increase when $\hat{N}$ increases. However, as it can be seen from Figure 2.6 that depicts the average amount of information at the end of time T at all nodes (for the same instances of Figure 2.5), the solution quality also increases when N increases, until the value of N^* is obtained.

Table 2.4 – Results obtained with the Vehicle Event model.

$\|V\|$	d	$\bar{T}$	Total	$\hat{N} = 5$				$\hat{N} = 6$				$\hat{N} = 7$				$\hat{N} = 8$			
				best sol.	Cpu	Gap	Nodes	best sol.	Cpu	Gap	Nodes	best sol.	Cpu	Gap	Nodes	best sol.	Cpu	Gap	Nodes
10	0.42	72	2392	1713.68	0.29	0	0	**1676.64**	0.41	0	0	1676.64	0.9	0	882	1676.64	2.28	0	2776
			2220	1712.62	0.17	0	18	1549.64	0.22	0	0	**1538.64**	0.58	0	335	1538.64	0.9	0	1401
			1744	1155.89	0.31	0	0	**1128.11**	0.62	0	201	1128.11	0.76	0	0	1128.11	1.77	0	1711
			2006	1372.78	0.3	0	0	**1337.80**	0.44	0	0	1337.80	1.07	0	932	1337.80	1.99	0	2882
			1797	1165.1	0.28	0	0	1157.84	0.41	0	0	**1127.11**	1.04	0	1112	1127.11	2.16	0	3156
			2073	1452.27	0.28	0	0	**1413.24**	0.57	0	229	1413.24	1.03	0	854	1413.24	2.16	0	3019
			2286	1656.07	0.72	0	133	**1620.57**	0.57	0	476	1620.57	1.09	0	1633	1620.57	2.21	0	5228
			1952	**1250.58**	0.43	0	0	1250.58	0.96	0	577	1250.58	1.45	0	1420	1250.58	2.93	0	3380
			2339	1674.43	0.52	0	119	**1671.60**	0.91	0	624	1671.60	1.65	0	1241	1671.60	2.71	0	3736
			1734	1148.44	0.22	0	0	1104	0.45	0	0	**1083.80**	1.11	0	855	1083.8	2.32	0	2585
15	0.4	72	3808	**3059.50**	1.6	0	660	3059.50	2.92	0	3090	3059.50	18.6	0	18411	3059.50	60.74	0	60499
			3205	**2468.19**	1.37	0	472	2468.19	2.55	0	2675	2468.26	21.22	0	18452	2468.17	76.49	0	72026
			2753	2084.17	1.58	0	613	**2051.69**	2.69	0	2869	2051.69	23.97	0	20167	2051.69	55.97	0	55522
			3112	**2390.60**	1.04	0	254	2390.6	1.95	0	1752	2390.6	3.98	0	6506	2390.6	26.41	0	24096
			2738	2066.72	1.28	0	233	**2059.60**	2.43	0	1518	2059.6	6.08	0	7493	2059.69	55.14	0	42246
			2876	2192.54	1.14	0	672	2175.99	2.08	0	2944	**2171.19**	16.71	0	17109	2171.19	35.73	0	46246
			3213	**2473.69**	1.51	0	587	2473.69	2.48	0	3392	2473.69	13.63	0	7894	2473.69	56.47	0	39081
			3669	3004.91	0.9	0	463	**2912.56**	1.58	0	1366	2912.56	4.57	0	6191	2912.56	39.16	0	27267
			3112	**2324.19**	1.63	0	228	2324.19	2.19	0	1676	2324.19	6.05	0	9176	2324.19	36.71	0	29815
			2797	**2123.41**	1.23	0	354	2123.41	2.14	0	2541	2123.41	10.4	0	7976	2123.41	59.36	0	54844
20	0.4	72	5149	4316.39	2.16	0	1705	**4301.50**	5.54	0	9258	4301.50	43.89	0	47785	4301.50	210.99	0	206763
			4574	**3838.59**	2.42	0	1280	3838.59	20.03	0	16696	3838.59	69.96	0	65746	3838.6	586.09	0	565564
			4239	**3528.70**	2.51	0	2004	3528.70	17.8	0	16840	3528.70	61.47	0	57857	3528.70	310.7	0	311865
			4924	**4167.57**	2.07	0	782	4167.57	4.68	0	5245	4167.57	38.34	0	35146	4167.57	175.14	0	176522
			4657	3932.1	2.88	0	1475	**3904.39**	6.35	0	9768	3904.7	69.72	0	59977	3904.4	427.21	0	359717
			5097	4363.2	2.49	0	1803	4329	17.24	0	13732	**4315.39**	63.97	0	64688	4315.39	293.42	0	290607
			4928	4182.1	2.3	0	1886	**4180.5**	20.06	0	20270	4180.5	56.79	0	57610	4180.5	367.51	0	331521
			4175	**3448.59**	2.27	0	1147	3448.6	14.85	0	13423	3448.6	50.02	0	47950	3448.6	288.1	0	288411
			5650	4849.23	2.83	0	1826	**4844.30**	18.16	0	17795	4844.5	57.56	0	56213	4844.3	434.15	0	450813
			5007	4268.09	1.53	0	302	**4239.89**	3.14	0	2577	4239.89	23.74	0	18533	4239.89	112.68	0	90607
				$\hat{N} = 7$				$\hat{N} = 8$				$\hat{N} = 9$				$\hat{N} = 10$			
10	0.42	120	3832	**2453.24**	1.09	0	875	2453.21	1.74	0	3326	2453.21	4.88	0	9513	2453.24	22.88	0	30409
			3564	2285.05	0.36	0	0	**2216.67**	0.88	0	783	2216.67	1.68	0	2986	2216.67	10.06	0	18917
			2752	1553.01	0.99	0	518	1553.01	1.41	0	1957	**1550.53**	3.52	0	6987	1550.53	17.23	0	22289
			3206	**1876.88**	1.08	0	1117	1876.88	1.94	0	2423	1876.88	5.06	0	9128	1876.88	22.74	0	27025
			2853	**1580.01**	0.97	0	0	1580.01	1.96	0	3570	1580.01	10.45	0	15240	1580.01	33.26	0	47042
			3321	2070.82	1.15	0	912	**2063.8**	2.01	0	4126	2063.79	13.18	0	20640	2063.8	24.66	0	32398
			3678	**2407.19**	1.37	0	1840	2407.19	2.01	0	5523	2407.19	13.07	0	23340	2407.19	33.2	0	59857
			3104	**1726.78**	1.35	0	1061	1726.78	2.9	0	3004	1726.78	13.12	0	13193	1726.78	29.73	0	31667
			3731	**2423.72**	1.59	0	1285	2423.72	2.44	0	4980	2423.72	6.65	0	13129	2423.72	28.22	0	34909
			2742	1485.29	1.13	0	510	1462.19	1.55	0	2066	1423.4	4.3	0	7205	**1422.2**	20.13	0	24431
15	0.4	120	5477	4073.59	23.21	0	20588	4073.59	73.38	0	92031	4072.7	519.72	0	470363	**4072.61**	2840.06	0	2262717
			5638	4272.14	18.3	0	18636	4272.14	51.74	0	58017	**4271.98**	243.95	0	231793	4271.98	1728.86	0	1673800
			4915	3615.4	5.11	0	8722	**3595.2**	60.25	0	58169	3595.2	172.51	0	128756	3595.2	825.35	0	639380
			5140	**3732.38**	18.51	0	19351	3732.38	40.68	0	50219	3732.38	209.33	0	196023	3732.38	1091.8	0	895180
			4899	**3613.1**	12.04	0	15090	3612.99	34.67	0	44224	3613	157.47	0	168952	3613.2	767.76	0	744047
			5068	**3630.49**	5.73	0	9184	3630.49	58.4	0	48098	3630.5	161.73	0	165570	3630.6	1161.39	0	988048
			5616	**4134.70**	22.83	0	19123	4134.70	56.16	0	56547	4134.70	404.05	0	274153	4134.70	1658.43	0	850429
			5626	**4256.64**	16.9	0	17250	4256.64	48.83	0	67235	4256.64	281.5	0	296837	4256.64	1702.87	0	1579158
			6224	**4761.89**	16.81	0	17767	4761.9	60.87	0	76541	4761.89	282.65	0	314020	4761.89	1270.3	0	1253523
			6234	4811.6	11.97	0	15453	**4811.39**	42.46	0	63248	4811.39	179.98	0	239578	4811.39	832.28	0	978057
				$N = 9$				$N = 10$				$N = 11$				$N = 12$			
8	0.42	240	5216	2400.8	1.48	0	1893	**2351.82**	2.8	0	4295	2351.82	12.43	0	15321	2351.82	19.77	0	29151
			4976	2106.46	2	0	3661	2035.21	4.36	0	10006	2015.5	16.15	0	23014	**2003.14**	28.87	0	44288
			5700	2895.17	2.09	0	3473	2872.24	8.23	0	17204	**2865.99**	29.32	0	56577	2865.99	25.92	0	49879
			3764	1262.58	1.48	0	1967	1248.32	3.28	0	6283	1207.12	12.65	0	24800	**1187.08**	20	0	36982
			4960	2290.94	1.88	0	2531	2268.63	2.88	0	6461	**2268.47**	10.65	0	18714	2268.47	25.32	0	51113
			4965	2205.5	1.43	0	1253	2180.72	2.52	0	4830	**2164.96**	4.12	0	8662	2164.96	31.6	0	40639
			4977	2139.89	1.91	0	2953	2086.69	10.05	0	13276	**2042.67**	19.93	0	29649	2042.72	64.89	0	86681
			5910	2897.95	1.13	0	2093	2897.29	2.68	0	5835	**2890.55**	11.4	0	22165	2890.55	36.38	0	70890
			5223	2570.32	1.99	0	2523	2557.37	2.4	0	6838	**2544.78**	14.81	0	26109	2544.78	37.16	0	63157
			5448	2684.69	1.98	0	3319	**2659.17**	4.19	0	9936	2659.17	15.46	0	25578	2659.17	44.29	0	78339

**** time limit : 3600 sec.*

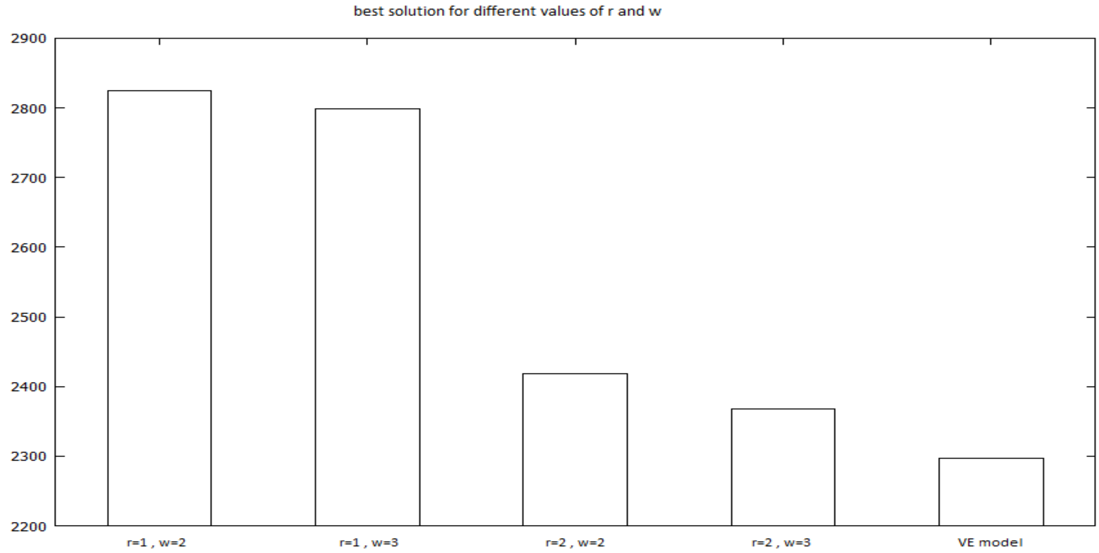


Figure 2.4 – *Average values of the best solutions obtained using NE model on the set of with* $|V| = 8$, $\bar{T} = 240$.

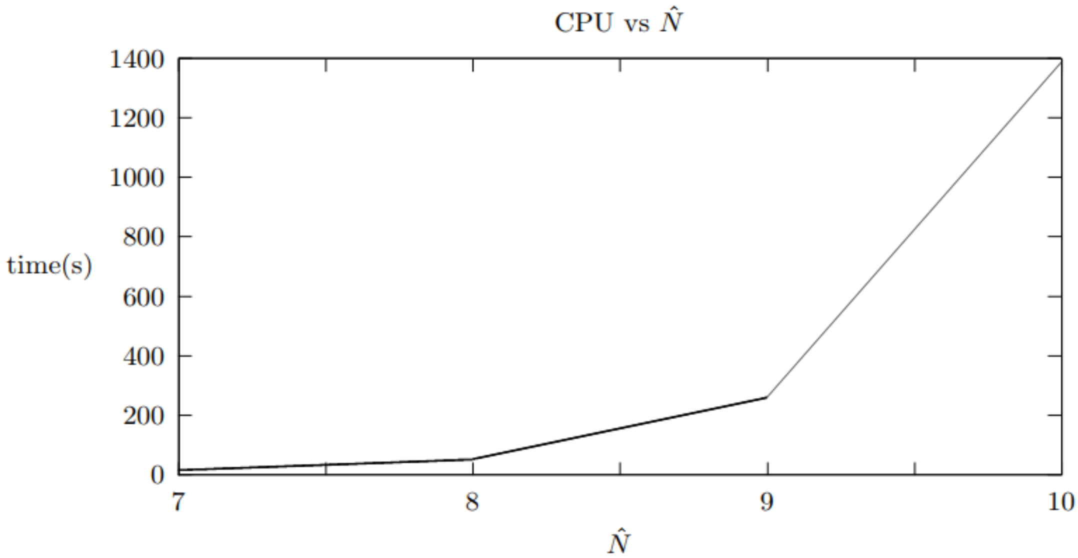


Figure 2.5 – *Average running times obtained by the Vehicle Event Model on the instances with* $|V| = 15$ *and* $\bar{T} = 120$*, for different values of* $\hat{N}$.

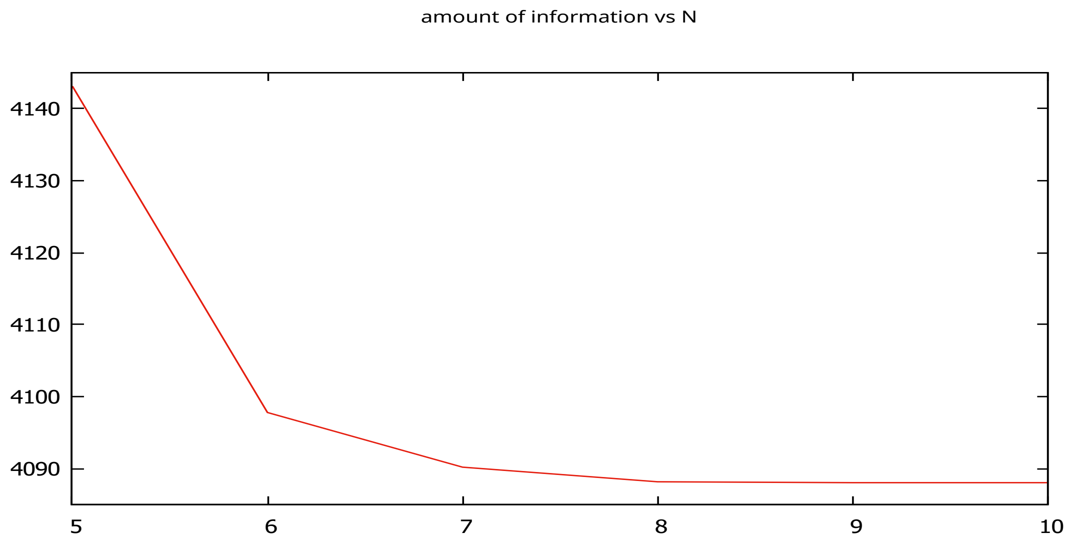


***Figure 2.6** – Average amount of information at time T at all nodes using the Event Vehicle Model with $|V| = 15$, $\bar{T} = 120$ versus different values of N.*

Chapter 3

Heuristics for a vehicle routing problem with information collection in wireless networks

Contents

3.1 Introduction

Wireless Networks (WN) have recently received great attention from the operations research (OR) community; as an example, we refer to the edition of a special issue in 2015 dedicated to reliable deployment techniques in Wireless Sensor Networks (Gomez-Pulido et Lanza-Gutierrez, 2015). Some applications defined on WN need to provide vehicle routing strategies with wireless information transmission to the vehicles involved. However, in these applications, innovation and research appears most in the development of routing protocols (Bhoi et al., 2017; Celik et Modiano, 2010; Moghadam et al., 2011; Velásquez-Villada et al., 2014) while there is still a gap in the development of vehicle routing strategies. Some very recent works helped filling this gap with the development of Mixed Integer Linear Programming (MILP) based exact methods (Basagni et al., 2014) and also one heuristic approach (Basagni et al., 2014). In this work, we contribute with the development of heuristics and heuristic strategies to a version of the VRP defined on WN. The problem treated in this chapter is exactly the problem described in Section 2.2, see figure (3.1).

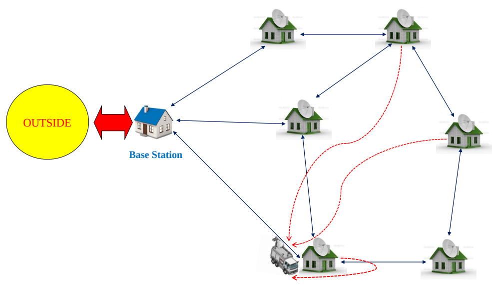


Figure 3.1 – The wireless transfer vehicle routing problem.

For the best of our knowledge, the authors in (Basagni et al., 2014) were the first ones to treat a variant of the WTVRP appearing in underwater wireless sensor networks. The authors considered a scenario with a set of surfacing and underwater nodes where they look for a routing to an autonomous underwater vehicle (AUV) during a given time period. The AUV must leave and return to a surface node while information generated by the set of underwater nodes is collected along a path that *physically* visits each station where information is collected. The information generated in a given underwater node i at a time point t_1 which arrives to a surface node at a time point t_2 has a given value $v_{it_1t_2}$. The strategy adopted in (Basagni et al., 2014) is the maximization of the value of the information collected. The authors proposed two solution approaches: an Integer Linear Programming (ILP) formulation able to solve the problem with up to 12 underwater nodes in a time that varies from a few hours to a few days; a greedy adaptive heuristic able to provide solutions to the problem with up to 35 underwater nodes.

A variant of the WTVRP is treated in chapter 1 , which differs in only one additional imposition: if a transfer starts in a station, then all the information available at that station at the beginning of the transmission needs to be extracted. In the chapter 1, three different objective functions are discussed and experiments are used to access how one strategy, i.e., the optimization of one objective function, affects the others and impacts the periodicity of the remaining information. In chapter 2, the exact solution of the WTVRP defined here is investigated. The adoption of the strategy that maximizes the amount of information collected at the end of time horizon allowed the introduction of three different MILP models: one discrete time model and two event based models. The discrete time model, that discretizes the routing and transfer times, provides the best results for small size instances. However, when the size of the instances increases a vehicle event model, where the size of the model depends on a parameter establishing a maximum number of visits, provided the best results. As we could expect, all the models fail to solve the problem to optimality for large size instances. In the present work, we use the vehicle event model and discuss several matheuristic based approaches with the aim of solving large instances of the WTVRP.

The development of commercial solvers and the increasing processing capacity of computers are making possible to optimally solve increasingly larger size MILP problems. When the instances are too large or too hard to solve new approaches based on matheuristics are becoming more popular. For matheuristics applied to complex routing problems see for instance (Agra et al., 2014; Wang et al., 2017). For surveys on matheuristics we refer to (Archetti et Speranza, 2014; Ball, 2011; K.F. et V., 2010).

Our contribution is as follows. We introduce three heuristics that construct an initial solution to the WTVRP. Two of them are matheuristics based on the computational efficient mathematical model introduced in chapter 2, and one is a greedy heuristic. Taking into account the specificities of these constructive heuristics, two improvement heuristics are discussed. As the size of the MILP model depends on the possible number of visits, the two matheuristics use the MILP model by setting a small number of visits. Hence, a first improvement heuristic, called *best insertion*, tests the insertion of new visits in the vehicle route obtained with the matheuristic. In order to improve the solution obtained from the greedy heuristic, a fix-and-optimize heuristic is provided. This heuristic fixes the vehicle route in the MILP model and solve the resulting restricted model. Finally, a general exchange heuristic that exchanges a number of consecutive visits is presented. The combination of constructive and improvement heuristics will lead to different heuristic strategy approaches. Computational results are conducted to test each heuristic strategy approach.

In Section 3.2 the constructive, the improvement heuristics and the strategies combining different types of heuristics are described. In Section 3.3 we describe the computational experiments carried out to compare the heuristic strategies.

3.2 Heuristics

When the number of stations is large, solving the vehicle event model to optimality requires to overestimate N^*, since N^* is not known. This implies to solve the vehicle event model for large values of N, leading to large size MILP models that, in general, cannot be solved within reasonable running time. We refer the reader to section 2.4, for a deep discussion on the performance of the MILP model. Here we discuss several heuristic strategies combining different types of heuristics that we will classify into constructive (Section 3.2.1) and improvement heuristics (Section 3.2.2). Most of these heuristics use the MILP vehicle event model.

3.2.1 Constructive heuristics

Here we describe three constructive heuristics designed to derive good initial feasible solutions: a simple combinatorial relaxation heuristic (Section 3.2.1), a fix-and-relax heuristic (Section 3.2.1); and a greedy heuristic (Section 3.2.1).

N-MILP heuristic

As discussed above, the size of the vehicle event model depends on the parameter $\hat{N}$ indicating the maximum number of vehicle stops. For small values of $\hat{N}$, the MILP model can be quickly solved using a commercial solver. However, imposing a small value for this parameter forces the solution procedure to act as a heuristic. The N-MILP heuristic consists in using the vehicle event model considering a relatively small value of $\hat{N}$. This will give the optimal solution, i.e., a vehicle route, with a maximum of $\hat{N}$ vehicle visits.

Fix-and-relax heuristic

This heuristic also uses the vehicle event model in order to define an initial route. In contrast with the N-MILP heuristic, a large value for $\hat{N}$ will be assumed. In each iteration k, all variables are relaxed except the path variables x_{jk} for $j \in V$, which remain binary. The resulting relaxed MILP is solved. Constraints (2.51) ensure there must exist at most a j_k such that $x_{j_k k}$ is equal to 1. We fix $x_{j_k k} = 1$ and $x_{jk} = 0$ for $j \neq j_k$, and the process is repeated until $j_k = 1$ (i.e. until the vehicle returns to the base station). With this procedure a path $R = (x_{j_1 1}, x_{j_2 2}, \ldots, x_{1s})$ is obtained for $s \leq N$. Finally, the route variables are fixed and the resulting restricted vehicle event model is solved (with the time variables restricted to be integer). The process is detailed in Algorithm 1.

Greedy heuristic

In the following, we present a greedy algorithm that constructs a vehicle route. Starting at the base station, in each iteration, the next visit is chosen in order to maximize the amount of information that can be extracted. Each iteration involves several choices: (i) which neighbor node to visit next; (ii) how long the vehicle shall stay in that node; (iii)

Algorithm 1: Fix-and-relax

$k \leftarrow 1$.
repeat
  Relax all integer variables except x_{jk}, for $j \in V$.
  Solve the relaxed model, and let $\bar{x}$ denote the resulting vector solution.
  Let j_k be the node index such that $\bar{x}_{j_k k} = 1$.
  Set $x_{j_k,k} \leftarrow 1$ and $x_{jk} \leftarrow 0$ for all $j \neq j_k$.
  Set $k \leftarrow k+1$.
until $j_k = 1$;
Solve the restricted model with all x_{jk} variables fixed.

which nodes will be selected to transfer information; (iv) how much information shall be collect from each node.

First, we consider a criterion to calculate the time of permanence at a given station j. This criterion depends upon the information that will be collected from each node. Let $(\bar{\beta}(j))$ denotes the vector $(\beta(j))$ ordered in decreasing order. Let $coll(j)$ denote the maximum amount of information that can be collected by a vehicle positioned in node j, assuming that station j and the stations in $range(j)$ (stations in the transfer range of j) have sufficient quantity of information to transfer at the maximum rate. That is, the maximum amount of information that can be collected from node j depends only on the transfer constraints (2.59) and multi-transfer constraints (2.60) and not of the quantity available at the nodes:

$$coll(j) = \min\{R, \sum_{l=1}^{\min\{M,|range(j)|\}} (\bar{\beta}(j))_\ell\}.$$

Let $coll_k(j,t)$ denote the maximum amount of information collected in time period t if the vehicle arrives at the end of time period $k-1$ and collects the maximum information during periods k to $t-1$. If the amount of information at the stations in $range(j)$ is large enough, $coll_k(j,k)$ will be equal to $coll(j)$. During the stop at station j, the amount of information collected at each consecutive period will decrease over time. If the vehicle arrives at the beginning of time period k at station j, the time spent at j will be denoted by $t_j(k)$, and it is obtained as follow:

$$t_j(k) = \min\{argmax_{t \geq k}\{coll_k(j,t) > l * coll(j)\}, (\bar{T} - t_{j1} - k)^+\}$$

where l is a parameter satisfying $0 < l < 1$ and $(z)^+ = \max\{0, z\}$. In the numerical results we consider $l = 0.8$. That is, the first argument in the min function ensures that the vehicle stays in node j while it can extract at least 80% of the maximum information that can be extracted from that node. As one of the problem restrictions enforces the vehicle to be at the base station at the end of the time horizon T, one needs to ensure (second argument of the min function) that a node can be visited at time k only if the minimum time needed to return to the base station, t_{j1}, is less than or equal to $\bar{T} - k$. The traveling times t_{j1} are computed by solving the shortest path problem from j to 1.

Now we consider the decision of which node to visit next. Assume that at the beginning of period k the vehicle is leaving station i, as shown in the Figure 3.2. The next station is chosen accordingly to the following average speed information transfer parameter:

$$transf(i,j,k) = \frac{\sum_{t=k+t_{ij}}^{k+t_{ij}+t_j(k+t_{ij})-1} coll_{k+t_{ij}}(j,t)}{t_{ij}+t_j(k+t_{ij})}.$$

The station with largest value of $transf(i,j,k)$ is selected.

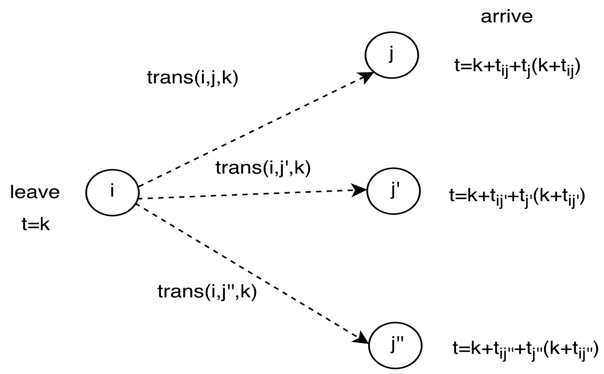


***Figure 3.2** – Choice of the neighbor station according to criterion of the greatest transfer.*

The algorithm stops when there is no candidate station to visit due to time limitations, since the vehicle needs to return to the base station. This situation can be identified when the vehicle is leaving node i, by verifying that $t_j(k+t_{ij}) = 0$ for all $j \in range(i)$. The full description of the greedy algorithm is given in Algorithm 2.

Algorithm 2: Greedy algorithm

```
i ← 1
k ← 1
STOP ← false
repeat
    Let j* ← argmax{transf(i,j,k) | j ∈ range(i)};
    if t_{j*}(k + t_{ij*}) > 0 then
        i ← j*
        k ← k + t_{ij*} + t_{j*}(k + t_{ij*})
    end
    else
        STOP ← true
    end
until STOP = true;
```

3.2.2 Improvement heuristics

In this section, we present heuristics that aim to improve an initial solution of the WTVRP; each heuristic developed to upgrade a given criteria. In that way, each heuristic will be suitable for a particular type of initial solution, for example, solutions based on short routes (with a small number of vehicle visits), or solutions obtained with a specific constructive algorithm. Hence, the improvement heuristics will be combined with the different constructive methods described in the previous section. Three improvement heuristics will be presented: a Fix-and-optimize heuristic (Section 3.2.2); a best insertion heuristic (Section 3.2.2); and an exchange heuristic (Section 3.2.2).

Fix-and-optimize

This heuristic finds the optimal transmission planning for a given (fixed) route; thus it finds a local optima of the WTVRP.

Let $\overline{x}$ denote a vector with the value of the routing variables in the given solution. The improvement is done by fixing the routing variables $x_{ik} = \overline{x}_{ik}$, for each pair $i \in V$ and $k \in N$, in the vehicle event model. Then the resulting restricted model is solved. The restricted model allows to adjust the time spent during each stay, at each of the visited nodes, as well as the quantities to transfer from each node during the stay at each node. Although the restricted model is a mixed integer program, it can be solved to optimality quickly (see results on Section 3.3).

This heuristic is suitable when the routing decisions were taken without considering the mathematical model. In our case, it will be more suitable to be combined with the greedy heuristic.

Best insertion heuristic

Consider a feasible route R with nodes $i_1 = 1, i_2, \ldots, i, k, \ldots, i_r = 1$ where r is the route length and i_l represents the node visited in position l. The process of inserting a node j into position l consists of adding a node to the path at position l, as shown the Figure 3.3.

After the insertion, a new route that includes one more node (with length $r + 1$) is obtained. To obtain the best insertion in position l of a given route, the MILP vehicle event model is used by setting $\hat{N} = r + 1$ and fixing all the positions of the route except the l^{th} visit. The routing variables are fixed as follows: $x_{i_k,k} = 1, k < l$ and $x_{i_k,k+1} = 1$ for $k > l$. To find the best possible insertion, all the possible positions from 1 to r are examined and the best one is chosen.

The insertion process is repeated until no improvement on the objective function is observed. This algorithm is detailed in Algorithm 3.

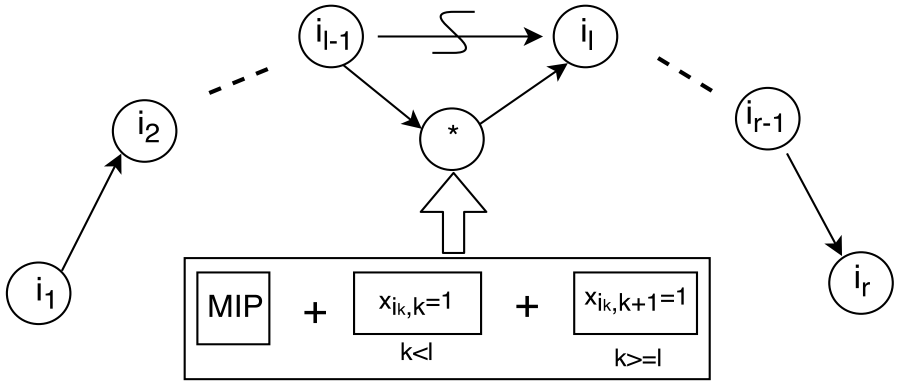


Figure 3.3 *– Insertion of the node j in the l^{th} position of the route.*

This heuristic is suitable to improve initial solutions considering a short route (with small number of visits). Thus it may be combined with constructive heuristics whose computational effort depends on the number of visits, which is, for instance, the case of the N-MILP heuristic.

Exchange heuristic

Consider an initial route R. In the exchange heuristic, at each iteration, a position k of the current route solution and a number l of nodes are selected. Then the nodes of route R visited in positions $k, k+1, \ldots, k+l-1$ are exchanged, using the following exchange procedure.

Exchange(k,l): Given a route R with nodes $i_1 = 1, i_2, \ldots, i_r = 1$, the procedure exchanges l nodes starting from the position k (nodes visited in the position $k, k+1, \ldots, k+l-1$) with a new set of nodes. In order to perform the nodes exchange, the routing variables $x_{i_t,t}$ are fixed to one, for $t = 1, \ldots, k-1$ and $t = k+l, \ldots, r$, and the restricted vehicle event model is solved.

The optimal solution for this MILP will give a new route $\bar{R}$ with objective function value $\bar{z}$, (see Figure 3.4 for an example with $l = 2$). As the initial route R is a feasible route for the restricted MILP model, then $\bar{z} \leq z$.

In each iteration of the exchange heuristic a route R is considered. The integer k is randomly generated between 1 and $r - l + 1$ and the exchange procedure *Exchange*(k,l) is used to obtain a new route. This process is repeated a certain number of iterations. The value of k is selected so that the same node is not repeated in two consecutive iterations. This algorithm is detailed in Algorithm 4.

On one hand, when l is large, the restricted MILP model becomes large and the solution approach becomes slow. On the other hand, with $l = 1$ there is the possibility of the *Exchange*(k,l) procedure obtain the initial route R because the graph may not be complete. Thus, there may be few nodes that are simultaneously neighbors from

Algorithm 3: Best insertion heuristic

Consider an initial feasible solution obtained with r visits
Let $\overline{R} \leftarrow (i_1 = 1, i_2, \ldots, i_{r-1}, i_r = 1)$ denote the route of the solution
Let $\overline{z}$ denote the value of the objective function of the solution
repeat
  $N \leftarrow r + 1$
  $z^* \leftarrow \overline{z}$
  $R^* \leftarrow \overline{R}$
  $\overline{z} \leftarrow \infty$ **for** *l from 2 to* $r - 1$ **do**
    Using R^*, set $x_{i_k,k} = 1, k < l$ and $x_{i_k,k+1} = 1$ for $k > l$
    Solve the restricted MILP model **if** *the optimal value,* z'*, of the restricted model is lower than* $\overline{z}$ **then**
      $\overline{z} \leftarrow z'$
      Set $\overline{R}$ as the vehicle route of the solution obtained
    end
  end
  $r \leftarrow r + 1$
until *No improvement in the objective function is observed* $(\overline{z} \geq z^*)$;

Algorithm 4: Exchange heuristic

Consider an initial route $R \leftarrow (i_1 = 1, i_2, \ldots, i_r = 1)$
$k1 \leftarrow r$
for *i from 1 to iter* **do**
  $k \leftarrow Random(1, r - 1)$ with $k \neq k1$
  $k1 \leftarrow k$
  Let R' be the routing solution obtained when applying $Exchange(k, l)$ to the route R
  Update $R \leftarrow R'$
end

the nodes visited in the $k - 1^{th}$ and $k + 1^{th}$ positions. In the computational results, we consider l equal to 2 and *iter* equal to 20.

3.2.3 Heuristic strategies

By combining different constructive and improvement heuristics, we face the possibility of deriving different heuristic strategies. However, as explained above, some improvement heuristics were designed to improve solutions with particular characteristics, thus obtained through particular constructive heuristics.

The N-MILP and fix-and-relax heuristics use the vehicle event model. Thus, they provide solutions which are optimal for the considered route (local optimum solutions). Conversely, the greedy heuristic is a combinatorial algorithm that doesn't use the MILP

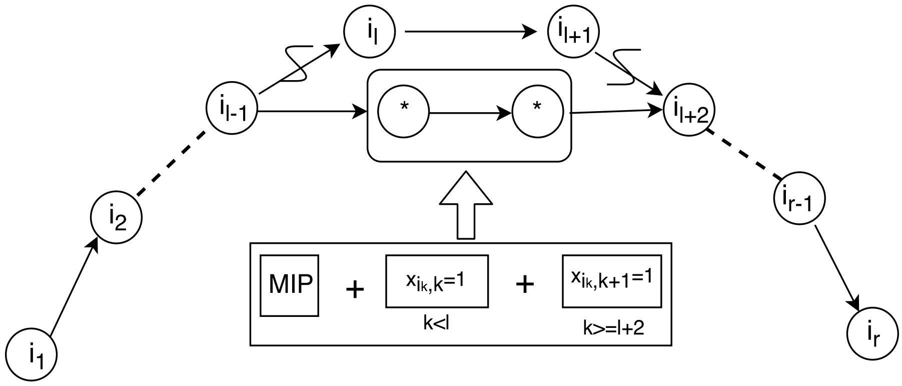


__Figure 3.4__ – Heuristic Exchange(n,2): exchange of two neighboring nodes by nodes j and j' starting at position n.

model and provides solutions that may not be optimal for the route obtained. Hence, the fix-and-optimize heuristic will be used only to improve the greedy algorithm, since it cannot be effective in improving routing solutions from the two other constructive heuristics.

The best insertion heuristic is useful to improve solutions using a small number of visits. It may have a greater impact when combined with the heuristics based on the MILP vehicle event model, since the running times of those heuristics will depend on the number of visits $\hat{N}$, and for small values of $\hat{N}$ they are in general fast. Hence, we will apply the best insertion heuristic to improve solutions obtained with the N-MILP and fix-and-relax heuristics.

The exchange heuristic is suitable to be applied to solutions obtained from any heuristic procedure. Here, we will use this heuristic to improve solutions already improved with the other improvement heuristics.

A general overview of the heuristics and their relations in order to derive full heuristic strategies is given in Figure 3.5.

<table>
<tr><th>Constructive heuristics</th><th colspan="2">Improvement heuristics</th></tr>
<tr><td>N-MILP</td><td rowspan="2">Best insertion</td><td rowspan="3">Exchange</td></tr>
<tr><td>Fix-and-relax</td></tr>
<tr><td>Greedy</td><td>Fix-and-optimize</td></tr>
</table>

__Figure 3.5__ – Combination of the heuristics procedures in order to derive different heuristic strategies.

From this discussion, we can derive several heuristic strategies that combine constructive with improvement heuristics:

- N-MILP followed by Best insertion,
- Fix-and-relax followed by Best insertion,

- Greedy followed by Fix-and-optimize,
- N-MILP followed by Best insertion followed by Exchange,
- Fix-and-relax followed by Best insertion followed by Exchange,
- Greedy followed by Fix-and-optimize followed by Exchange.

In the next section, we provide a computacional comparison of these strategies in order to identify the best approach for an instance of a given size.

3.3 Computational experiments

In this section, we report the computational tests conducted to evaluate heuristics and compare the several heuristic strategies that combine constructive with improvement heuristics introduced in Section 3.2.

All the experiments were performed using a server with 15 CPU's Intel ®Xeon (R) E5540@ 2.53Ghz X4, with 16 GB of RAM.

A set of instances was randomly generated as described in chapter 1 and 2. The vertices in V are located on a square grid of length $\ell = 8$. The base station is located on the bottom left vertex and the remaining stations are placed randomly on a square of length $\ell' = 6$, in the upper right of the grid. The distance matrix is given by the euclidean distance between the stations. The graph edges are selected randomly. In order to obtain a certain graph density d, starting from a complete graph, edges are removed randomly, while ensuring connectivity, until the desired graph density is obtained. In this work, we generate instances varying $|V|$ in $\{20, 50, 100\}$ to cover different size instances and with $d = 0.4$. We considered the values of $\bar{T} \in \{72, 120, 240\}$ and the data generation rates r_j are randomly generated in the interval $[1, 5]$. The values of α_{ij} were randomly generated in $\{1/12, 1/13, 1/14\}$ if $i = j$ and in $\{1/5, 1/6, 1/7\}$ otherwise. The following values parameters were set: $r_{cov} = 4$, $R = 20$ and $M = 3$.

3.3.1 Basic computational results for the heuristic approaches

In each table presented in this section, column (MILP) provides information on the solution obtained by solving the vehicle event model with CPLEX solver in a time limit of one hour.

Table 3.1 reports the results obtained with N-MILP heuristic. The first column gives the parameters used in the instance generation. The next three columns present the results obtained by solving the MILP vehicle event model. The second column ($\bar{z}$) gives the objective function value of the best solution found, the third column (Cpu) gives the running time in seconds (the asterisks mean the running time limit was attained) and the fourth column (DGap) gives the duality gap at the end of the execution (DGap$= \frac{\bar{z} - \underline{z}}{\bar{z}} \times 100$, where $\underline{z}$ is the best lower bound known). The last four columns report the results obtained with the N-MILP heuristic, where for 20 and 50 nodes we consider $\hat{N} = 5$ and for 100 nodes we consider $\hat{N} = 4$. The fifth column gives the objective function value of the best solution, the sixth column gives the corresponding vehicle route, column (Gap) shows the gap, in percentage, between the values obtained with the MILP model and the N-MILP heuristic (Gap $= \frac{z_{mip} - z_N}{z_{mip}} * 100$, where z_{mip} is the value presented in the second column and z_N is the value presented in the fifth column). A negative value means that the solution obtained with the N-MILP heuristic is better than the solution obtained with the MILP model. The last column gives the running time (in seconds).

***Table 3.1** – Computational results for the MILP model with a run time limit of one hour and the N-MILP heuristic.*

	MILP			N-MILP heuristic			
Parameter	$\bar{z}$	Cpu	DGap	$\bar{z}$	Route	Gap	Cpu
$\|V\| = 20$	4880,91	173,24	0	4944,66	5-18-8-12-1	1,31	6,62
$\bar{T} = 120$	5538,05	69,80	0	5596,26	18-14-2-6-1	1,05	4,58
$M = 8$	7722,55	61,77	0	7787,96	20-13-16-6-1	0,84	3,32
$R = 30$	6103,32	204,44	0	6170,30	19-10-16-5-1	1,10	7,69
	6002,21	131,28	0	6068,65	11-16-12-7-1	1,11	4,71
	5190,95	301,47	0	5253,14	8-20-17-13-1	1,20	7,13
	4880,42	96,17	0	4928,08	9-6-13-19-1	0,98	4,56
	4750,10	101,54	0	4889,72	3-4-17-7-1	2,94	3,69
	4459,60	41,69	0	4691,88	9-6-19-2-1	5,21	3,92
	6548,51	175,71	0	6661,82	12-2-17-14-1	1,73	6,25
$\|V\| = 50$	17776,60	***	12,23	17722,30	17-41-44-18-1	-0,30	96,49
$\bar{T} = 120$	16140,40	***	14,36	16128,70	30-5-11-16-1	-0,07	237,55
$M = 8$	16757,80	***	13,45	16737,60	31-46-19-7-1	-0,12	246,64
$R = 30$	15009,50	***	14,30	15068,40	42-17-6-25-1	0,39	258,63
	15985,20	***	14,12	16016,00	5-34-45-12-1	0,19	214,92
	16578,10	***	14,14	16565,40	30-25-44-39-1	-0,07	291,98
	16174,60	***	15,06	16285,10	6-14-22-31-1	0,68	242,99
	17997,70	***	13,66	18012,00	4-16-47-2-1	0,08	335,79
	17554,00	***	12,69	17611,30	28-49-48-46-1	0,32	329,39
	17557,00	***	13,76	17588,60	33-4-11-37-1	0,18	244,82
$\|V\| = 100$	55740,40	***	9,45	56406,79	11-60-26-1	1,19	180,65
$\bar{T} = 200$	56446,00	***	9,26	57207,39	37-35-54-1	1,35	135,81
$M = 12$	52327,00	***	9,94	53202,40	83-65-7-1	1,67	170,38
$R = 50$	56939,80	***	9,85	57675,49	32-99-43-1	1,29	972,52
	55578,00	***	9,24	56094,60	39-17-70-1	0,93	154,89
	57665,20	***	9,40	58206,40	21-53-48-1	0,94	149,71
	51226,00	***	9,97	51860,80	67-31-42-1	1,24	227,07
	53887,20	***	9,40	54514,99	19-100-52-1	1,16	94,75
	56568,20	***	9,71	57131,80	27-90-3-1	0,99	104,99
	56449,40	***	9,45	57122,60	48-80-24-1	1,19	673,39

We can see that only for two instances the value of the best solution obtained with the N-MILP heuristic was more than 2% higher than the best solution obtained with the MILP model. In four instances (all with 50 nodes) the N-MILP heuristic provided the best solution. As we exepcted, the N-MILP heuristic runs fast: always below 8 seconds for 20 nodes, 6 minutes for 50 nodes, and 12 minutes for 100 nodes.

Table 3.2 reports the results obtained with the greedy heuristic. The first two columns repeat information given in the corresponding column of Table 3.1. The following three columns give the objective function value, the gap and the route of the best solution obtained with the greedy algorithm. Again, the gap measures the relative difference, in percentage, between the value of the greedy solution and the value given in column MILP. The last two columns give the objective function value and the corresponding gap of the solution obtained with the greedy heuristic followed by the fix-and-optimize heuristic.

Table 3.2 *– Computational results obtained with the greedy heuristic and with the greedy heuristic followed by the fix-and-optimize improvement.*

	MILP	Greedy heuristic			Greedy+FO	
$\|V\|$	$\bar{z}$	$\bar{z}$	Gap	Route	$\bar{z}$	Gap
20	4880,91	4897,49	0,34	5-18-8-17-8-1	4881,07	0,01
	5538,05	5713,06	3,16	6-17-2-16-1	5598,44	1,09
	7722,55	7818,96	1,25	20-7-13-18-1	7776,75	0,70
	6103,32	6294,46	3,13	16-10-6-5-11-1	6241,89	2,27
	6002,21	6278,63	4,60	7-10-3-14-7-1	6258,21	4,27
	5190,95	5358,59	3,23	8-15-20-13-20-1	5276,09	1,64
	4880,42	5080,04	4,09	9-13-19-12-1	4980,57	2,05
	4750,10	4872,56	2,58	7-17-4-3-5-19-1	4868,97	2,50
	4459,60	4655,10	4,38	10-15-18-4-16-2-1	4636,90	3,98
	6548,51	6772,08	3,41	2-17-9-10-6-1	6684,60	2,08
50	17776,60	17779,20	0,01	18-44-41-17-1	17737,40	-0,22
	16140,40	16260,20	0,74	14-50-49-30-24-19-1	16185,40	0,28
	16757,80	16951,04	1,15	7-11-16-31-1	16924,64	1,00
	15009,50	15225,90	1,44	7-10-6-10-1	15185,40	1,17
	15985,20	16250,20	1,66	12-9-31-36-1	16177,30	1,20
	16578,10	16673,00	0,57	39-22-25-44-39-1	16574,30	-0,02
	16174,60	16502,20	2,03	6-4-18-31-1	16483,20	1,91
	17997,70	18311,70	1,74	40-35-18-23-14-1	18255,90	1,43
	17554,00	17672,60	0,68	28-48-27-22-1	17629,20	0,43
	17557,00	17676,60	0,68	32-5-11-37-1	17655,10	0,56
100	55740,40	56027,20	0,51	34-64-60-11-40-65-78-19-1	55969,60	0,41
	56446,00	56713,40	0,47	37-21-19-12-95-5-35-97-1	56601,00	0,27
	52327,00	53073,40	1,43	7-83-51-37-22-16-8-91-7-1	52818,20	0,94
	56939,80	57395,60	0,80	40-38-84-99-43-47-25-99-32-1	57318,70	0,67
	55578,00	56274,40	1,25	59-20-40-17-29-89-91-92-26-1	56095,40	0,93
	57665,20	58063,40	0,69	48-41-67-53-94-60-40-4-85-1	57868,40	0,35
	51226,00	51750,40	1,02	15-45-58-65-55-54-29-4-1	51674,20	0,87
	53887,20	54245,80	0,67	52-17-31-63-72-32-100-67-75-1	54017,40	0,24
	56568,20	56849,00	0,50	31-32-74-24-67-78-9-31-1	56615,10	0,08
	56449,40	57315,40	1,53	42-25-73-24-78-54-8-38-1	57180,80	1,30

We can see that the fix-and-optimize heuristic always improve the greedy solution. The combination of the two heuristics provide solutions whose objective function values are very close to the one given in column MILP, specially for large size instances with 50 and 100 nodes. For 100 nodes, the relative difference is always below 1% except for one instance. We can also observe that for 100 nodes the greedy solution tends to add more visits than the N-MILP heuristic.

In Table 3.3, we report the results obtained with the fix-and-relax heuristic. For

20, 50 and 100 nodes, parameter $\hat{N}$ was set to 6, 7 and 8, respectively. The last four columns give the values of the best solution, the corresponding vehicle route, the gap between the objective function value and the value of the best solution (presented in the second column), and the running time (in seconds), respectively, obtained with the fix-and-relax heuristic.

***Table 3.3** – Computational results obtained with the fix-and-relax heuristic.*

	MILP	fix-and-relax			
$\|V\|$	$\bar{z}$	$\bar{z}$	route	Gap	cpu
20	4880,91	4953,04	12-8-18-11-12-1	1,48	7,27
	5538,05	5636,80	6-12-11-18-4-1	1,78	6,59
	7722,55	7875,96	19-7-13-3-19-1	1,99	6,29
	6103,22	6180,02	16-10-6-5-14-1	1,26	9,68
	6002,21	6097,97	7-12-16-2-7-1	1,60	8,14
	5190,95	5247,81	8-15-17-20-13-1	1,10	11,00
	4880,42	5059,50	9-14-12-19-12-1	3,67	7,72
	4750,10	4993,72	4-5-3-4-3-1	5,10	7,84
	4459,60	5154,30	11-9-2-9-11-10-1	15,58	6,36
	6548,51	6664,75	2-17-12-14-8-1	1,77	8,36
50	17776,60	17761,60	18-41-44-41-17-4-1	-0,08	173,18
	16140,40	16119,10	16-5-11-43-30-40-1	-0,13	49,84
	16757,80	16796,10	7-15-31-7-10-7-1	0,22	73,18
	15009,50	15152,00	10-27-15-7-43-30-1	0,95	44,39
	15985,20	15990,60	12-45-40-11-12-1	0,03	45,72
	16578,10	16556,20	39-25-44-39-13-1	-0,13	52,57
	16174,60	16378,50	6-4-24-36-2-1	1,26	51,43
	17997,70	18022,60	40-16-47-30-2-1	0,14	49,04
	17554,00	17618,90	28-48-3-27-3-1	0,37	48,63
	17557,00	17663,70	32-34-11-4-35-43-1	0,61	103,81
100	55740,40	55974,60	34-28-68-40-11-10-26-1	0,42	1774,37
	56446,00	56600,11	22-95-37-35-54-20-95-1	0,27	1988,06
	52327,00	52827,40	7-61-65-83-58-92-1	0,96	1446,43
	56939,80	57010,90	40-43-99-32-25-47-69-1	0,12	1404,87
	55578,00	55395,40	39-17-62-59-70-39-83-1	-0,33	1929,53
	57665,20	57853,00	53-67-41-7-10-89-1	0,33	1210,01
	51226,00	51244,90	42-45-58-4-35-15-38-1	0,04	1786,29
	53887,20	54163,80	2-48-100-19-36-74-37-1	0,51	1620,97
	56568,20	56513,30	31-53-32-93-39-3-27-1	-0,10	1604,13
	56449,40	56843,60	46-13-94-24-80-78-12-1	0,70	1447,75

From the Gap column, we can see that for the easiest instances (with 20 nodes), the performance of the fix-and-relax heuristic is clearly worst than solving the MILP model with a time limit of one hour. However for 50 and 100 nodes, the heuristic provides solutions with a gap below 1% in all but one instance, and for five instances it provides a better solution than the one obtained with the MILP model. The running times increase with the increase of the number of nodes. However, even for the 100 nodes case, the running times are always below the 2000 seconds.

In Table 3.4, we report the results obtained with the two constructive heuristics based on the event vehicle model combined with the best insertion heuristic. From the third to sixth column, we report the results obtained with the constructive N-MILP heuristic. Column (N-MILP) gives the value of the solution obtained with the N-MILP heuristic, and the following three columns give the information (objective function value, gap, and running time) corresponding to the solution obtained with the heuristic approach combining the N-MILP heuristic (used to obtain the initial solution) with the best insertion heuristic (used to improve the initial solution). The last four columns report similar information obtained with the fix-and-relax heuristic combined with the best insertion heuristic. In this case, the initial solution is obtained with fix-and-relax heuristic and its objective function value is reported in column (fix&relax).

***Table 3.4** – Computational results with the N-MILP heuristic and the fix-and-relax heuristic combined with the best insertion heuristic.*

	MILP	N-MILP + best insertion				fix-and-relax + best insertion			
$\|V\|$	$\bar{z}$	N-MILP	$\bar{z}$	Gap	Cpu	fix&relax	$\bar{z}$	Gap	Cpu
20	4880,91	5145,95	4918,15	0,76	26,77	5165,59	5021,14	2,87	14,66
	5538,05	5718,10	5622,62	1,53	12,13	5674,89	5622,62	1,53	11,93
	7722,55	7927,39	7768,47	0,59	10,35	8019,82	7789,89	0,87	12,29
	6103,32	6283,85	6103,32	0,00	9,65	6295,59	6226,46	2,02	14,11
	6002,21	6189,77	6050,54	0,8	9,78	6326,21	6192,29	3,17	9,74
	5190,95	5374,00	5190,97	0,00	12,50	5295,01	5253,89	1,21	11,43
	4880,42	5002,92	4906,61	0,54	10,55	5049,03	4986,19	2,17	12,74
	4750,10	5001,32	4809,34	1,25	11,75	5143,65	4932,49	3,84	12,86
	4459,60	4560,40	4560,40	2,26	11,94	5165,28	4578,02	2,66	21,02
	6548,51	6737,40	6657,30	1,66	10,13	6785,39	6637,74	1,36	12,96
50	17776,60	17829,20	17736,10	-0,23	43,65	17763,00	17724,9	-0,29	43,88
	16140,40	16252,40	16119,80	-0,13	30,89	16282,10	16198,40	0,36	47,22
	16757,80	16852,90	16750,40	-0,04	36,64	16771,30	16757,60	0,00	65,63
	15009,59	15170,70	15038,70	0,19	46,78	15109,90	15076,50	0,45	48,79
	15985,20	16115,70	15970,20	-0,09	48,93	15989,70	15940,30	-0,28	61,73
	16578,10	16697,40	16540,00	-0,23	35,45	16615,99	16610,20	0,19	58,71
	16174,60	16448,20	16305,80	0,81	34,62	16380,13	16361,60	1,16	55,17
	17997,70	18127,30	18049,30	0,29	37,19	18102,40	18093,60	0,53	52,18
	17554,00	17690,50	17611,80	0,33	36,94	17612,70	17612,70	0,33	35,08
	17557,00	17737,90	17640,50	0,48	32,82	17680,70	17654,70	0,56	71,66
100	55740,40	56985,20	55639,80	-0,18	267,91	56087,20	55538,00	-0,36	1580,41
	56446,00	57729,40	56326,00	-0,21	446,72	56682,70	56343,30	-0,18	1267,04
	52327,00	53827,60	52333,80	0,01	155,84	52467,40	52158,50	-0,32	918,25
	56939,80	58219,40	56947,40	0,01	285,15	57185,80	56941,60	0,00	1252,38
	55578,00	56568,60	55255,40	-0,58	331,53	55454,20	55319,39	-0,47	1091,69
	57665,20	58592,60	57582,40	-0,14	221,79	57789,40	57255,20	-0,71	1029,93
	51226,00	52186,20	50988,10	-0,46	1150.93	51393,40	51013,40	-0,42	1954,06
	53887,20	55275,80	53896,00	0,02	391,52	54426,70	53824,00	-0,12	1033,62
	56568,20	57680,60	56383,00	-0,33	362,81	56774,70	56375,50	-0,34	873,65
	56449,40	57611,80	56511,20	0,11	235,50	56850,00	56421,30	-0,05	1027,11

Again, the gaps show that for the easiest instances (with 20 nodes), the performance of the two heuristic strategies tested (N-MILP combined with best insertion and fix-and-relax combined with best insertion) provide worst solutions than solving the MILP model with a time limit of one hour. However for 50 and 100 nodes, both heuristic strategies are very competitive in terms of quality of the solution when compared against solving the MILP model. Both approaches are better in ten instances and worst in the remaining ten. However, the running times are much lower than the one hour spent in solving the MILP model. Between the two approaches it is not clear which one provides the best solutions. However, considering the running times for 100 nodes, the strategy based on the fix-and-relax is clearly slower than the one using the N-MILP heuristic.

Table 3.5 compares the greedy solution improved with the exchange heuristic against the greedy solution improved with the best insertion heuristic and the exchange heuristic.

Again, when the number of nodes increases, the greedy heuristic combined with the improvement heuristics becomes more competitive than solving the vehicle event MILP model with a time limit of one hour. The running times are always lower (always below 1500 seconds) and, for 100 nodes, the objective function values are in general better than the ones obtained with the MILP model. Between the two tested approaches, none of the approaches is clear better than the other.

3.3.2 Graphical comparison of the best heuristic approaches

In this section, we compare the constructive heuristics as well as the constructive heuristics combined with the improvement heuristics. The comparison is done with respect to two parameters: the quantity of information remaining in the stations at time m (corresponding to figures (a)) and the running time (corresponding to figures (b)). The comparison is performed for $m \in \{20, 50, 100\}$ The results report average values obtained over all the tested instances.

Figures 3.6-3.8 compare the constructive heuristics with the exception that the greedy heuristic includes the fix-and-optimize procedure that allows to obtain an initial local optimum solution (since, as explained in Section 3.2.2, the solution is optimal for the given route). From these figures we can see that for $|V| = 20$ and $|V| = 50$ the N-MILP heuristic gives the best results. Even for $|V| = 50$ the N-MILP heuristic generates solutions with average running time of 50 seconds, whose value is close to the best solution with time limit of one hour. However, for $|V| = 100$ the greedy heuristic provides better solutions than the N-MILP heuristic and spends less computational time. The fix-and-relax heuristic provides poor results for $|V| = 20$. However, for $|V| = 100$ it generates the best solutions among the constructive heuristics but the running times are very hight.

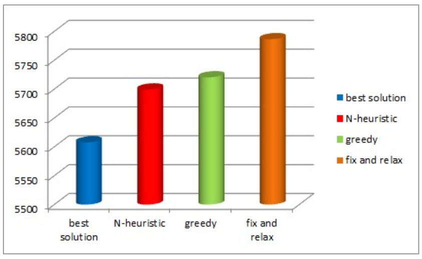
Next, based on the previous results, we compare the best heuristic approaches combining the constructive and improvement heuristics. This includes the following heuristic strategies: N-MILP followed by Best insertion; fix-and-relax followed by Best insertion; N-MILP followed by Best insertion followed by Exchange; and Greedy followed by fix-and-optimize followed by Exchange. The strategy S5, i.e., the fix-and-relax followed by Best insertion followed by Exchange, is not presented since the running times without the Exchange heuristic are already too hight. Figures 3.9-3.11 present this comparison.

In these graphics, we denote the N-MILP heuristic combined with the best insertion heuristic as (N+best); the fix-and-relax combined with the best insertion as (fix+best), the greedy heuristic combined with the fix-and-optimize and the exchange heuristic as (Gr+Exch) and the N-MILP heuristic followed by the best insertion heuristic and combined with the exchange heuristic as (N+Ins+Exch).

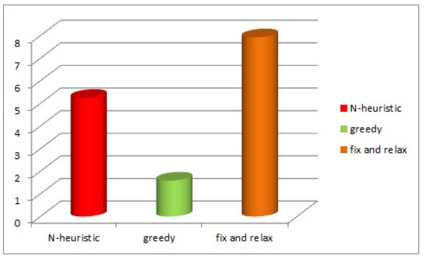
We can observe that, for the easiest instances, with $\mid V \mid = 20$, the MILP model provides the best solutions although all the heuristic approaches are quite fast. However, when $\mid V \mid$ increases, the quality of the MILP solution decreases and the heuristic strategies become more competitive. For $\mid V \mid = 100$, all the heuristic strategies provide better solutions than the MILP model within a time limit of one hour. The greedy heuristic combined with the fix-and-optimize and the exchange heuristic provides the best solutions while the N-MILP heuristic combined with the best insertion is the fastest approach.

Table 3.5 *– Computational results with the greedy heuristic combined with the best insertion and the exchange heuristic*

	MILP	Greedy + Exchange			N-MILP+B.Ins+Exchange		
$\|V\|$	$\bar{z}$	$\bar{z}$	Cpu	Gap	$\bar{z}$	Cpu	Gap
20	4880,91	4880,91	12,37	0,00	4918,15	21,02	0,76
	5538,05	5538,06	12,08	0,00	5557,67	14,01	0,35
	7722,55	7729,48	15,07	0,09	7774,60	15,16	0,67
	6103,32	6103,32	13,38	0,00	6103,32	22,04	0,00
	6002,21	6029,38	12,12	0,45	6050,53	23,40	0,81
	5190,95	5190,97	16,53	0,00	5190,97	29,22	0,00
	4880,42	4980,55	8,93	2,05	4900,00	19,51	0,40
	4750,10	4756,01	13,99	0,12	4750,15	23,98	0,00
	4459,60	4515,21	13,40	1,25	4442,00	33,83	-0,39
	6548,51	6572,84	15,82	0,37	6555,12	22,59	0,10
50	17776,60	17737,40	43,96	-0,22	17736,10	63,72	-0,23
	16140,40	16093,90	51,85	-0,29	16114,90	65,12	-0,16
	16757,80	16737,60	49,84	-0,12	16750,40	45,00	-0,04
	15009,50	15089,40	41,23	0,53	15009,00	118,43	0,00
	15985,20	16037,00	55,30	0,32	15964,70	101,91	-0,13
	16578,10	16516,90	55,61	-0,37	16496,30	139,28	-0,49
	16174,60	16289,30	40,56	0,71	16196,10	87,83	0,13
	17997,70	17937,50	57,51	-0,33	18012,00	93,27	0,08
	17554,00	17611,30	45,42	0,33	17599,60	112,63	0,26
	17557,00	17588,60	51,82	0,18	17555,70	87,50	-0,01
100	55740,40	55516,80	557,92	-0,40	55500,60	846,60	-0,43
	56446,00	56305,40	660,80	-0,25	56256,20	993,86	-0,34
	52327,00	52021,60	1011,37	-0,58	52192,00	605,40	-0,26
	56939,80	56693,80	918,44	-0,43	56727,60	911,75	-0,37
	55578,00	55238,40	1145,41	-0,61	55255,60	1268,06	-0,58
	57665,20	57383,40	502,63	-0,49	57450,80	626,73	-0,37
	51226,00	51075,20	730,26	-0,29	50954,40	1420,84	-0,53
	53887,20	53647,00	582,47	-0,45	53885,80	376,26	0,00
	56568,20	56253,00	448,97	-0,56	56256,50	979,76	-0,55
	56449,40	56362,60	429,55	-0,15	56278,90	729,45	-0,30

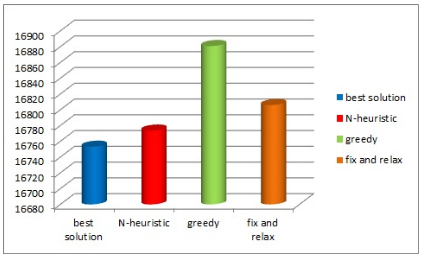


(a) remaining

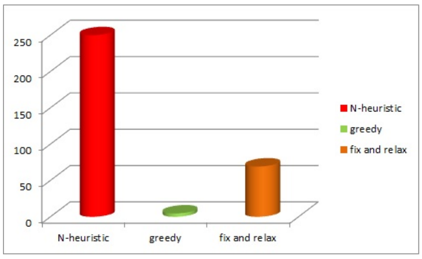


(b) cpu

***Figure 3.6** – Comparison of constructive heuristic approaches on instances with* $| V |= 20, \bar{T} = 120$.

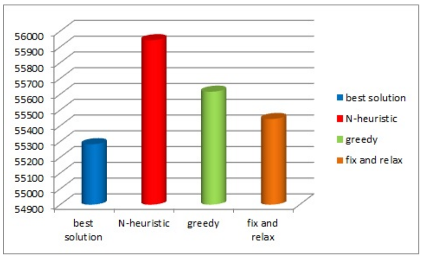


(a) remaining

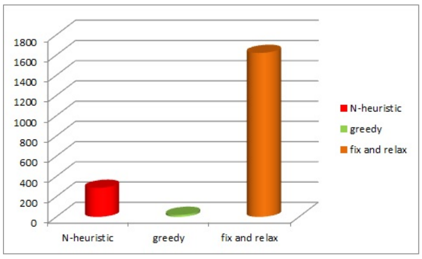


(b) cpu

***Figure 3.7** – Comparison of constructive heuristic approaches on instances with* $| V |= 50, \bar{T} = 120$.

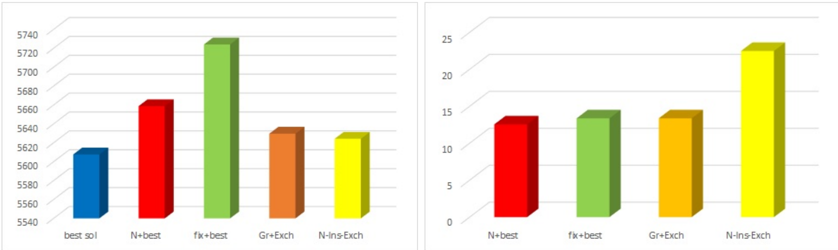


(a) remaining



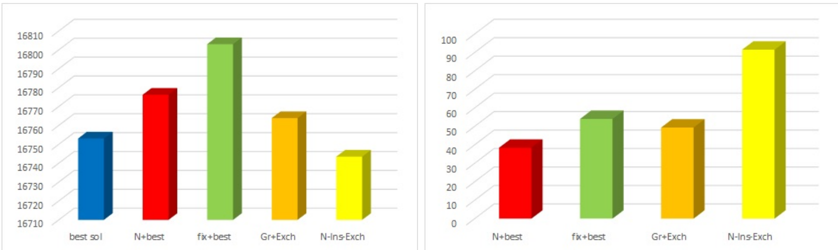
(b) cpu

***Figure 3.8** – Comparison of constructive heuristic approaches on instances with* $| V |= 100, \bar{T} = 200$.



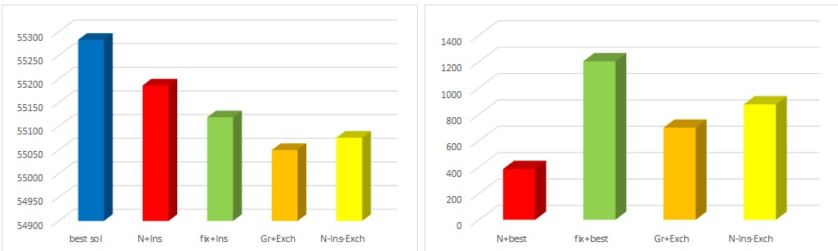
(a) remaining (b) cpu

***Figure 3.9** – Comparison of heuristic strategies on instances with* $\mid V \mid= 20, \bar{T} = 120$.



(a) remaining (b) cpu

***Figure 3.10** – Comparison of heuristic strategies on instances with* $\mid V \mid= 50, \bar{T} = 120$.



(a) remaining (b) cpu

***Figure 3.11** – Comparison of heuristic strategies on instances with* $\mid V \mid= 100, \bar{T} = 200$.

Conclusion

In this thesis, we have introduced the WT-VRP, a version of the VRP defined on wireless networks. The new characteristic added here to this well studied problem is the fact that information can be delivered without physically visiting a node in the network. A MILP formulation was provided to model the feasible space of the WT-VRP.

In the first chapter we discussed three different criteria to measure the efficiency of a solution which results in three different objective functions for the MILP formulation. Computational experiments were conducted on random instances. We conclude that the WT-VRP becomes more difficult as the total amount of information collected increases and, as we could expect, as the size of the time period increases. Our MILP formulation is computationally easier to solve when the average (over time) of the amount of information collected is maximized (**FO2**). However, we saw that the optimal solutions obtained with this criteria impacts the average satisfaction of the network (**FO3**). We have also studied the periodicity of the solutions obtained by each different efficiency criteria. In average, the solutions obtained with **FO3** are more suitable to achieve periodicity. The results obtained in the first chapter are important to access the hardness of the WT-VRP.

In the second chapter three mixed integer programming models are introduced. The first model DT is based on a time discretization, where each decision is a multiple of the time unity. The second model NE is an event model, where the visits to stations and transfer operations are modeled as events. Finally, the third model VE considers the vehicle stops as events. A computational study based on randomly generated instances was conducted to compare the three models. The DT model presents a high number of both variables and constraints while the NE and VE models need to be feed with parameters for the maximum number of events permitted. The results show that the NE model is always the worst and that the best model (DT or VE) depends on the instance. For shorter time horizons the DT model performs well, while for longer time horizons the VE is usually faster. The performance of the VE model is better when the optimal number of vehicle stops can be estimated.

Since most of the practical instances cannot be solved to optimality within a reasonable amount of computational time (see chapter 1,1), we propose in the last chapter several heuristic approaches that combine both constructive and improvement heuristics. In order to derive initial feasible solutions, three constructive heuristics are proposed. Two of them use the MILP model and the other is a greedy heuristic. Three improvement heuristics are also proposed. These heuristics were derived to improve the initial solutions and take into account the particularities of the constructive algorithm used to obtain the initial solution. A fix-and-optimize heuristic fixes the routing decisions of the initial solution and solves the resulting restricted MILP model. This heuristic is used to improve the initial solution obtained with the greedy algorithm since this algorithm doesn't take into account the MILP model. As the size of the MILP model depends on the maximum possible number of visits, $\hat{N}$, the constructive heuristics based on the MILP model are fast for small values of $\hat{N}$. Thus an improvement heuristic that starts form an initial route with a small number of vehicle visits and iteratively tests the inclusion of another visit is proposed. Finally an exchange heuristic that exchanges a consecutive set of nodes by new ones is proposed. This heuristic is combined with all the constructive heuristics.

Computational tests have shown that for the easiest instances, with a small number of nodes and time periods, good quality solutions can, in general, be obtained by solving the MILP model using a solver with a running time limit of one hour. However, when the instances are larger, this approach tends to be poor and to be outperformed by the heuristic strategies that combine the constructive heuristics with the improvement heuristics. In particular, for the largest instances with $| V |= 100$ nodes and $m = 200$, periods the greedy heuristic combined with a fix-and-optimize and an exchange heuristics provided the best solutions with average running times close to 10 minutes.

Perspectives

This section proposes some ways to complete the studies we have done or to improve the methods proposed. These tracks essentially concern four complementary aspects:

Sending information from the vicinity of a station: The vehicle could send information not only when it is located in a station but also when the vehicle is in a position near a station, which means, in a defined station neighborhood.

Increase the number of vehicles: One of the ways in which the work could be continued is adding more vehicles for the collection of information. This is necessary if we consider that some vehicles, for example drones, have limited energy. In this case, by imposing that only one vehicle collects the information, it would cause an accumulation of information not collected in the stations.

Add uncertainties: The work can be improved by adding uncertainty in the time it takes a vehicle to travel from one station to another and/or in the generation of information in each station.

Add more base stations: In this work we have only considered a single base station, we could improve the model by considering more than one base station. In this case, the vehicles could start at one base station and end the route in another.

List of Figures

List of Tables

Bibliography

(Agra et al., 2013a) A. Agra, H. Andersson, M. Christiansen, & L. A. Wolsey, 2013a. A maritime inventory routing problem: Discrete time formulations and valid inequalities. *Networks 64*, 297–314.

(Agra et al., 2013b) A. Agra, M. Christiansen, & A. Delgado, 2013b. Mixed integer formulations for a short sea fuel oil distribution problem. *Transportation Science 47*, 108–124.

(Agra et al., 2017) A. Agra, M. Christiansen, & A. Delgado, 2017. Discrete time and continuous time formulations for a short sea inventory routing problem. *Optimization and Engineering 18*(1), 269–297.

(Agra et al., 2014) A. Agra, M. Christiansen, A. Delgado, & L. Simonetti, 2014. Hybrid heuristics for a short sea inventory routing problem. *European Journal of Operational Research 236*(3), 924–935.

(Agra et al., 2013) A. Agra, M. Christiansen, R. Figueiredo, L. Hvattum, M. Poss, & C. Requejo, 2013. The robust vehicle routing problem with time windows. *Computers & Operations Research 40*(3), 856–866.

(Agra et al., 2012) A. Agra, M. Christiansen, R. Figueiredo, L. M. Hvattum, M. Poss, & C. Requejo, 2012. Layered formulation for the robust vehicle routing problem with time windows. Dans les actes de *Combinatorial Optimization - Second International Symposium, ISCO 2012, Athens, Greece, April 19-21, 2012, Revised Selected Papers*, 249–260.

(Akyildiz et al., 2002) I. Akyildiz, W. Su, Y. Sankarasubramaniam, & E. Cayirci, 2002. Wireless sensor networks: a survey. *Computer networks 38*(4), 393–422.

(An et al., 2015) Y. An, Y. Zhang, & B. Zeng, 2015. The reliable hub-and-spoke design problem: Models and algorithms. *Transportation Research Part B: Methodological 77*, 103 – 122.

(Archetti et Speranza, 2014) C. Archetti & M. G. Speranza, 2014. A survey on matheuristics for routing problems. *EURO Journal on Computational Optimization 2*(4), 223–246.

(Baldacci et al., 2008) R. Baldacci, M. Battarra, & D. Vigo, 2008. Routing a heterogeneous fleet of vehicles. Dans les actes de *The vehicle routing problem: latest advances and new challenges*, 3–27. Springer.

(Ball, 2011) M. O. Ball, 2011. Heuristics based on mathematical programming. *Surveys in Operations Research and Management Science 16*(1), 21 – 38.

(Basagni et al., 2014) S. Basagni, L. Bölöni, P. Gjanci, C. Petrioli, C. Phillips, & D. Turgut, 2014. Maximizing the value of sensed information in underwater wireless sensor networks via an autonomous underwater vehicle. Dans les actes de *Proceedings of IEEE INFOCOM'14*, 988–996.

(Bhoi et al., 2017) S. K. Bhoi, P. M. Khilar, & M. Singh, 2017. A path selection based routing protocol for urban vehicular ad hoc network (uvan) environment. *Wireless Networks 23*(2), 311–322.

(Biswas et Morris, 2004) S. Biswas & R. Morris, 2004. Opportunistic routing in multi-hop wireless networks. *ACM SIGCOMM Computer Communication Review 34*(1), 69–74.

(Bräysy et Gendreau, 2005) O. Bräysy & M. Gendreau, 2005. Vehicle routing problem with time windows, Part I: Route construction and local search algorithms. *Transportation science 39*(1), 104–118.

(Cancela et al., 2015) H. Cancela, A. Mauttone, & M. Urquhart, 2015. Mathematical programming formulations for transit network design. *Transportation Research Part B: Methodological 77*, 17 – 37.

(Celik et Modiano, 2010) G. Celik & E. Modiano, 2010. Dynamic vehicle routing for data gathering in wireless networks. Dans les actes de *49th IEEE Conference on Decision and Control (CDC)*, 2372–2377. IEEE.

(Dethloff, 2001) J. Dethloff, 2001. Vehicle routing and reverse logistics: the vehicle routing problem with simultaneous delivery and pick-up. *OR-Spektrum 23*(1), 79–96.

(Fall, 2003) K. Fall, 2003. A delay-tolerant network architecture for challenged internets. Dans les actes de *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, 27–34. ACM.

(Gendreau et al., 1996) M. Gendreau, G. Laporte, & R. Séguin, 1996. Stochastic vehicle routing. *European Journal of Operational Research 88*(1), 3–12.

(Gomez-Pulido et Lanza-Gutierrez, 2015) Gomez-Pulido, J. A. & J. M. Lanza-Gutierrez (Eds.), 2015. *Journal of Heuristics*, Chapter Heuristics for Reliable and Efficient Wireless Sensor Networks Deployments.

(Jain et al., 2004) S. Jain, K. Fall, & R. Patra, 2004. *Routing in a delay tolerant network*, Volume 34. ACM.

(Kavitha et Altman, 2009) V. Kavitha & E. Altman, 2009. Queuing in space: Design of message ferry routes in static ad hoc networks. Dans les actes de *2009 21st International Teletraffic Congress*, 1–8.

(K.F. et V., 2010) D. K.F. & S. V., 2010. *Survey: Matheuristics for Rich Vehicle Routing Problems*, Volume 6373 de *Lecture Notes in Computer Science*, Chapter Hybrid Metaheuristics, 206–221. Springer, Berlin, Heidelberg.

(Mainwaring et al., 2002) A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, & J. Anderson, 2002. Wireless sensor networks for habitat monitoring. Dans les actes de *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, 88–97.

(Malowidzki et al., 2016) M. Malowidzki, T. Dalecki, P. Bereziński, M. Mazur, & P. Skarźyński, 2016. Adapting standard tactical applications for a military disruption-tolerant network. Dans les actes de *2016 International Conference on Military Communications and Information Systems (ICMCIS)*, 1–5.

(Moghadam et al., 2011) K. Moghadam, G. Badawy, T. Todd, D. Zhao, & J. Díaz, 2011. Opportunistic vehicular ferrying for energy efficient wireless mesh networks. Dans les actes de *2011 IEEE Wireless Communications and Networking Conference*, 458–463. IEEE.

(Montoya-Torres et al., 2015) J. Montoya-Torres, J. Franco, S. Isaza, H. Jiménez, & N. Herazo-Padilla, 2015. A literature review on the vehicle routing problem with multiple depots. *Computers & Industrial Engineering 79*, 115–129.

(Płaczek, 2012) B. Płaczek, 2012. Selective data collection in vehicular networks for traffic control applications. *Transportation Research Part C: Emerging Technologies 23*, 14 – 28. Data Management in Vehicular Networks.

(Pentland et al., 2004) A. Pentland, R. Fletcher, & A. Hasson, 2004. Daknet: Rethinking connectivity in developing nations. *Computer 37*(1), 78–83.

(Psaraftis, 1995) H. Psaraftis, 1995. Dynamic vehicle routing: Status and prospects. *Annals of Operations Research 61*(1), 143–164.

(Redi et Ramanathan, 2011) J. Redi & R. Ramanathan, 2011. The darpa wnan network architecture. Dans les actes de *Military Communications Conference, 2011-milcom 2011*, 2258–2263. IEEE.

(Teh et al., 2008) S. Teh, L. Mejias, P. Corke, & W. Hu, 2008. Experiments in integrating autonomous uninhabited aerial vehicles (uavs) and wireless sensor networks. Dans les actes de *Proceedings of the 2008 Australasian Conference on Robotics & Automation*.

(Toth et Vigo, 1997) P. Toth & D. Vigo, 1997. An exact algorithm for the vehicle routing problem with backhauls. *Transportation science 31*(4), 372–385.

(Tse et Viswanath, 2005) D. Tse & P. Viswanath, 2005. *Fundamentals of Wireless Communication*. New York, NY, USA: Cambridge University Press.

(Velásquez-Villada et al., 2014) C. Velásquez-Villada, F. Solano, & Y. Donoso, 2014. Routing optimization for delay tolerant networks in rural applications using a distributed algorithm. *International Journal of Computers Communications & Control 10*(1), 100–111.

(Vieira et al., 2015) R. G. Vieira, A. M. da Cunha, & A. P. de Camargo, 2015. An energy management method of sensor nodes for environmental monitoring in amazonian basin. *Wireless Networks 21*(3), 793–807.

(Wang et al., 2017) K. Wang, Y. Shao, & W. Zhou, 2017. Matheuristic for a two-echelon capacitated vehicle routing problem with environmental considerations in city logistics service. *Transportation Research Part D: Transport and Environment 57*, 262 – 276.

(Winkler et al., 2008) M. Winkler, K.-D. Tuchs, K. Hughes, & G. Barclay, 2008. Theoretical and practical aspects of military wireless sensor networks. *Journal of Telecommunications and Information Technology*, 37–45.

(Wolsey, 1998) L. A. Wolsey, 1998. *Integer programming*. New York, NY, USA: Wiley-Interscience.