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

Luis Flores-Luyo, Agostinho Agra, Rosa Figueiredo, Eladio Ocana

▶ To cite this version:

HAL Id: hal-02338334

https://hal.archives-ouvertes.fr/hal-02338334

Submitted on 29 Oct 2019

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

Luis Flores-Luyo[1,3] · **Agostinho Agra**[2] ·
Rosa Figueiredo[3] · **Eladio Ocaña**[1]

Abstract We consider a wireless network where a given set of stations is continuously generating information. A single vehicle, located at a base station, is available to collect the information via wireless transfer. The wireless transfer vehicle routing problem (WTVRP) is to decide which stations should be visited in the vehicle route, how long shall the vehicle stay in each station, and how much information shall be transferred from the nearby stations to the vehicle during each stay. The goal is to collect the maximum amount of information during a time period after which the vehicle returns to the base station. The WTVRP is NP-hard. Although it can be solved to optimality for small size instances, one needs to rely on good heuristic schemes to obtain good solutions for large size instances. In this work, we consider a mathematical formulation based on the vehicle visits. Several heuristics strategies are proposed, most of them based on the mathematical model. These strategies include constructive and improvement heuristics. Computational experiments show that a strategy that combines a combinatorial greedy heuristic to design a initial vehicle route, improved by a fix-and-optimize heuristic to provide a local optimum, followed by an exchange heuristic, affords good solutions within reasonable amount of running time.

Keywords Vehicle Routing Problem · Wireless Networks · Matheuristic

1
Instituto de Matemática y Ciencias Afines, Lima, Peru
E-mail: {lflores,eocana}@imca.edu.pe

2
CIDMA, Universidade de Aveiro, Aveiro, Portugal
E-mail: aagra@ua.pt

3
Laboratoire Informatique d'Avignon, Avignon Université, Avignon, France
E-mail: {rosa.figueiredo}@univ-avignon.fr

1 Introduction

The vehicle routing problem (VRP) is one of the most studied problems in operations research (OR) with different variants defined and treated in the past five decades (see [16, 24]). The emergence of new computer network architectures add new features to well studied combinatorial optimization problems (e.g. scheduling problems [23], resource allocation problems [29, 22]) and, as we could expect, also to VRPs [7, 20, 26]. The present work studies a VRP appearing in wireless networks applications.

Wireless Networks (WN) have recently received great attention from the 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 [13]. Many applications defined on WN need to provide vehicle routing strategies with wireless information transmission to the vehicles involved (see [11]). In these applications, innovation and research appears most in the development of routing protocols [5, 6, 19, 26], while a smaller set of works addresses the development of vehicle routing strategies, essentially in a two-phase manner [12, 14, 17, 21]. To the better of our knowledge, the works presented in [4, 10, 15, 9] are the only ones to focus on the simultaneous design of a vehicle route and a wireless transmission planning. Notice that the authors in [15] assume a fixed architecture for the vehicle routing (cycle path or zig-zag path).

In the context considered in this work, a base station is connected with the outside while a vehicle is in charge of collecting information from the other stations. The stations are supplied with technology capable of sending information via wireless connection to the vehicle whenever it is located in a sufficiently close station. 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). Simultaneous transmissions are allowed. Information to be sent outside the network is continuously generated, at a constant rate, in each station. The Wireless Transmission VRP (WTVRP) treated in this work looks for the vehicle route as well as for an efficient planning on how to gather information from stations, in order to minimize the total amount of remaining information in the network. Applications for the WTVRP appear in situations where there is a need for providing connection for difficult environments [4, 18, 20, 27].

The authors in [4] were the first ones to propose a solution method for a VRP 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. Information generated in a given underwater node at a time point t_1 which arrives to a surface node at a time point t_2 has a value proportional to $t_2 - t_1$. The strategy adopted in [4] is the maximization of the total 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 very close variant of the problem studied in the present work is treated in [9], which differs from the WTVRP 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 [9], three different objective functions are discussed and experiments are used to investigate how one strategy, i.e., the optimization of one objective function, affects the others and impacts the periodicity of the remaining information. In [10], 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 Mixed ILP (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, in which 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.

The development of commercial solvers and the increasing processing capacity of computers are making possible to optimally solve larger size MILP models. But sooner or later, the instances of interest became too large or too hard to be optimally solved and new approaches based on matheuristics have to come into play in order to treat the problem. For surveys on matheuristics we refer the reader to [2, 3, 8]. For works on matheuristics applied to complex routing problems see for instance [1, 28]. In the present work, we use the vehicle event model presented in [10] and propose several matheuristic based approaches with the aim of solving large instances of the WTVRP.

Our contributions are summarized next. We introduce three heuristics for the construction of an initial solution to the WTVRP. Two of them are matheuristics based on the computational efficient mathematical model introduced in [10], and one is a greedy heuristic. 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. Taking into account the specificities of these constructive heuristics, two improvement heuristics are discussed. 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 proposed.

The paper is organized as follows. Section 2 states formally the WTVRP while notations and assumptions are presented. In Section 3, we describe the vehicle event model presented in [10] for the WTVRP. Then, in Section 4 the constructive, the improvement heuristics and the strategies combining different types of heuristics are presented. In Section 5, we describe the computational experiments carried out to compare the heuristic strategies while final conclusions are given in Section 6.

2 Problem description and notation

The WN is modeled by a directed graph $D = (V, A)$ with the node set $V = \{1, \ldots, n\}$ representing the n stations of the network and the arc set A representing 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). The terms node and station will be used indistinguishably throughout the text. Let $T = \{1, 2, \ldots, m\}$ be the time horizon considered. At the beginning of the time horizon, each station $j \in V \setminus \{1\}$ contains an amount Q_j of data. For each station $i \in V \setminus \{1\}$, data 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 by radio),

$$q_{jk} \geq (k - t_{last})r_j,$$

where t_{last} is the time of the last extraction. If node j has not been visited before time period k, then $q_{jk} = Q_j + kr_j$.

For an illustration of the WTVRP described in this section, see Figure 1 from [10]. Only the base station is properly equipped to send information outside the network. An 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. Data 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 $(i, j) \in A$. Wireless transmission can be used to transfer data from a station $j \in V$ to the vehicle located in a station $i \in V \setminus \{j\}$.

Wireless transmission is only possible for close enough stations. Let r_{cov} be a maximum distance allowing wireless transmission. A station j can wireless transfer its data to (the vehicle in) station i whenever $d_{ij} \leq r_{cov}$. We define the set of nodes that can send information to node i as $range(i) = \{j \in V : d_{ji} \leq r_{cov}\}$. In Figure 1, the vehicle is located at station 5 receiving information from all stations in $range(5) = \{4, 5, 7\}$.
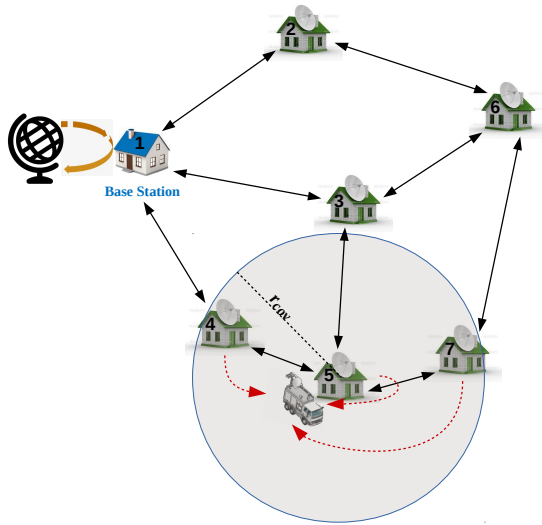
Fig. 1 The wireless transfer vehicle routing problem with 7 stations [10].

Following the same assumptions made in [10, 9], the transmission speed is inversely proportional to the square of the distance between the stations involved and it depends on two additional factors. First, on the amount of information transmitted; second, on physical factors as the equipments used and obstacles among stations. The transmission occurs with a fixed transmission power of P_t and the received power P_r is given by

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

where D is the distance between the receiver and transmitter and η is the pathloss parameter which we shall take in this paper to be equal to two. We assume that the vehicle has an antenna with an elevation of one unit. The coordinates of a sensor are given by $(x_s, y_s, 0)$ and those of the vehicle are $(x_b, y_b, 1)$, e.g., 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 [25] and write it as

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

Let β be the antenna's gain and σ be the noise at the receiver (assuming independent channels and as a consequence no interferences of other transmissions on the received signal from a sensor).

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

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

implying the maximum amount of information per time unit that can be sent from node j to node i is at most

$$\frac{1}{\alpha_{ji}(1 + d_{ji}^2)},$$

with parameter $\alpha_{ji} = \frac{\beta P_t}{2\sigma}$ representing the physical limitations of sending information between stations i and j.

Any station is free to transfer only part of its information to the vehicle and simultaneous transmissions are possible. Parameter M denotes the maximum number of nodes that can transfer information simultaneously to the vehicle in each time period while R denotes the maximum amount of information that can be transferred in each time period.

The Wireless Transmission Vehicle Routing Problem (WTVRP) [10], 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 station) together with an efficient planning for collecting information from stations $V \setminus \{1\}$. The criteria for measuring the efficiency of a collect planning is the total amount of information collected.

3 The vehicle event model

The authors in [10] introduced and compared three different MILP models for the exact solution of the WTVRP: a discrete time model and two event based models. Computational experiments on small and medium-sized instances showed, typically, an optimal vehicle route includes only a small number of nodes. In this section, we describe the vehicle event model introduced in [10] in which an event is defined as a vehicle stop.

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 routing variables indicate the node visited at the kth vehicle stop, indexed by the event $k \in N$.

For each $i \in V$, $k \in N$, a binary variable is defined as follows.

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

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

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

$$\gamma_k : \text{time (in time periods) spent at the } k^{th} \text{ event.}$$

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

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

Finally, 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 Model introduced in [10] is as follows.

$$Minimize \left\{ \sum_{i \in V} (Q_i + mr_i) - \sum_{i \in V, j \in range(i), k \in N} f_{jik} \right\} \qquad (2)$$

$$\sum_{j:(1,j) \in A} x_{j1} = 1, \qquad (3)$$

$$\sum_{k \in N} x_{1k} = 1, \qquad (4)$$

$$\sum_{j \in V} x_{jk} \leq 1, \qquad \forall k \in N \setminus \{1\}, \qquad (5)$$

$$x_{jk} \leq \sum_{i:(i,j) \in A} x_{i,k-1}, \qquad \forall j \in V, k \in N, \qquad (6)$$

$$x_{jk} \leq 1 - \sum_{\ell=1}^{k-1} x_{1\ell}, \qquad \forall j \in V \setminus \{1\}, k \in N, \qquad (7)$$

$$t_k \geq t_{k-1} + \gamma_{k-1} + t_{ij}(x_{i,k-1} + x_{jk} - 1), \qquad \forall (i,j) \in A, k \in N, \qquad (8)$$

$$t_1 \geq \sum_{j:(1,j) \in A} t_{1j} x_{1j}, \qquad (9)$$

$$t_k \leq m, \qquad \forall k \in N, \qquad (10)$$

$$q_{jk} = Q_j + r_j t_k - \sum_{\ell=1}^{k-1} \sum_{i \in range(j)} f_{ji\ell}, \qquad \forall j \in V, \, k \in N, \qquad (11)$$

$$f_{jik} \leq q_{jk} + r_j \xi_{jik}, \qquad \forall j \in V, \, i \in \text{range}(j), \, k \in N, \qquad (12)$$

$$f_{jik} \leq \frac{\xi_{jik}}{\alpha_{ji}(1 + d_{ji}^2)}, \qquad \forall j \in V, \, i \in \text{range}(j), \, k \in N, \qquad (13)$$

$$\sum_{j \in range(i)} f_{jik} \leq R\gamma_k, \qquad \forall i \in V, \, k \in N, \qquad (14)$$

$$\sum_{i \in range(j)} \xi_{jik} \leq \gamma_k, \qquad \forall j \in V, \, k \in N, \qquad (15)$$

$$\sum_{j \in V, i \in range(j)} \xi_{jik} \leq M\gamma_k, \qquad \forall k \in N, \qquad (16)$$

$$\xi_{jik} \leq m x_{ik}, \qquad \forall i \in V, \, j \in range(i), \, k \in N, \qquad (17)$$

$$f_{jik} \geq 0, \qquad \forall i \in V, \, j \in range(i), \, k \in N, \qquad (18)$$

$$q_{jk} \geq 0, \qquad \forall j \in V, \, k \in N, \qquad (19)$$

$$t_k \in \mathbb{Z}^+ \qquad \forall k \in N, \qquad (20)$$

$$\gamma_k \in \mathbb{Z}^+, \qquad \forall k \in N, \qquad (21)$$

$$\xi_{jik} \in \mathbb{Z}^+, \qquad \forall i \in V, \, j \in range(i), \, k \in N, \qquad (22)$$

$$x_{ik} \in \{0,1\}, \qquad \forall i \in V, \, k \in N. \qquad (23)$$

Constraints (3)–(7) are the Routing Constraints. Equations (3) (respectively (4)) ensure the vehicle leaves (respectively, ends) its route at the base node. Inequalities (5) ensure at most one visit labeled k is made. Inequalities (6) state that, if the kth visit is made to node j, then the $k-1^{th}$ visit occurred in a predecessors of node j. Constraints (7) ensure that each routing variable is equal to zero once the vehicle has returned to the base node.

Time Constraints are Constrains (8)–(10). Inequalities (8) ensure the start time of the kth visit takes into account the start time of the previous visit, the time spent on the last visit as well as the traveling time between the two stations visited. Constraints (9) restrict the start time of the first visit while constraints (10) impose all the visits to start during the time horizon.

Constraints (11)–(17) are the Information Transfer Constraints. Constraints (11) set the amount of information at each node at the beginning of each visit. Constraints (12) ensure that the amount that can be transferred cannot exceed the information available at the station. Constraints (13) (respectively

(14)) limit the transfer amount taking into account the transfer rate (respectively, the maximum transfer quantity per period). Constraints (15) ensure that, when visiting a node i, the time used to transfer information from each node j to node i cannot exceed the total time of the visit. Constraints (16) 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. Transfer variables are linked to routing variables by Constraints (17), ensuring that a node j can transfer information to a node i during the kth visit if the kth visit occurred at node i. Constraints (18)–(23) define the variables domain. Finally, the objective function (2) is in charge of minimizing the amount of information kept in the nodes at the end of the time horizon.

Clearly, the size of the vehicle event model depends on the size of N, i.e., on the value of $\hat{N}$. Let N^* be the maximum vehicle visits. The value N^*, is difficult to estimate. From one hand, overestimating N^*, i.e., choosing a value $\hat{N} > N^*$, ensures us to obtain the optimal solution, but the model becomes large and it may not be solved within reasonable running times. On the other hand, by underestimating N^*, i.e., choosing a value $\hat{N} < N^*$, the model becomes smaller, thus easier to solve, and by solving it a feasible solution is obtained. However, this feasible solution may not be optimal and the solution cost may increase as $\hat{N}$ decreases.

As we have already mentioned, two other MILP formulations were presented in [10] to the WTVRP: a discrete time model and another event model. The authors showed that the presented vehicle model outperforms the other two models for longer time horizons. The three MILP formulations model the times related to the node visits and transfer operations differently. From the experiments described in [10], the authors concluded that depending on the modeling strategy adopted, we can obtain different optimal solutions w.r.t. to transfer operations. Although, the impact of this choice in the optimal solutions obtained for random instances was small. For a deep analysis of the three models and a discussion on problem assumptions, we refer the reader to [10].

4 Heuristics

When the number of stations is large, overestimating N^* leads to huge size MILP models that, in general, cannot be solved within reasonable running time. In this section, we discuss several heuristic strategies combining different types of heuristics that we will classify into constructive (Section 4.1) and improvement heuristics (Section 4.2). As we will see, most of them are matheuristics based on the MILP vehicle event model described in Section 3.

4.1 Constructive heuristics

Here we describe three constructive heuristics designed to derive good initial feasible solutions: a simple combinatorial relaxation heuristic (Section 4.1.1), a fix-and-relax heuristic (Section 4.1.2); and a greedy heuristic (Section 4.1.3).

4.1.1 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 (see [10]). 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.

4.1.2 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 (5) 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.

Algorithm 1 Fix-and-relax

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

4.1.3 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) 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 (13) and multi-transfer constraints (14) 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:

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

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 $m - k$. 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 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.

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.
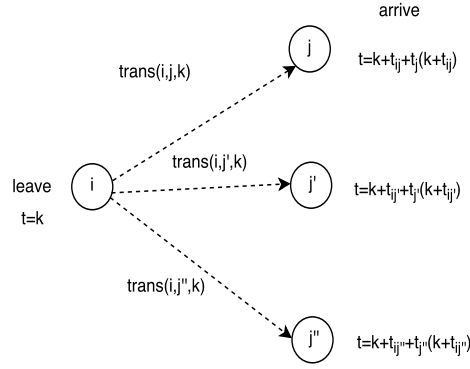
Fig. 2 Choice of the neighbor station according to criterion of the greatest transfer.

Algorithm 2 Greedy algorithm

 1: Set $i \leftarrow 1$
 2: Set $k \leftarrow 1$
 3: Set STOP $\leftarrow$ false
 4: **repeat**
 5:     Let $j^* \leftarrow argmax\{transf(i,j,k)|j \in range(i)\}$;
 6:     **if** $t_{j^*}(k + t_{ij^*}) > 0$ **then**
 7:         $i \leftarrow j^*$
 8:         $k \leftarrow k + t_{ij^*} + t_{j^*}(k + t_{ij^*})$
 9:     **else**
10:         STOP $\leftarrow$ true
11:     **end if**
12: **until** STOP = true

4.2 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 4.2.1); a best insertion heuristic (Section 4.2.2); and an exchange heuristic (Section 4.2.3).

4.2.1 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 5).

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.

4.2.2 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.
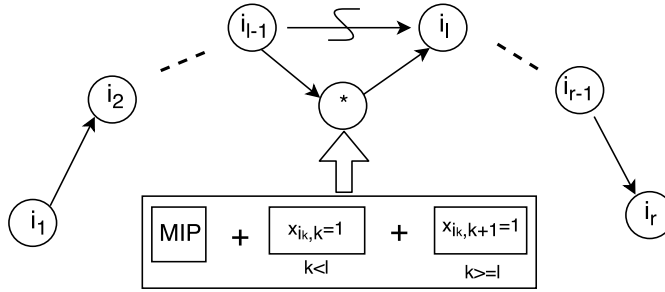
Fig. 3 Insertion of the node j in the lth position of the route.

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 lth 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.

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.

4.2.3 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.

Algorithm 3 Best insertion heuristic

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

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 4 for an example with $l = 2$). As the initial route R is a feasible route for the restricted MILP model, then $\bar{z} \leq z$.
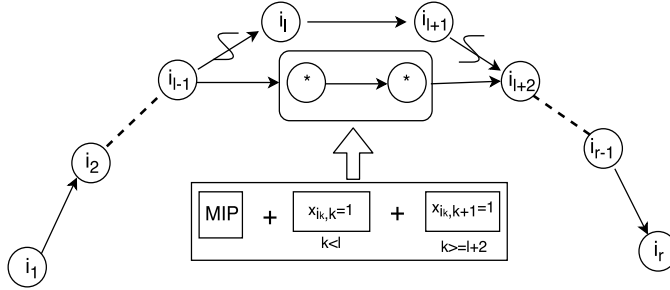

Fig. 4 Heuristic Exchange(l,2), i.e., the exchange of two neighboring nodes by nodes j and j' starting at position l.

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.

Algorithm 4 Exchange heuristic

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

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 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.

4.3 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 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 5.

Constructive heuristics	Improvement heuristics	
N-MILP	Best insertion	Exchange
Fix-and-relax		
Greedy	Fix-and-optimize	

Fig. 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.

5 Computational experiments

In this section, we report the computational tests conducted to evaluate the heuristics and compare the several strategies introduced in Section 4, which combine constructive and improvement heuristics. First, Subsection 5.1 presents computational results obtained on a set of random instances, indicating which heuristic strategy is appropriate for an instance of a given size. Then, Subsection 5.2 presents computational results obtained on a set of realist instances described in [4].

All the experiments presented here were performed using a server with 15 CPU's Intel ®Xeon (R) E5540@ 2.53Ghz X4, with 16 GB of RAM. To solve the MILP models, the IBM CPLEX Optimizer 12.6.1.0 solver was used.

5.1 Random instances

The set of random instances was generated as described in [10, 9]. 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 $m \in \{72, 120, 240\}$ 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 $M = 3$.

5.1.1 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 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).

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 2 reports the results obtained with the greedy heuristic. The first two columns repeat information given in the corresponding column of Table 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 1 Computational results for the MILP model with a run time limit of one hour and the N-MILP heuristic.

Parameter	MILP			N-MILP heuristic			
	$\bar{z}$	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
$m=120$	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
$m=120$	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
$m=200$	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 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, 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.

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

Table 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

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 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).

Again, the gaps show that for the easiest instances (with 20 nodes), the performance of the two heuristic strategies tested (N-MILP combined with

Table 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

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 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

Table 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

model. Between the two tested approaches, none of the approaches is clear better than the other.

5.1.2 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 6-8 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 4.2.1, 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

Table 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	16737,00	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	17937,40	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

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 9-11 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
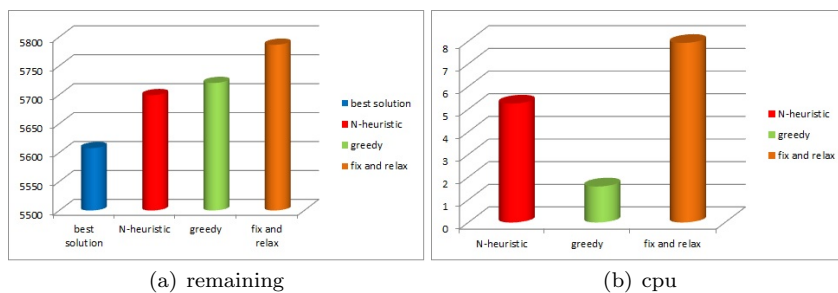
(a) remaining

(b) cpu

Fig. 6 Comparison of constructive heuristic approaches on instances with $|V| = 20, m = 120$.

(a) remaining

(b) cpu
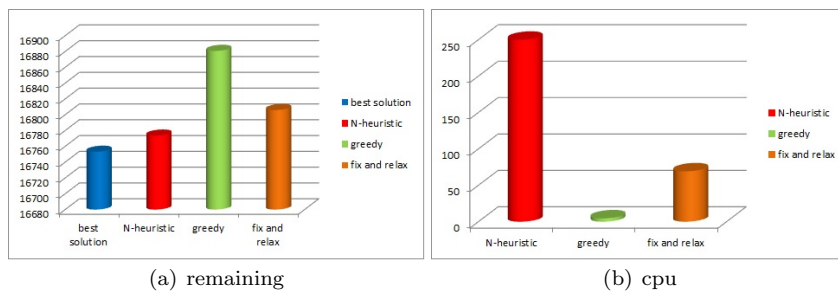
Fig. 7 Comparison of constructive heuristic approaches on instances with $|V| = 50, m = 120$.
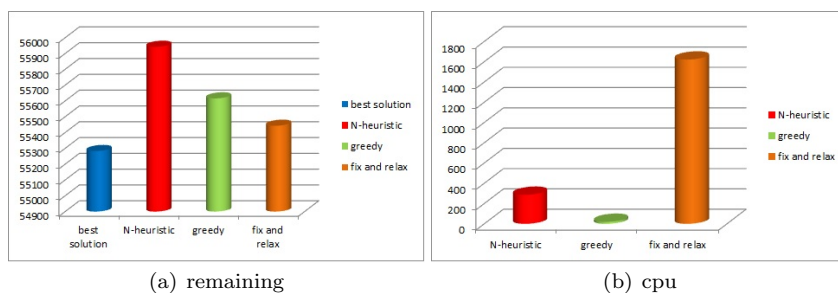

(a) remaining

(b) cpu

Fig. 8 Comparison of constructive heuristic approaches on instances with $|V| = 100, m = 200$.

by the best insertion heuristic and combined with the exchange heuristic as (N+Ins+Exch).

We can observe that, for the easiest instances, with $|V| = 20$, the MILP model provides the best solutions although all the heuristic approaches are quite fast. However, when $|V|$ increases, the quality of the MILP solution decreases and the heuristic strategies become more competitive. For $|V| = 100$, all the heuristic strategies provide better solutions than the MILP model within a time limit of one hour. This contrasts with the case where only constructive heuristics are considered, showing that combining constructive with
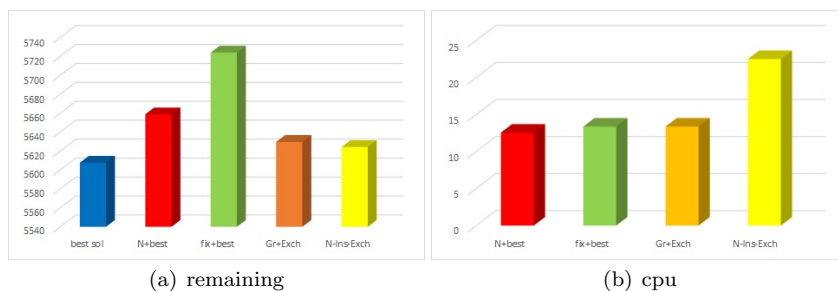
(a) remaining   (b) cpu

Fig. 9 Comparison of heuristic strategies on instances with $| V |= 20, m = 120$.

(a) remaining   (b) cpu
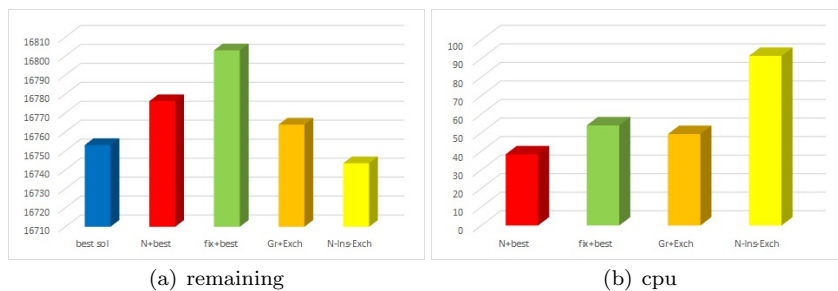
Fig. 10 Comparison of heuristic strategies on instances with $| V |= 50, m = 120$.
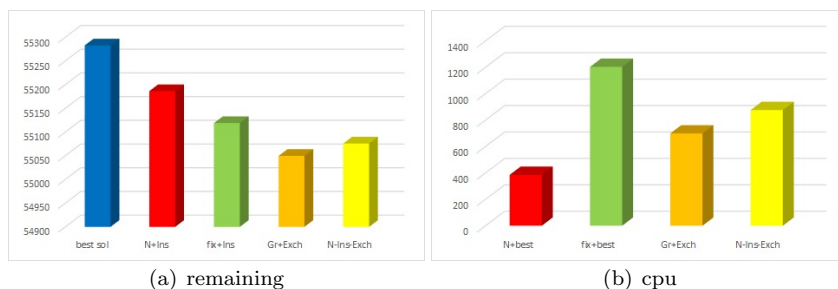

(a) remaining   (b) cpu

Fig. 11 Comparison of heuristic strategies on instances with $| V |= 100, m = 200$.

improvement heuristics is essential to solve the problem. Also, the exchange
heuristic allows to improve the solutions obtained with the greedy heuristic
combined with the fix-and-optimize and the N-MILP heuristic combined with
the best insertion, but this improvement has a significant impact on the in-
crease of the running time. Among all the tested heuristic strategies, the greedy
heuristic combined with the fix-and-optimize and the exchange heuristic pro-
vides the best solutions for the largest instances, while the MILP model and the
N-MILP heuristic combined with the best insertion provide better results for
smaller instances. This result is expected since the the MILP model heuristic
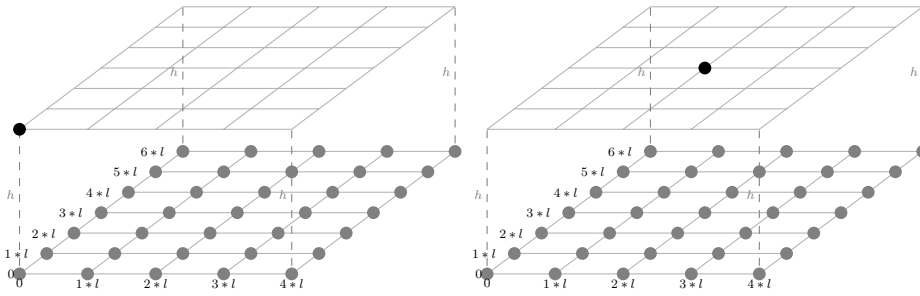and the N-MILP heuristic are based on the mathematical model which tends

Fig. 12 A grid topology inspired from [4] with 35 underwater stations and a unique surface station located either (rigth) in a extreme grid node or (left) in the central grid node.

to provide good quality solutions for the easiest instances and perform poorly on hard instances.

5.2 Realistic instances

As we have mentioned in the introduction, Basagni *et al.* in [4] described a VRP appearing in underwater wireless sensor networks for submarine surveillance. This problem is related to the WTVRP considered in this work with three major differences. First, the authors in [4] assume that stations must be physically visited for information transmission. Second, multiple stations on the surface connect the network with the outside. Third, the objective function is the maximization of the value of the information transmitted. Although the two problems are not the same, the solution of the WTVRP can contribute in the context described in [4] whenever equipments allow wireless transmission. The grid topology and technological assumptions described in [4] are used here to generate a set of realistic instances for the WTVRP.

The largest topology described in [4] is depicted in Figure 12. We consider $5 \times 7 = 35$ nodes uniformly deployed on a rectangle of $4l \times 6l$ meters taking $l = 500$ (gray nodes in Figure 12). The Cartesian coordinates (x, y) of each station are done by the position of each gray node in the grid while the z coordinates are randomly generated from 100 to 300 meters. The vehicle speed is set as 1.8 meters per second. Assuming a video encoded using the standard H.264 codec, each five minutes recording produces 9MB of information to be transfered. The transfer speed is assumed to be equal to 10Mbps which implies $\alpha_{ii} = 375$ for each station i. Acoustic channel data rate is set to 10Kbps which implies $\alpha_{ij} = 0.375$ for each pair of stations $i \neq j$. We assume a time period of 12 hours with the time unit defined as 5 minutes, which implies $m = 144$. Since the transfer equipments have a maximum speed of 100Mbps, i.e. 3750MB every 10 minutes, we define $R = 3750$. Also, we assume $M = 5$, i.e., the existence of 5 channels for simultaneous transmissions. Finally, we define the maximum range for a wireless transfer $r_{cov} = 800$ meters.

Table 6 Computational results with the N-MILP heuristic combined with the best insertion and the MILP model on the set of realistic instances.

Instances	d	MILP ($\hat{N} = 20$)		MILP ($\hat{N} = 25$)		Greedy			N-MIP+best insertion		
		$\bar{z}$	DGap	$\bar{z}$	DGap	$\bar{z}$	Length	Cpu	$\bar{z}$	Length	Cpu
A.1	0,81	32343	33,10	32496	99,99	30221	31	0,40	27339	26	886,85
A.2		33960	100,24	39363	100,20	31206	29	0,13	26325	27	1015,76
A.3		31198	30,95	32571	100,25	29295	32	0,10	26160	28	1188,49
A.4		32835	34,83	31342	100,25	29878	31	0,13	25803	28	1058,74
A.5		32957	34,33	35772	100,23	29386	32	0,16	25995	28	1012,29
B.1	0,81	32632	33,54	31846	99,91	30327	31	0,16	27064	27	782,43
B.2		33778	100,08	31332	100,00	30082	31	0,11	27318	26	797,08
B.3		33209	34,93	31883	100,23	30430	31	0,10	26813	27	973,58
B.4		34034	36,49	36689	100,22	29045	32	0,15	26165	28	1409,04
B.5		32111	32,69	32792	100,25	29325	32	0,10	25284	29	1420,10
A.1	0,49	30591	99,99	30110	100,00	30638	30	3,59	17552	35	3560,00
A.2		31354	97,23	28140	100,00	31198	30	0,13	17585	35	3105,80
A.3		31363	100,22	28304	100,00	31818	30	0,12	17437	35	2751,83
A.4		31524	100,00	28163	100,05	30083	32	0,12	17536	35	2709,26
A.5		30011	100,14	28045	100,16	30320	31	0,12	17510	35	3117,12
B.1	0,49	30390	28,62	29044	99,99	30197	31	0,12	17402	35	2770,80
B.2		30436	28,89	27229	100,11	31325	30	0,11	18349	34	2461,94
B.3		29272	57,84	30223	100,00	31075	30	0,11	18088	34	2420,64
B.4		30202	28,64	28601	96,91	31530	28	0,11	18899	33	2561,15
B.5		30695	93,92	27727	94,47	30816	30	0,11	17661	35	3018,83

*** *time limit: 3600 sec.*

Using these assumptions, five different complete graphs were generated with 35 underwater nodes. In order to simulate the existence of physical underwater obstacles, a random set of edges was deleted from each complete graph generating two sets of graphs: one set with densities equal to 0.81 and another with densities equal to 0.49.

For each different graph obtained, we consider two scenarios for the position of the unique base station located at the surface (black points in Figure 12) connected to each underwater station. The Cartesian coordinates (x, y) of the surface station is defined as $(0, 0)$ (right grid in Figure 12; instances A.i in Table 6) or $(2l, 3l)$ (left grid in Figure 12; instances B.i in Table 6). That implies a total of 20 realistic instances.

From the conclusions presented in the previous section, the N-MILP heuristic combined with the best insertion is the most appropriate strategy to solve the set of realistic instances. In order to obtain an estimative for the value of $\hat{N}$ to be used in the MILP model, we also run the greedy heuristic on this set of instances. Since, the value obtained for $\hat{N}$ was prohibitive for the execution of the MILP model, we run the MILP model with $\hat{N} = \{20, 25\}$. Table 6 presents the results obtained by the heuristic strategies as well as by the event MILP model. Notations on this table are the same used in the previous ones. Additionally, column "d" informs the density of the graphs while column "Length" the number of events (vehicle visits) in the heuristic solutions obtained.

From Table 6, we conclude there is no significant difference between instances in set A and B which is explained by the fact that the vehicle visits the surface station only at the beginning and at end of its route. The only dif-

ference was for the MILP model with $\hat{N} \geq 20$, for which a better performance was obtained by this method. We can also observe that a value $\hat{N} \geq 25$ seems necessary but it is prohibitive for the MILP model on the realistic instances. For realistic instances with density $d = 0.81$, the quality of the solutions obtained by the Greedy heuristic were better than the ones obtained by the MILP model ($\hat{N} \in \{20, 25\}$). Finally, as we expected, the heuristic strategy N-MILP followed by best insertion achieved lower cost solutions for all the realistic instances. Interestingly, compared with the greedy heuristic, the better solutions obtained by "N-MILP+best insertion" considered longer paths when $d = 0.49$ but shorter when $d = 0.81$.

6 Conclusions

In this work, we considered a wireless transfer vehicle routing problem. Since most of the practical instances cannot be solved to optimality within a reasonable amount of computational time (see [9, 10]), we proposed several heuristic approaches that combine both constructive and improvement heuristics. In order to derive initial feasible solutions, three constructive heuristics were proposed. Two of them use the MILP model and the other is a greedy heuristic. Three improvement heuristics were 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 was used to improve the initial solution obtained with the greedy algorithm since this algorithm does not 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 were fast for small values of $\hat{N}$. Thus an improvement heuristic that starts from an initial route with a small number of vehicle visits and iteratively tests the inclusion of another visit was proposed. Finally, an exchange heuristic that exchanges a consecutive set of nodes by new ones was also presented. This heuristic was combined with all the constructive heuristics introduced in the present work.

Computational tests showed 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. Moreover, for a set of realistic instances occurring in submarine surveillance, a combined heuristic strategy based on a restricted MILP model followed by a best insertion procedure provided the best computational results.

Acknowledgements

References

1. A. Agra, M. Christiansen, A. Delgado, and L. Simonetti. Hybrid heuristics for a short sea inventory routing problem. *European Journal of Operational Research*, 236(3):924–935, 2014.
2. C. Archetti and M. G. Speranza. A survey on matheuristics for routing problems. *EURO Journal on Computational Optimization*, 2(4):223–246, 2014.
3. M. O. Ball. Heuristics based on mathematical programming. *Surveys in Operations Research and Management Science*, 16(1):21 – 38, 2011.
4. S. Basagni, L. Bölöni, P. Gjanci, C. Petrioli, C.A. Phillips, and D. Turgut. Maximizing the value of sensed information in underwater wireless sensor networks via an autonomous underwater vehicle. In *Proceedings of IEEE INFOCOM'14*, pages 988–996, April-May 2014.
5. Sourav Kumar Bhoi, Pabitra Mohan Khilar, and Munesh Singh. A path selection based routing protocol for urban vehicular ad hoc network (uvan) environment. *Wireless Networks*, 23(2):311–322, Feb 2017.
6. G.D. Celik and E. Modiano. Dynamic vehicle routing for data gathering in wireless networks. In *49th IEEE Conference on Decision and Control (CDC)*, pages 2372–2377. IEEE, 2010.
7. K. Collins and G. M. Muntean. An adaptive vehicle route management solution enabled by wireless vehicular networks. In *2008 IEEE 68th Vehicular Technology Conference*, pages 1–5, Sept 2008.
8. K. F. Doerner and V. Schmid. *Survey: Matheuristics for Rich Vehicle Routing Problems*, volume 6373 of *Lecture Notes in Computer Science*, chapter Hybrid Metaheuristics, pages 206–221. Springer, Berlin, Heidelberg, 2010.
9. L. Flores-Luyo, A. Agra, R. Figueiredo, E. Altman, and E. Ocaña Anaya. Vehicle routing problem for information collection in wireless networks. In *Proceedings of the 8th International Conference on Operations Research and Enterprise Systems, ICORES 2019, Prague, Czech Republic, February 19-21, 2019.*, pages 157–168, 2019.
10. L. Flores-Luyo, A. Agra, R. Figueiredo, and E. Ocaña. Mixed integer formulations for a routing problem with information collection in wire-

less networks. *European Journal of Operational Research*, 280(2):621–638, 2020.

11. M. Di Francesco, S.K. Das, and G. Anastasi. Data collection in wireless sensor networks with mobile elements: A survey. *ACM Transactions on Sensor Networks*, 8(1):1–31, 2011.

12. S. R. Gandham, M. Dawande, R. Prakash, and S. Venkatesan. Energy efficient schemes for wireless sensor networks with multiple mobile base stations. In *GLOBECOM '03. IEEE Global Telecommunications Conference*, volume 1, pages 377–381, 2003.

13. J. A. Gomez-Pulido and J. M. Lanza-Gutierrez, editors. *Journal of Heuristics*, chapter Heuristics for Reliable and Efficient Wireless Sensor Networks Deployments. 2015.

14. L. Gu and A.J. Stankovic. Radio-triggered wake-up for wireless sensor networks. *Real-Time Systems*, 29(2-3):157–182, 2005.

15. V. Kavitha and E. Altman. Queuing in space: Design of message ferry routes in static ad hoc networks. In *2009 21st International Teletraffic Congress*, pages 1–8, 2009.

16. G. Laporte. Fifty years of vehicle routing. *Transportation Science*, 43(4):408–416, 2009.

17. J. Luo and J.-P. Hubaux. Joint mobility and routing for lifetime elongation in wireless sensor networks. In *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, volume 3, pages 1735–1746, 2005.

18. M. Malowidzki, T. Dalecki, P. Bereziński, M. Mazur, and P. Skarżyński. Adapting standard tactical applications for a military disruption-tolerant network. In *2016 International Conference on Military Communications and Information Systems (ICMCIS)*, pages 1–5, May 2016.

19. K.R. Moghadam, G.H. Badawy, T.D. Todd, D. Zhao, and J.A.P. Díaz. Opportunistic vehicular ferrying for energy efficient wireless mesh networks. In *2011 IEEE Wireless Communications and Networking Conference*, pages 458–463. IEEE, 2011.

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

21. J. Rao, T. Wu, and S. Biswas. Network-assisted sink navigation protocols for data harvesting in sensor networks. In *2008 IEEE Wireless Communications and Networking Conference*, pages 2887–2892, 2008.

22. S. R. Shishira, A. Kandasamy, and K. Chandrasekaran. Survey on meta heuristic optimization techniques in cloud computing. In *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 1434–1440, Sept 2016.

23. L. Teylo, U. de Paula Junior, Y. Frota, D. de Oliveira, and L. Drummond. A hybrid evolutionary algorithm for task scheduling and data assignment of data-intensive scientific workflows on clouds. *Future Generation Comp. Syst.*, 76:1–17, 2017.

24. P. Toth and D. Vigo. *Vehicle Routing: Problems, Methods, and Applications, Second Edition*. Society for Industrial and Applied Mathematics,

Philadelphia, PA, USA, 2014.

25. D. Tse and P. Viswanath. *Fundamentals of Wireless Communication*. Cambridge University Press, New York, NY, USA, 2005.

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

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

28. K. Wang, Y. Shao, and W. Zhou. 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, 2017.

29. Z.-H. Zhan, X.-F. Liu, Y.-J. Gong, J. Zhang, H. S.-H Chung, and Y. Li. Cloud computing resource scheduling and a survey of its evolutionary approaches. *ACM Comput. Surv.*, 47(4):63:1–63:33, July 2015.