Defining Routes for Emergency Response from Climate Events: a Data-oriented Approach

Luis Felipe Oliveira, Yuri Frota, and Daniel de Oliveira

***Abstract*—Extreme Events have recently become a topic of interest mainly due to the impacts of climate change worldwide over the last years. Such events commonly have a severe impact on major cities. Especially in Brazilian cities, the combination of climate change and unplanned urban growth on steep slopes has led to floods and landslides that killed thousands of people over the past decade. Once an extreme event impacts a city, a series of emergency events (henceforth named e-events) are registered by the citizens (by calling the local emergency telephone number). These e-events may vary from a small flood to a large-scale landslide. Thus, it is a top priority for governments to respond to these e-events as fast and as efficiently as possible. The problem is that the number of e-events is usually larger than the number of people (and vehicles) to respond to them. Another challenge is that these vehicles have to cross an impacted city with major traffic jams. The routes have to consider real-time traffic data to avoid streets with heavy traffic. This article proposes an approach named `IRONSTONE`, which aims at collecting the topology of the city and real-time traffic data to generate optimized routes for teams to respond to e-events. The proposed approach was evaluated with real scenarios of the city of Niterói and the results are promising.**

***Index Terms*—Emergency Events, Emergency Response, e-events, Routing, Optimization, Smart Cities.**

I. INTRODUCTION

The emergence of the *Smart City* concept has driven the development of many initiatives and approaches worldwide over the last decade [1]–[4]. Such initiatives have the potential to improve the lives of citizens in multiple dimensions. One of these dimensions is related to living [5], which aims at proposing solutions to improve the quality of life of the citizens in their daily duties. One task that is crucial in this dimension is how to provide effective and efficient responses to emergencies, especially the ones caused by extreme events in large urban centers [6], [7] (*e.g.*, storms, earthquakes, *etc.*) where the impacts can be catastrophic.

Let us consider as an example the city of Niterói in Brazil (this example will be used consistently throughout this article). Niterói has an estimated population of 515,317 inhabitants and an area of 129.375 km^2, thus becoming the fifth most populous city in Rio de Janeiro metropolitan area. Several important streets and avenues in the city suffer from frequent floodings, such as Marquês do Paraná Avenue. Despite the efforts of the local government to carry out interventions in these places, the water flow capacity is still not sufficient. With the occurrence of storms or intermittent rain, the car traffic on this avenue is blocked and it impacts the traffic throughout the city (since this avenue is the main exit towards the city of Rio de Janeiro). In addition, floods strongly impact people living in risk areas (*e.g.*, slopes, riverbanks, *etc.*), which was the case with the landslide in Morro do Bumba in 2010 [1]

Although the definition of an extreme event encompasses a wide spectrum of phenomena, we focus on this article on climate events. Major climate events produce a plethora of problems (*e.g.*, landslides, floods, *etc.*), which are usually reported by the citizens to the Civil Defense Department. Every time a citizen reports a problem associated with a climate event, an Emergency Event (henceforth called just *e-event*) is registered. Each e-event is prioritized by the experts and inserted in a queue to be responded to. Commonly, after a major climate event, the number of e-events can be huge. For example, on April 10th, 2019, after a thunderstorm, the city of Niterói registered more than 25 e-events in only two hours (from 8 a.m. to 10 a.m.) as presented in Fig. 1 (each brown pin represents an e-event). Thus, how to rapidly respond to problems caused by climate events has become a topic of interest for the research community.

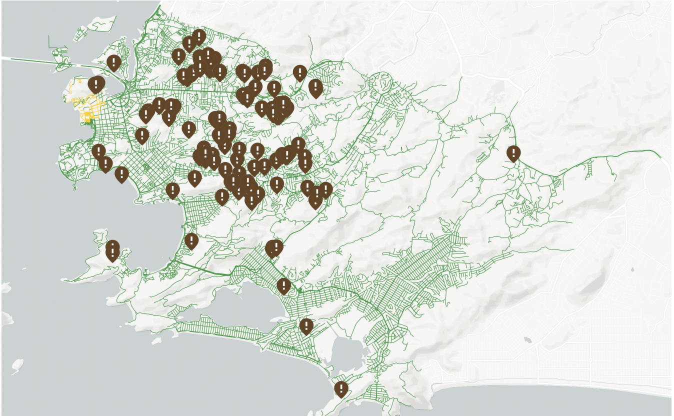


Fig. 1. Registered e-events in 04/10/2019 in the City of Niterói. Each brown pin represents an e-event that occurred and had to be responded to by the Civil Defense Department.

One challenge that arises is that these e-events have to be analyzed *in loco* by a team of experts (*e.g.*, if a house has collapsed, civil engineers must come to the site to check the status and define actions to solve the problem). In the context of the city of Niterói, the Civil Defense Department has only three teams at this moment, each one with a vehicle, to respond to all registered e-events. Thus, one of the most complex tasks

Luis Felipe Oliveira, Yuri Frota and Daniel de Oliveira are with Institute of Computing from Universidade Federal Fluminense, Niterói, Brazil e-mail:lfxaoliveira@id.uff.br, {yuri, danielcmo}@ic.uff.br.

Manuscript received April 19, 2025; revised August 26, 2025.

---

[1] https://shorturl.at/oHKN4 - in Portuguese

to be performed is how to define routes to these vehicles to respond to as many as possible e-events and as fast as possible.

Formally, this problem can be formulated as a multiple traveling salesman problem (mTSP) [8], [9] which is a generalized version of classical traveling salesman problem (TSP), in which a set of vehicles instead of a single one is involved to cover all e-events. More specifically, we are interested in the mTSP variant that includes multiple depots (*i.e.,* each vehicle starts from different locations) and open routes (*i.e.,* vehicles do not need to return to their original location), denoted as MO-mTSP [10], [11], [12]. Moreover, each e-event presents its priority (determined by its severity) that must be respected to achieve a feasible route in the solution. For instance, Fig. 2(a) illustrates a MO-mTSP optimal solution in a scenario defined by two vehicles and six e-events (red circles) where the labels represent the e-events priorities. All elements are arranged on a Cartesian map and by convention, the transfer time between elements is considered equal to the Euclidean distance between them. The routes do not respect the e-events priorities, because the prime goal of MO-mTSP is to only minimize the distance/time traveled by vehicles, which results in a late service to more urgent e-events.

On the other hand, Fig. 2(b) depicts an optimal solution for MO-mTSP with priority constraints, where the main idea is to also prioritize the most urgent e-events to improve their response times. This scenario leads us to a new mTSP variant from here on called MO-mTSP-P (Multiple depots Open Route mTSP with Priority constraints). We refer the reader to [9] for a detailed analysis of known variants and methods of mTSP.

In practice, one of the complexities of the MO-mTSP-P problem is that the existing resources are commonly scarce, *i.e.*, the number of vehicles is considerably smaller than the amount required to respond to all registered e-events. Thus, the director of the Civil Defense Department (who is regularly responsible for defining vehicle routes) must define which specific routes each vehicle will be responsible for to respond to a set of registered e-events. This type of vehicle route definition ends up prioritizing the so-called *Hot Spots*, *i.e.* areas with many e-events. This type of prioritization is already found in many applications, *i.e.*, police patrol [13].

The current solution used to tackle the MO-mTSP-P problem is to manually generate the routes, which may be tedious and error-prone depending on the number of considered e-events. However, the definition of the routes must be as effective and efficient as possible. Previous studies show that a small response time can reduce mortality in cases of medical emergencies, and consequently reduce financial losses [14]. Today, considering the aforementioned challenges, the ideal scenario is to have an approach that automatically defines routes to vehicles to respond to e-events considering real-time traffic data [15].

In this article, we propose an approach named `IRONSTONE` (Data-Oriented Approach for Routing Vehicles To Respond Emergency Events). `IRONSTONE` analyzes real-time traffic data and generates a graph that represents the streets in a radius (*e.g.*, 10 km) from the e-event. Based on the generated graph, `IRONSTONE` uses a multi-start random constructive heuristic that schedules vehicles to respond to e-events. In its current version, `IRONSTONE` uses Tomtom API to estimate the travel time between the points where the vehicles are and the e-events. `IRONSTONE` also uses historical data to estimate the required time to respond to an e-event. When a specific e-event is responded to, the proposed approach is executed to regenerate the routes considering the new registered events and the traffic conditions. `IRONSTONE` is designed to minimize the total response time so the objective function minimizes the sum of the times required to respond to each e-event The proposed approach was evaluated with real data from the Civil Defense Department of the City of Niterói. Thus, the main contributions of this article are as follows:

- The formulation of the Multiple depots Open Route mTSP with Priority Constraints (MO-mTSP) problem as an integer programming problem;
- The design of a multi-start routing heuristic that generates feasible routes to respond to e-events;
- The development of an architecture (`IRONSTONE`) that integrates the e-events data, traffic data, and the proposed heuristic;
- An experimental evaluation of the proposed approach with real data from the city of Niterói and synthetic data.

The remainder of this article is structured as follows. In Section II we present related work. Section III presents the Mathematical Formulation. Section IV presents the proposed approach and the heuristic. Section V shows the experimental evaluation, and finally, section VI talks about the conclusion and future work.

II. RELATED WORK

Over the last few years, a growing number of researchers applied efforts to use optimization techniques to foster a fast response to events. Such events vary from natural disasters to 911 calls. This section discusses some of the works that are related to `IRONSTONE` at a certain level.

Usanov *et al.* [16] propose a heuristic to increase the coverage area of fire brigade trucks during major accidents. In their paper, the authors propose a mathematical model to reallocate idle units to reduce the response time in case of new e-events. The heuristic proposed by the authors allows the manager to balance the number of idle unit movements and the coverage area. The model was tested and compared with three benchmarks, with simulations considering 10 years of historical data in the city of Amsterdam, where it was found that there is a substantial improvement compared to models that do not deal with the relocation of idle trucks.

Sakuraba *et al.* [17] present a study in which optimization techniques are applied to find viable paths between emergency units to the population in need, as well as a work-troop scheduling problem to repair areas to expand access to the refugee population. Among the contributions of the paper, the authors present a mathematical formulation and a heuristic to solve the problem of sequencing road repair tasks. The study was carried out using data from the city of Port-au-Prince (Haiti) a couple of hours after the 2010 earthquake.

Wex *et al.* [18] present a decision support model that minimizes the sum of completion times for emergency responses

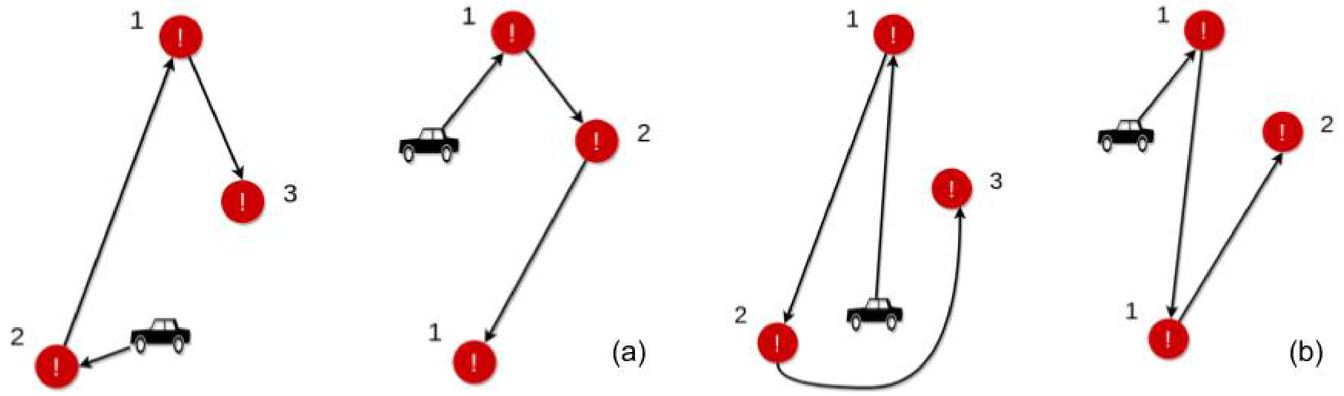


Fig. 2. (a) An example solution of MO-mTSP without priority constraints, and (b) with priority constraints.

after major disasters weighted by their severity. The authors compare several heuristics including those based on the Monte Carlo model and GRASP method. The results are promising, with low processing time and an improvement of up to 80 percent concerning best practices. However, the work of Wex *et al.* does not take into account traffic data.

Erkut *et al.* [19] address the issue of ambulance allocation for medical emergencies in cases of heart attacks. The authors propose a cost function in which the survival rate of patients can be estimated. This function is decreasing as time passes and is incorporated into the allocation and home care model. The authors empirically show the superiority of the proposed model in survival rates when applied to data from the Edmonton health system.

Kula *et al.* [20] present another study in the field of medical responses to emergencies that proposes a mathematical model to minimize ambulance traveling time from causalities to hospitals by selecting the shortest route. The model aims to answer two questions: which ambulance attends which casualty and which hospital to take the casualty.

Ding *et al.* [21] use deep learning techniques to propose a scheduling approach for emergency vehicles (*i.e.*, police, ambulances, *etc*) that considers a route planning module that is coupled to a collaborative traffic signal control module. In this work, the authors assume as a premise that the proposed approach can change traffic signals to generate the best possible route for the emergency vehicle. Unfortunately, this is not the scenario found in many cities. The experiments made by the authors show that their approach outperforms the state-of-the-art baselines when traffic signal control is available.

Other authors aim at minimizing emergency vehicles' travel time by using preemption methods, such as managing the traffic lights on the path [22] [23] [24]. Shaaban *et al.* [25] propose an approach that uses both optimization and preemption techniques which selects the optimal path and then unblocks the intersection en route at the right time to clear it before the emergent vehicle reaches.

III. MATHEMATICAL FORMULATION FOR THE MO-MTSP-P PROBLEM

In this section we define the aforementioned MO-mTSP-P problem using a mathematical formulation. Let us define a direct graph $G = (V, A)$, with vertex set $V = (K \cup E)$ as the set of all vertices, where K is the set of vehicles and E is the set of e-events that need to be responded. $A = \{(i, j), \forall i, j \in V\}$ is denoted the set of arcs, and for each arc $(i, j) \in A$, let t_{ij} represent the minimum time (in minutes) to reach vertex j from vertex i. Moreover, each vertex $i \in V$ has an associated time t_i representing an upper bound of the time required to respond to an e-event i (if $i \in E$), or the time left to respond to an event by a vehicle i (if $i \in K$). Note that if vehicle $i \in K$ is not responding to any e-event, then $t_i = 0$. Let us also define $(E_1, E_2, \cdots, E_q)$ as a partition of events set E into q priority subsets, *i.e.*, $(E_1 \cup \cdots \cup E_q) = E$, and $(E_p \cap E_w) = \emptyset$, for every $p, w = 1, \cdots, q$ with $p \neq w$, where events in E_p has higher priority than events in E_w iff $p < w$. The variables used in our formulation are presented in Table I.

TABLE I
TABLE OF VARIABLES

Variables	Description
$x_{ij} \in \{0, 1\}$	Binary variable that assumes value 1, if and only if a vehicle in some route travel through arc $(i, j) \in A$.
$w_i \in \mathbb{R}_+$	Arrival time in vertex $i \in V$.
$t_{min} \in \mathbb{R}_+$	time needed to attend the last e-event in the longest route.

The formulation for the routing problem is presented following where the objective Function 1 minimizes the time of the longest route. Constraints 2 ensure that every e-event is responded, while Constraints 3 express that at most one vehicle can leave from any vertex. Constraints 4 guarantee that routes can only reach the location of e-events (and no other location that is not associated with an e-event). Moreover, constraints 5 are subtour elimination inequalities (where M is a large enough positive integer), while Constraints 6 rule that the service order respects the priorities of the e-events. Finally, Constraints 7 establish the correct time to attend the last e-event in the longest route.

$$\min t_{min} \tag{1}$$

subject to:

$$\sum_{ij \in A} x_{ij} = 1,\ \forall j \in E \tag{2}$$

$$\sum_{ji \in A} x_{ji} \leq 1,\ \forall j \in E \cup K, \tag{3}$$

$$\sum_{ik \in A} x_{ik} = 0,\ \forall k \in K \tag{4}$$

$$w_i + t_i + t_{ij} - w_j - M(1 - x_{ij}) \leq 0,\ \forall (i,j) \in A \tag{5}$$

$$w_i - w_j \leq 0,\ \forall i \in E_l, \forall j \in E_{l+1},\ l = 1, ..., q-1 \tag{6}$$

$$w_i + t_i - t_{min} \leq 0,\ \forall i \in E \tag{7}$$

IV. PROPOSED APPROACH: `IRONSTONE`

`IRONSTONE` is a data-oriented approach that aims at defining routes for responding to e-events by consuming real-time traffic data. This section provides details regarding `IRONSTONE` architecture and its routing heuristic.

A. Architecture

The architecture of `IRONSTONE` is presented in Fig. 3, and is composed of five main components: (i) Data warehouse, (ii) e-events Processor, (iii) Traffic Data Extractor, (iv) Routing Heuristic, and (v) API. To allow for routing vehicles to respond to e-events, all data related to e-events must be gathered. `INRONSTONE` accesses data available in a Data Warehouse (DW) [26] that contains e-event data aggregated in multiple time and space granularities. In its current version, the DW is implemented following the ROLAP model (Relational OLAP) using PostgreSQL RDBMS. All data are obtained by submitting SQL queries that return a relation that contains the type of e-event, the time, the date, the latitude, the longitude, and the e-event priority (a Likert scale that varies from 1 to 5, where 1 is the top priority e-event). All these data are processed by the *e-event Processor* component, which filters the e-events that are active in the current time window. The time window is based on a θ threshold, where the time window is defined by $[CurrDate - \theta, CurrDate + \theta]$, where $CurrDate$ is the current date and time.

Once `IRONSTONE` has access to e-events data, the *Traffic Data Extractor* component creates a directed graph to support the routing task, as the graph G defined in Section III. All real-time traffic data are obtained by accessing the TomTom API [2] through an in-house algorithm written in Python that considers the latitude and longitude of each e-event and the position of the vehicles. TomTom was the chosen API because it offers a vast and high-quality mapping database and highly accurate and frequently updated traffic information. In preliminary tests performed by the authors, we concluded that TomTom updates traffic data more frequently in Niterói than other APIs. With the traffic data available, `IRONSTONE` updates the graph that will be used as input to the routing heuristic. Finally, the *Routing Heuristic* component receives the georeferenced e-events data and the traffic graph. It generates the routes for each vehicle to respond to all e-events as fast as possible. The generated routes can be accessed by the user using the `IRONSTONE` API. Following, we provide more details regarding the proposed routing heuristic.

B. Routing Heuristic

Defining the routes for each vehicle can be a complex task depending on the number of e-events to be responded to. The number of possible routes for each vehicle may be huge, which makes the adoption of heuristic a natural solution for the MO-mTSP-P problem.

Heuristics do not aim at finding the best solution to a problem, instead, they try to find near-optimal solutions, *i.e.* solutions that are acceptable for a specific problem [27]. Considering the MO-mTSP-P problem the heuristic can be applied to improve the results and minimize the total time to respond to all the e-events registered by the Civil Defense Department of Niterói. On the other hand, we must be careful when developing the heuristic, because the method must be simple and fast enough (designed to execute in less than a minute) to be useful in real scenarios during major climate events.

Thus, in this section, we provide details regarding a fast random multi-start heuristic that generates feasible routes to respond to e-events in less than one minute.

The main idea of the proposed constructive heuristic is to build the route for each vehicle, by adding one e-event at a time to the route. The method iteratively creates a list of all possible e-events responses $e \in E$ in the route of a vehicle $k \in K$. This candidate's list is then restricted and sorted by time of arrival (*i.e.*, the time necessary for a vehicle in route k to reach and respond to an e-event e), and then one e-event is chosen to be part of a route in the partial solution. This process is repeated until all e-events are finally inserted in some route. The detailed algorithm is described in Algorithm 1.

Algorithm 1: $\alpha-$Heuristic

```
Input: $\alpha$
Output: Set of routes for every vehicle $k \in K$
$Route_k \leftarrow \emptyset\ , \forall k \in K$;
$\overline{E} \leftarrow E$;
$p = 1$;
while $\overline{E} \neq \emptyset$ do
    $CL \leftarrow \emptyset$;
    foreach $k \in K$ do
        foreach $e \in E_p$ do
            $CL \leftarrow CL \cup candidate(k, e)$;
    if $CL = \emptyset$ then
        $p = p + 1$;
    else
        $CL \leftarrow$ orderByTime($CL$);
        $RCL \leftarrow$ createRCL($CL$, $\alpha$);
        $(k', e') \leftarrow$ getRandom($RCL$);
        $Route_{k'} \leftarrow Route_{k'} + \{e'\}$;
        $\overline{E} \leftarrow \overline{E} - \{e'\}$;
return $Route$;
```

---

[2] https://www.tomtom.com/pt_br/

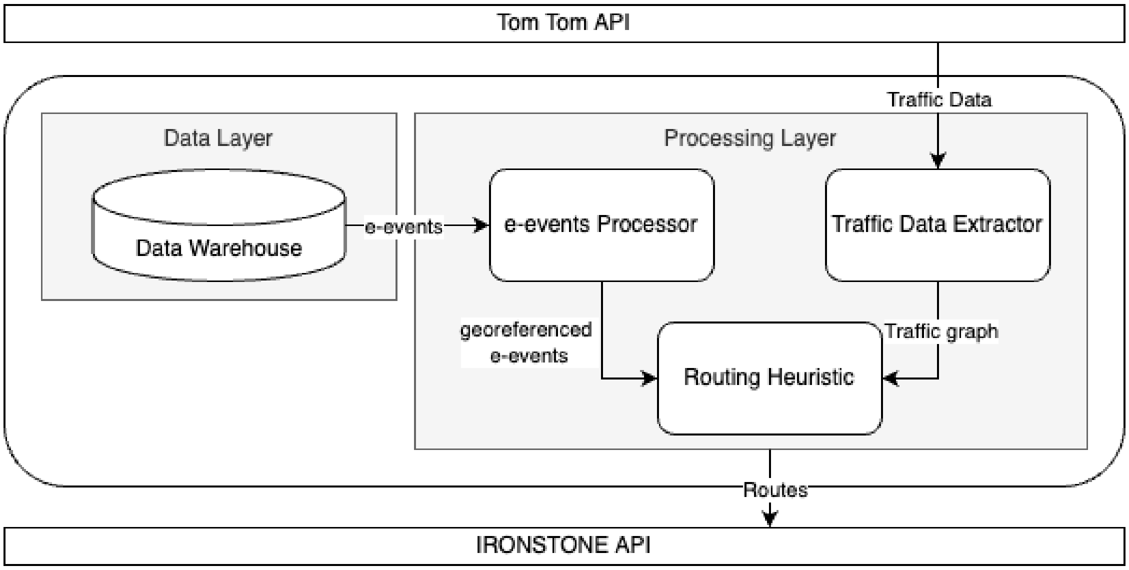


Fig. 3. `IRONSTONE` proposed architecture

The heuristic begins with empty routes, *i.e.*, each and every vehicle $k \in K$ is in its start position (line 1), and build each route, one e-event at a time in a loop from line 4 to 16. In this loop, a candidate's list (CL) is built by considering each vehicle in route $k \in K$ responding to each possible e-event $e \in E_p$ on lines 6 to 8. Note that only e-events with priority p are considered since the response order must respect e-events priorities. Once there is no more unresponded e-events with priority p, the method increases the priority on lines 9 to 10. After every possible association of an e-event to a route is computed, the CL is increasingly ordered by time of arrival – line 12. Then, a restricted candidate list (RCL) is created (line 13) considering only the first $\alpha\%$ best candidates. Finally, a random candidate (k', e') is selected from RCL and emergency event e' is assigned as the next destination in route k' – lines 14 to 16.

Given the randomized characteristics of the constructive algorithm, we need to repeat the entire process (Algorithm 2) to minimize the random effect and find out a better solution. Thus, the $\alpha-$Heuristic is executed Max_{Iter} times iterations without improvements after which the algorithm stops (lines 4 to 16). To evaluate different values of α we apply a dynamic approach. The parameter starts with value α_{ini} (line 1) and if the method fails to improve the solution after Max^{α}_{Iter} iterations, we increase the α value by a factor of Inc_{α} (lines 14 to 16), trying to increase the randomness of the proposed method. Note that if the solution is improved, the method reassigns α with its initial value (line 10). This α dynamic approach is adopted because depending on the value chosen for the parameter, the impact on the diversity of the solutions is different. If the value of α is close to 100%, the algorithm chooses e-events assignments from a larger list of candidates, which could lead to routes with longer response times. On the other hand, if a small percentage is chosen for α, then this may be too restrictive and decrease the diversity of the method. The method stops after 60 seconds, even if the maximum number of iterations was not reached.

It is worth mentioning that the proposed heuristic is not dynamic, *i.e.*, it generates routes given a specific configuration of e-events, car traffic, and emergency vehicles and does not consider new e-events or changes in the traffic conditions after the algorithm starts. However, it can be easily adapted to be applied in a scenario where e-events are dynamically registered and the traffic conditions change constantly. Since the proposed heuristic is designed to execute in less than a minute, it can be executed in multiple phases, *i.e.* it can be invoked every time an e-event is responded or a specific number Σ of new e-events is registered. Let us consider the example presented in Fig. 4 where in Phase 1, six e-events are responded to by two vehicles A and B. After the heuristic generates the routes, vehicle B already responded to the first e-event, and new two e-events are registered (in yellow) with priority 1. Then, in Phase 2, the heuristic regenerates the routes considering the position of vehicle B and the new two e-events.

Algorithm 2: Routing Heuristic

```
   Input: $\alpha_{ini}, Inc_{\alpha}, Max^{\alpha}_{Iter}, Max_{Iter}$
   Output: Set of routes for every vehicle $k \in K$
1  $\alpha \leftarrow \alpha_{ini}$;
2  $f^* \leftarrow \infty$;
3  $it \leftarrow 0 \quad it_{\alpha} \leftarrow 0$;
4  while $it < Max_{Iter}$ do
5      $Route \leftarrow \alpha-$Heuristic($\alpha$);
6      if $f(Route) < f^*$ then
7          $Route^* \leftarrow Route$;
8          $f^* \leftarrow f(Route)$;
9          $it \leftarrow 0 \quad it_{\alpha} \leftarrow 0$;
10         $\alpha \leftarrow \alpha_{ini}$;
11     else
12         $it \leftarrow it + 1 \qquad it_{\alpha} \leftarrow it_{\alpha} + 1$;
13
14     if $it_{\alpha} > Max^{\alpha}_{Iter}$ then
15         $\alpha \leftarrow \alpha + Inc_{\alpha}$;
16         $it_{\alpha} \leftarrow 0$;
17 return $Route*$;
```

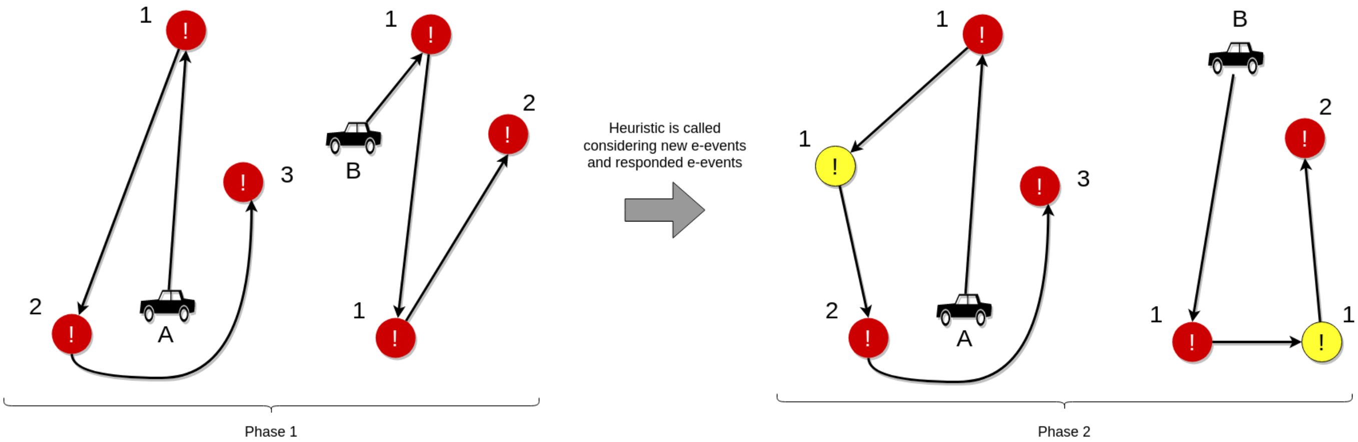


Fig. 4. Two phases of the proposed heuristic

V. EXPERIMENTAL EVALUATION

To evaluate `IRONSTONE`, two different experiments were conducted. The main results achieved so far are presented and discussed in this section. Firstly, we provide information on how the computing environment was set up. Then, we discuss the experimental evaluation of `IRONSTONE`. All data used in the experiments presented in this section are available for download in our group repository in GitHub[3]. The heuristic explained in Section IV was implemented in Python using Jupyter Notebooks to foster reproducibility. Experiments were carried out on a computer with Intel Core i5 CPU 8265U @ 1.80GHz, 16GB RAM, running under Windows 11 home OS. Moreover, the mathematical formulation, denoted here as `IP-mTSP`, was solved using the IBM ILOG CPLEX version 22.1, with a time limit of 7,200 seconds and all other CPLEX parameters left to their default values.

Unlike the conventional TSP problem, no benchmark instance is available to test MO-mTSP-P. Hence, the experiments discussed in this section used as input a series of real-world data (e-events from 2002 to 2022) provided by the Civil Defense Department of Niterói. We have selected days with more than ten e-events. A total of 11 days (*i.e.*, instances) are considered in the experiments. Each instance contains a series of metadata associated with a specific day: (i) an ID that identifies the instance, (ii) the day on which the citizens registered the e-event, (iii) the number of e-events with an associated priority, and (iv) the number of vehicles available. Table II presents the characteristics of each of the obtained instances.

Moreover, the parameter settings of the proposed routing heuristic were obtained with the irace parameter configuration tool [28] based on 5 instances not considered in the experiments. The parameters are shown in Table III with their corresponding types and intervals, as well as the best values returned by irace.

TABLE II
DESCRIPTION OF THE INSTANCES USED IN THE EXPERIMENTS

ID	Day	# of e-events	# of vehicles
DC1	04/10/2019	15	3
Ins01	01/01/2021	10	2
Ins02	12/08/2020	14	2
Ins03	09/22/2020	16	2
Ins04	03/05/2020	20	3
Ins05	01/13/2020	16	2
Ins06	12/10/2020	23	3
Ins07	04/09/2019	27	3
Ins08	03/01/2020	15	3
Ins09	12/09/2020	16	3
Ins10	09/23/2020	17	3

TABLE III
PARAMETER SETTINGS

Parameter	Type	Interval	IRACE
α_{ini}	Real	[0.1,0.5]	0.2
Inc_{α}	Real	[0.1,0.3]	0.1
Max^{α}_{Iter}	Integer	[500, 2000]	1048
Max_{Iter}	Integer	[3000, 10000]	4612

We have performed two experiments to evaluate `IRONSTONE`. The first one considers only one instance and compares the routes generated by `IRONSTONE` with (i) the routes generated by the experts in the Civil Defense Department of Niterói, (ii) a simple greedy algorithm that emulates the routing procedure done by the experts (*i.e.*, vehicles always seek the closest unresponded e-event with the lowest priority), and (iii) the exact method based in the mathematical formulation proposed in Section III. However, as the real routes are confidential and strategic data for the Civil Defense Department, the authors had access only to the routes for a single day, and, since there is no existing study on MO-mTSP-P, the second experiment with the remaining instances compares the results of `IRONSTONE` with only the greedy and exact methods.

For the first experiment, we considered the e-events that occurred on April 10th, 2019. There were 15 e-events on that day and three available vehicles to respond to such e-events.

---

[3] https://github.com/UFFeScience/ironstone

The goal is to compare the routes generated by experts of the Civil Defense Department to the ones generated by the proposed approaches. Fig. 5 illustrates the routes generated by the experts of the Civil Defense Department (a), routes generated by a greedy algorithm that emulates the real routes (b), and the routes generated by the proposed heuristic.

Table IV presents the results obtained for the first experiment. The first column reports the solution value presented by the experts, while the second and third columns present respectively, the solution and the execution time in seconds for the greedy approach. The next two columns display the average solution and time of for the heuristic `IRONSTONE` respectively, after ten runs. The next two columns display the same information as the second and third columns, but for the mathematical formulation.

The routes proposed by the experts generated a solution whose time of the longest route to respond to e-events was about 231.3 minutes, similar to the best solution presented by the exact method (230.4), that was not able to identify the optimal solution within the time limit of 2 hours. On the other hand, the greedy heuristic is very fast and could find a solution of 205.3, 26 minutes less than the solution presented by the expert. The best solution was found by `IRONSTONE`, which was able to reduce the solution value to 180.1 minutes (51.2 minutes less than the expert solution) in only 60 seconds. By analyzing Fig. 5, one can note that in the real routes (a) and the ones generated by the greedy algorithm (b), the vehicles have to cover a larger area, and consequently face more traffic. On the other hand, the routes generated by the proposed heuristic keep the vehicles in certain areas, thus reducing the chance to face heavy traffic.

As previously mentioned, we do not have access to more real routes, thus we now replicate the experiment in other instances comparing `IRONSTONE` with only the greedy and exact methods. Table V shows the results comparing the proposed approaches, the columns of this table have the same meaning as the columns in Table IV (except for column Expert). The results in boldface represent that the optimal solution value was found by each method. We note that `IRONSTONE` heuristic obtained the best average results in all 10 instances when compared to the greedy method. The greedy method reached an average longest route time of 289.3 minutes while `IRONSTONE` reached an average of 266.9 minutes using the average of 58.73 seconds of execution time. Moreover, the value obtained by the proposed heuristic was very accurate, reaching a gap value of only 2.33% when compared to the best solution found by the exact method, even being able to find the optimal value in one instance (Ins08). On the other hand, the exact method was able to find the optimal solution in 7 (out of 10) instances but at the cost of a running time substantially greater (2427.67 seconds on average) than that required to be used in a practical scenario. Thus, we can observe that the proposed heuristic performs well in complex scenarios, with a large number of vehicles and e-events. It can be especially useful on critical days when the number of e-events can be very high.

TABLE IV
COMPARISON OF THE ROUTES GENERATED BY THE PROPOSED HEURISTIC WITH THE REAL ROUTES GENERATED BY EXPERTS.

Expert	Greedy	Time	`IRONSTONE`	Time	`IP-mTSP`	Time
231.3	205.3	0.01	180.1	60.0	230.4	7200.00

TABLE V
COMPARISON OF THE ROUTES GENERATED BY THE PROPOSED HEURISTIC WITH THE ROUTES GENERATED BY THE GREEDY ALGORITHM.

ID	Greedy	Time	`IRONSTONE`	Time	`IP-mTSP`	Time
Ins01	246.2	0.01	227.8	48.75	**226.9**	3.98
Ins02	315.0	0.01	291.8	59.06	**284.9**	405.29
Ins03	439.9	0.01	404.9	60.00	**380.3**	30.33
Ins04	257.7	0.02	240.2	60.00	237.7	7200.00
Ins05	311.5	0.01	298.3	59.94	**295.0**	1358.81
Ins06	311.0	0.03	284.8	60.00	271.8	7200.00
Ins07	316.9	0.04	298.0	60.00	285.8	7200.00
Ins08	194.3	0.01	**178.0**	59.57	**178.0**	44.05
Ins09	211.1	0.02	201.4	60.00	**197.1**	153.23
Ins10	290.3	0.02	243.8	60.00	**242.3**	681.03
Avg	289.3	0.01	266.9	58.73	255.2	2427.67

VI. CONCLUSIONS AND FUTURE WORK

In this article, we address the MO-mTSP-P problem faced by the Civil Defense Department of the City of Niterói during e-events when the number of emergency calls is greater than the system's ability to respond to them properly. Thus, it is a top priority to determine the response order according to the defined priority and severity.

The proposed approach, named `IRONSTONE`, uses historical data (provided by the Civil Defense Department) and real-time traffic data gathered from the Tomtom Routing API to generate a graph representing the streets of the city. This graph is used as input to the proposed heuristic to generate routes for each vehicle to respond to e-evens as fast as possible. The use of real-time traffic data provides a better estimate of travel time which makes the solution more accurate.

The proposed heuristic aims to minimize the total response time and we compared the generated routes with the solution provided by an expert, a greedy algorithm, and an exact method. The results show that the proposed heuristic reduced in 51.2 minutes the time of the longest route compared to the solution provided by an expert and used in a real scenario. When compared to a greedy algorithm the heuristic proposed reduced the response time up to 46.5 minutes in a day with 17 e-events and 3 vehicles. On average the proposed heuristic generated routes where the response times are reduced by 22.3 minutes compared to the greedy method, and only 11.7 minutes longer than the best solution found by the exact method. The results show to be more promising in larger instances which makes the optimization problem more complex for an expert to do without adequate computing support. As future work, we plan to improve the heuristic by developing local search methods to use different types of metaheuristics as GRASP and ILS, while maintaining execution time under a minute.

Fig. 5. Routes generated for responding to the e-events on April 10th, 2019. Image (a) shows the routes generated by the experts of the Civil Defense Department. Image (b) shows the routes produced by a greedy algorithm that emulates the real routes. Last, image (c) shows the routes generated by the proposed heuristic.

ACKNOWLEDGMENTS

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. The authors also thank CNPq and FAPERJ.

REFERENCES

[1] A. Hameed, J. Violos, and A. Leivadeas, “A deep learning approach for iot traffic multi-classification in a smart-city scenario,” *IEEE Access*, vol. 10, pp. 21193–21210, 2022.

[2] R. Sharma and R. Arya, “UAV based long range environment monitoring system with industry 5.0 perspectives for smart city infrastructure,” *Comput. Ind. Eng.*, vol. 168, p. 108066, 2022.

[3] S. Tiwari, A. Jain, N. M. O. S. Ahmed, Charu, L. M. Alkwai, A. K. Y. Dafhalla, and S. A. S. Hamad, “Machine learning-based model for prediction of power consumption in smart grid- smart way towards smart city,” *Expert Syst. J. Knowl. Eng.*, vol. 39, no. 5, 2022.

[4] V. Lourenço, P. Mann, A. Guimaraes, A. Paes, and D. de Oliveira, “Towards safer (smart) cities: Discovering urban crime patterns using logic-based relational machine learning,” in *2018 International Joint Conference on Neural Networks, IJCNN 2018, Rio de Janeiro, Brazil, July 8-13, 2018*, pp. 1–8, IEEE, 2018.

[5] P. Lombardi, S. Giordano, H. Farouh, and W. Yousef, “Modelling the smart city performance,” *Innovation: The European Journal of Social Sciences*, vol. 25, 06 2012.

[6] P. Kumar, R. Chandra, C. Bansal, S. Kalyanaraman, T. Ganu, and M. Grant, “Micro-climate prediction - multi scale encoder-decoder based deep learning framework,” KDD, p. 3128–3138, 2021.

[7] D. Rolnick, P. L. Donti, L. H. Kaack, K. Kochanski, A. Lacoste, K. Sankaran, *et al.*, “Tackling climate change with machine learning,” vol. 55, no. 2, 2022.

[8] T. Bektas, “The multiple traveling salesman problem: an overview of formulations and solution procedures,” *Omega: The International Journal of Management Science*, vol. 34, no. 3, pp. 209–219, 2006.

[9] O. Cheikhrouhou and I. Khoufi, “A comprehensive survey on the multiple traveling salesman problem: Applications, approaches and taxonomy,” *Computer Science Review*, vol. 40, p. 100369, 2021.

[10] X. Wang, D. Liu, and M. Hou, “A novel method for multiple depot and open paths, multiple traveling salesmen problem,” in *2013 IEEE 11th International Symposium on Applied Machine Intelligence and Informatics (SAMI)*, pp. 187–192, 2013.

[11] M. Assaf and M. Ndiaye, “Solving an open path multiple depot multiple traveling salesman problem after transformation,” in *2017 7th International Conference on Modeling, Simulation, and Applied Optimization (ICMSAO)*, pp. 1–4, 2017.

[12] P. Lou, K. Xu, J. Yan, and Z. Xiao, “An improved partheno-genetic algorithm for open path multi-depot multiple traveling salesmen problem,” *Journal of Physics: Conference Series*, vol. 1848, p. 012002, apr 2021.

[13] B. Cunha Sá, G. Muller, M. Banni, W. Santos, M. Lage, I. Rosseti, Y. Frota, and D. de Oliveira, “Polroute-ds: a crime dataset for optimization-based police patrol routing,” *Journal of Information and Data Management*, vol. 13, Aug. 2022.

[14] S. Humagain, R. Sinha, E. Lai, and P. Ranjitkar, “A systematic review of route optimisation and pre-emption methods for emergency vehicles,” *Transport reviews*, vol. 40, no. 1, pp. 35–53, 2020.

[15] B. Gong, Z. Yang, and C. Lin, “Dispatching optimization and routing guidance for emergency vehicles in disaster,” in *2009 IEEE International Conference on Automation and Logistics*, pp. 1121–1126, IEEE, 2009.

[16] D. Usanov, G. Guido Legemaate, P. M. van de Ven, and R. D. van der Mei, “Fire truck relocation during major incidents,” *Naval Research Logistics (NRL)*, vol. 66, no. 2, pp. 105–122, 2019.

[17] C. Sakuraba, A. Santos, C. Prins, L. Bouillot, A. Durand, and B. Allenbach, “Road network emergency accessibility planning after a major earthquake,” *EURO Journal on Computational Optimization*, vol. 4, no. 3-4, pp. 381–402, 2016.

[18] F. Wex, G. Schryen, S. Feuerriegel, and D. Neumann, “Emergency response in natural disaster management: Allocation and scheduling of rescue units,” *European Journal of Operational Research*, vol. 235, no. 3, pp. 697–708, 2014.

[19] E. Erkut, A. Ingolfsson, and G. Erdoğan, “Ambulance location for maximum survival,” *Naval Research Logistics (NRL)*, vol. 55, no. 1, pp. 42–58, 2008.

[20] U. Kula, O. Tozanli, and S. Tarakcio, “Emergency vehicle routing in disaster response operations,” in *POMS 23rd Annual Conference, Chicago (April 20-23)*, 2012.

[21] L. Ding, D. Zhao, Z. Wang, G. Wang, C. Tan, L. Fan, and H. Ma, “Learning to help emergency vehicles arrive faster: A cooperative vehicle-road scheduling approach,” *arXiv preprint arXiv:2202.09773*, 2022.

[22] C. Chen, H. Wei, N. Xu, G. Zheng, M. Yang, Y. Xiong, K. Xu, and Z. Li, “Toward a thousand lights: Decentralized deep reinforcement learning for large-scale traffic signal control,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 3414–3421, 2020.

[23] L. Bieker-Walz and M. Behrisch, “Modelling green waves for emergency vehicles using connected traffic data,” *EPiC Series in Computing*, no. 62, pp. 1–11, 2019.

[24] S. Humagain and R. Sinha, “Routing emergency vehicles in arterial road networks using real-time mixed criticality systems,” in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–6, IEEE, 2020.

[25] K. Shaaban, M. A. Khan, R. Hamila, and M. Ghanim, “A strategy for emergency vehicle preemption and route selection,” *Arabian Journal for Science and Engineering*, vol. 44, no. 10, pp. 8905–8913, 2019.

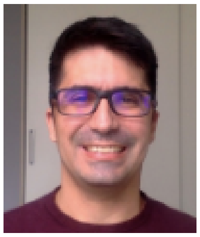
[26] R. Kimball and M. Ross, *The Data Warehouse Toolkit: The complete guide to dimensional modeling*. New York: Wiley, 2002.

[27] G. Gigerenzer, “Why heuristics work,” *Perspectives on psychological science*, vol. 3, no. 1, pp. 20–29, 2008.

[28] M. López-Ibáñez, J. Dubois-Lacoste, L. Pérez Cáceres, M. Birattari, and T. Stützle, “The irace package: Iterated racing for automatic algorithm configuration,” *Operations Research Perspectives*, vol. 3, pp. 43–58, 2016.

Luis Felipe Oliveira is a master's student at the Institute of Computing at Universidade Federal Fluminense (UFF). He graduated in Engineering from CEFET in 2006 and obtained an M.S. degree in Engineering from UFF (2016). His research interests include optimization problems and machine learning.

Yuri Frota is an associate professor in the Department of Computer Science at Universidade Federal Fluminense, where he has been since 2010. He graduated in Computer Science from UECE (1999), obtained an M.S. in Computer Science from UFC (2002), and a Ph.D. also in Computer Science from UFRJ (2008). He has experience in computer science, focusing on algorithms, and is engaged in the following subjects: combinatorial optimization and integer programming.



Daniel de Oliveira is an associate professor of computer science at Universidade Federal Fluminense in Niterói, Brazil since 2013. His current research interests include scientific workflows, provenance, cloud computing, data scalable and intensive computing, high-performance computing, and distributed and parallel databases. He serves or served on the program committee of major international and national conferences (VLDB, IPAW, IEEE eScience, SBBD, *etc*.). He has published many technical papers and is a co-author of the book "Data-Intensive Workflow Management For Clouds and Data-Intensive and Scalable Computing Environments" published by Morgan & Claypool (now Springer) in 2019.