MACHINE LEARNING TECHNIQUES APPLIED TO THE VEHICLE ROUTING PROBLEM FOR WASTE SITE INSPECTIONS

FRANCESCO NEBULA, ROBERTO PALUMBO & GABRIELLA GIGANTE
Complex, Intelligent and Autonomous Systems, Italian Aerospace Research Centre (CIRA), Italy

ABSTRACT

During past decades, illegal acts of waste dumping occurred many times in the Campania region of southern Italy. Tons of waste have been dumped in agricultural areas and often illegally burned posing serious health threats to the neighbouring population. Authorities are trying to mitigate such phenomenon by means of continuous monitoring and through actions of remediation. Nevertheless, the number of sites to be visited is often very high with respect to the resources that can be deployed on the territory (such as personnel, patrols, etc.). At the same time, these resources are limited and therefore they should be used in an optimized way. In addition, patrols have to comply with assigned time constraints, considering both the travel time and the time needed to inspect the site. Currently, the tours are assigned by human operators according to their own experience but this solution is highly inefficient. Therefore, a computational tool capable of finding optimal inspection tours in reasonable time is highly desired. For this reason, an appropriate process of risk assessment and priority assignment was set up in order to put in place an optimal resource management from an operational point of view. This work confronts the problem of finding the vehicle routes in a way that a "priority index" is globally maximized. This kind of problem belongs to the class of NP-hard problems and it cannot be solved exactly in practicable time even for a few sites. This paper explores the application of some machine learning techniques to solve the described problem, also performing a comparison with the results obtained using a classical heuristic algorithm taken from literature. In more detail, the use of data clustering and Monte Carlo Tree Search approach is investigated with satisfactory results.

Keywords: vehicle routing problem with profits, clustering, Monte Carlo Tree Search.

1 INTRODUCTION

In the recent years, many illegal waste sites were discovered in the Campania region, in southern Italy. In fact, many agricultural and sub-urban areas are scattered with countless micro-waste dumps. Such dumps typically contain urban, agricultural or even industrial wastes and often flammable materials. These materials may sometimes undergo open-burning because of spontaneous combustion or even because of malicious intents. Open-burnings are a source of cancerogenic and toxic substances, and may cause several diseases in the neighbouring population [1]. Of course, local authorities are trying to mitigate such phenomenon by means of a continuous monitoring of the region and through several actions of remediation.

Within the Crowd for the Environment (C4E) project, several emerging information technologies are integrated in an innovative framework to help local environmental authorities discover and monitor illegal dumping sites. In particular, the detection of the dumping sites comes from spontaneous reports of citizens (that use a specific mobile app for crowd sourcing) and periodic satellite image acquisitions (artificial intelligence extracts areas potentially affected by illegal dumping). Each discovered site is characterized by its spatial position, an estimation of its volume and a rough estimation of the kind waste therein. These kinds of information allow the authorities set up an appropriate process of risk assessment. This process allows to define a "priority" for the inspection of each waste site (which is

WIT Transactions on Ecology and the Environment, Vol 257,
www.witpress.com, ISSN 1743-3541 (on-line)
doi:10.2495/WMEI220121

related to the potential danger posed by the site) and then arrange an inspection campaign to assess the validity of the site report and to decide the remediation actions needed.

Actually, one of the objectives of the C4E project is to help local authorities organize the on-field inspections of illegal dump sites.

Currently, the tour assignment to each patrol is provided by human operators only, based on their own experience. Unfortunately, this method is highly inefficient being slow and inaccurate. For these reasons, it is strongly desired to determine the tour assignments automatically with the help of a computational tool. Consequently, our problem consists in finding a computational procedure that determines the vehicles routes in such a way that an appropriate index – hereafter referred as the "overall collected priority" – is maximized, also considering some time constraints like, for example, the work shifts of the operators. This problem belongs to the class of the so-called vehicle routing problems (VRPs) and is generally very hard to solve, even for few numbers of nodes (NP-hard problems).

In a previous work [2] a heuristic algorithm from literature [3] was successfully applied to solve the described problem in a similar operative scenario. However, in this work, we explore an alternative approach leveraging on machine learning (ML) techniques, comparing its outcomes to the same heuristics.

This paper is organized as follows. Section 2 reports the description of the problem to be solved. Section 3 describes the methodology and tools used to build the proposed solution. Section 4 describes the procedure followed to evaluate the performances of the proposed solution, and the main results are shown. Finally, Section 5 collects the main conclusions of this work.

2 PROBLEM STATEMENT

2.1 Background

VRP is one of the most studied combinatorial optimization tasks [4]. The attention paid for VRP, along with its many variants, is justified by its many applications in transportation and logistics, vital fields of the modern economy. Because of its practical relevance, there is a strong interest in finding new solution approaches, despite the existence of several heuristic and approximate methods.

Generally speaking, in a VRP, a fleet of vehicles can visit a set of customers, achieving the fulfilment of a specific service. The objective is to schedule the visits in a way that the use of the available resources is optimized, with respect to some given constraints.

In particular, in the VRP with Profits (VRPP) – also known as Team Orienteering Problem (TOP) [5] – the overall collected profits must be maximized, under some predefined constraints (time deadlines, vehicles capacities, etc.) that make usually impossible to visit all customers using one patrol only. Consequently, both the order of visits and the subset of nodes to be visited must be found.

2.2 Approach

For our specific scenario, the following two-steps procedure is implemented. Firstly, the illegal waste site reports for each municipality of the Campania region are initially grouped in some geographical zones (clusters) in order to satisfy a given criterion based, for example, on the maximum admissible visit time. Each cluster is then characterized by three features:

- a "centroid", defined as the cluster "centre of gravity" and representing the above-mentioned "customer";

- a "cumulative" priority, defined as the average of the priorities of the grouped micro-dumps, equal to their volumes weighted by the dangerousness of the contained materials;
- a "cumulative" collection time, sum of two different terms: the minimum travel time to visit (only once) all the sites with a fixed speed V (travelling salesman problem (TSP)) and the sum of times needed to inspect them. Hereafter this total time will be referred as "the time to collect" the cluster priority.

In the second phase, the ML solver is applied on the obtained set of clusters for each municipality. This solution is then compared to a baseline solution, described in Butt and Cavalier [3] and known under the acronym of MAXIMP, for validation purposes. Finally, when the above described strategic step is concluded, all the clusters within each municipality can be "tactically" visited, using the TSP paths computed in the previous step. Fig. 1 shows the described approach.

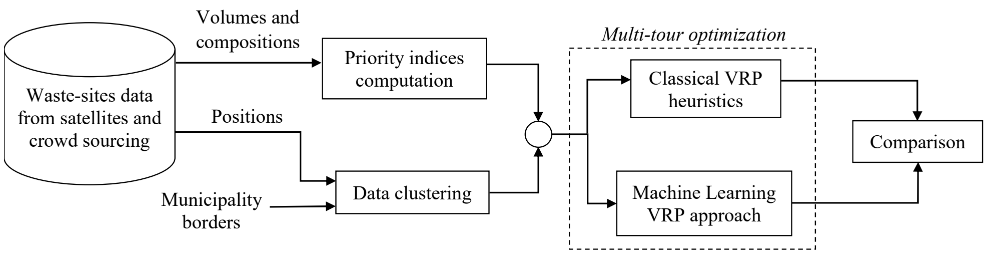

Figure 1: The overall approach followed in this work.

2.3 Formulation

Hereafter we formalize the strategic problem described in Section 2.2 for homogeneous vehicles starting from a unique depot (0) because the general case can be always decomposed in similar problems.

Let us define a complete graph $G=[N,A]$ where $N=\{0, 1, \ldots, n\}$ is the set of centroids and $A=\{a_{ij}\}$ is the set of the arcs between the i-th and j-th centroids ($i,j \in N-\{0\}$). We also define the following quantities:

- the distance matrix $\{d_{ij}\}$, composed by the road distances between the i-th and the j-th centroids, generally asymmetric, i.e., $\{d_{ij}\}\neq\{d_{ji}\}$;
- the priority p_i ($p_0=0$), i.e., the cluster volume, weighted by the harmfulness of the contained waste;
- the time b_i required to collect p_i ($b_0=0$), i.e., the time required for the cluster inspection.

Moreover, a tour is defined as "feasible" through a subset of G if it starts and ends in the depot, if it visits each centroid not more than once and if it is completed within a given maximum time T. Using this definition, our problem is to find not more than m feasible tours such that the overall priorities, collected over all tours, is maximized.

3 TOOLS

3.1 Geographic information systems

When dealing with spatial data and their relations, it is often necessary to analyse, manipulate and process spatial information. This can be carried out using geographic information systems (GIS). Specifically, we used the open-source application QGIS [6] together with Python scripts to:

- visualize the micro-dump sites on the map;
- analyse the characteristics of the sites, in terms of volume and composition;
- visualize the borders of the municipalities;
- compute the number of sites belonging to each municipality;
- compute the road distances between two selected points on the map.

To compute the road distances between the micro-dumps, it is required to know the road graph. The related GIS-readable shapefile was downloaded from OpenStreetMap [7] using a Python library named OSMnx [8]. Next, using the QGIS plug-in called QNEAT3 (QGIS Network Analysis Toolbox 3 [9]), it was possible to compute the matrix of the road distances. Fig. 2 reports the road graph for the area of interest (long./lat. intervals: 13.9°E–14.5°E, 40.79°N–41.15°N).

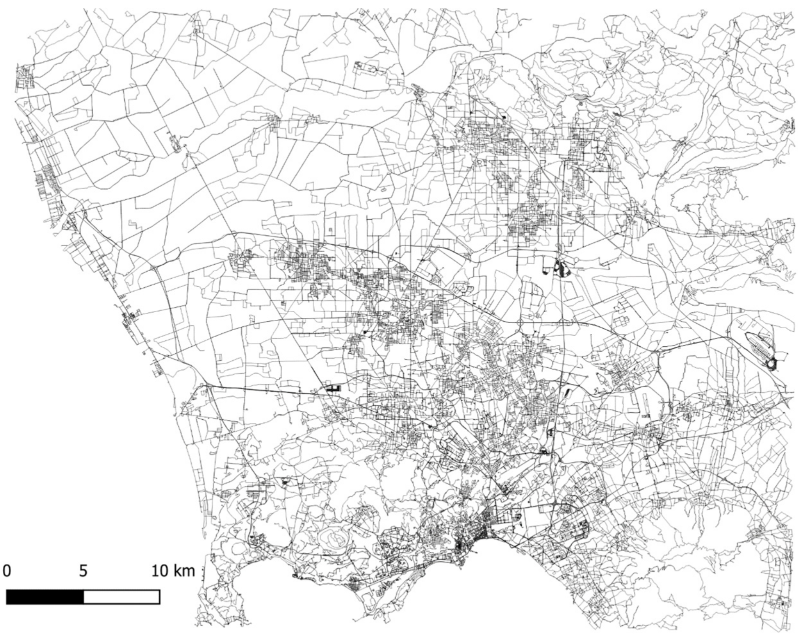

Figure 2: Road graph from OpenStreetMap [7].

It is worth to note that the road distance matrices are generally asymmetric because the connection between two points cannot be always achieved using two-ways roads and a longer path must then be followed. Consequently, points that apparently seem to be “close to each other”, could be very far apart when considering road distances.

3.2 HDBSCAN

Clustering algorithms are used to explore and analyse patterns of similarity in a dataset. Centre-based algorithms find only convex shapes with circular symmetry: they are a bad choice for finding clusters of arbitrary shapes. Instead, density-based clustering is a paradigm where clusters are identified as data partitions with a higher density than their surroundings and therefore they are more suitable for finding clusters of generic shapes. Objects that do not meet a certain density criterion are discarded as "noise".

In addition, partitioning methods group items into a predefined number of categories, while hierarchical algorithms build a hierarchy of data partitions in order to reveal the "intrinsic" data structure (so there is no need to specify the number of categories).

For the proposed problem, and in general for spatial data mining, it is desirable to obtain clusters with arbitrary shapes (in particular long-shaped clusters, due to the fact that micro-dumps usually lie along roads), with varying densities and also without specifying their number.

To this end, a suitable choice appears to be the HDBSCAN algorithm [10], a "Hierarchical" upgrade of DBSCAN (density-based spatial clustering of applications with noise). It is a density-based clustering algorithm, that builds a hierarchical cluster tree (dendrogram), from which the clusters are extracted. In this process, the algorithm is allowed to decide how many clusters to create, using an internal stability measurement.

More in detail, in HDBSCAN the dendrogram is "condensed" applying different cuts at different heights (densities), minimizing the points that are not falling in any cluster: this results in a more compact tree with fewer clusters that lose points and it will be used to select clusters more stable and persistent [11]. This process allows the tree to be cut at varying height, picking our varying-density clusters based on cluster stability.

HDBSCAN has two main parameters:

- "minimum cluster size", imposing the minimum cluster dimension;
- "min samples", less intuitive than the previous one, it represents "how much noise" one wants to exclude in the cluster generation and, consequently, "how many big spatial structures" one wants to consider in the clustering process.

3.3 Monte Carlo Tree Search with UCB-based policy

The Monte Carlo Tree Search (MCTS) [12] is a heuristic search algorithm for some specific decision processes, especially employed in software that play board games: in this context, MCTS is used to solve the game tree, with the notable advantage that only the game mechanics needs to be implemented. In fact, MCTS can theoretically be applied to any domain that can be described by "state and action" pairs and by a simulation to forecast the game outcome. MCTS is typically employed using a tree policy that is based on the upper confidence bound (UCB) [13] formula applied to trees (known as UCT). This approach is in fact effective in solving the "exploration/exploitation dilemma" allowing the algorithm to converge more easily to suboptimal solutions or better.

In this paper, we leverage on the computational power of MCTS to solve the VRP described in Section 2. To this aim, the most critical issue is to map this kind of problem into a game tree formalism. First of all, the MCTS is treated as single-player. The root node of the tree is the starting point of the route namely the depot. Each tree node is then a sub-sequence of visited centroids starting from the depot. Actions from each tree node are all the unvisited centroids that allow the return to depot within the admissible time window. The MCTS selection step is carried out using the abovementioned UCT policy. The rollout step is carried out using a random policy among the possible valid actions. As explained in Section 2, the VRP solution is represented by a sequence of visited centroids that maximizes the collected priority within a predefined amount of time. Therefore, in the MCTS context, the tree game ends as soon as there is no more time left to visit new nodes other than going back to the depot. When this condition is met, the node is considered terminal and a possible solution is found. In order for the UCT to expand the tree in the right directions, solutions must be properly classified as good/poor or promising/unpromising. The reward function and backpropagation step favour only nodes that yield solutions that maximize the collected priority within the time constraint. Following Mańdziuk and Nejman [14] we have also given importance to solutions that appear promising, rewarding – in a linear fashion – solutions for which the resulting collected priority falls in the interval [α*best_priority, best_priority) where α is a parameter less than 1 and set to 0.5, and *best_priority* is the current best collected priority found.

4 RESULTS

4.1 Raw input data

The data considered in this work are inside a specific area of the Campania region (Italy) particularly affected by the problem of the open-burning of waste. This area is identified in Fig. 3 by the dashed line (longitude/latitude intervals: 13.90°E–14.50°E, 40.79°N–41.15°N). The selected area is a rectangle of 51 km × 41 km and, although it represents a small area with respect to the overall Campania region, it contains 66% of the reported micro-dumps (4397 over 6629).

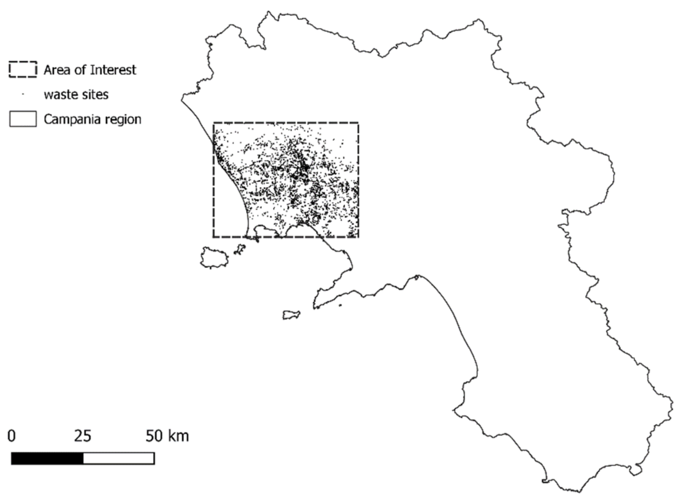

Figure 3: Area selected for the waste sites inspections within the Campania region (Italy).

4.2 Municipalities selection

A geo-tessellation of the Campania region is performed using the borders of the local administrative divisions (Municipalities). The reason of this grouping is that the local patrols are allowed to visit only sites included in their Municipalities.

Furthermore, Municipalities are selected considering the number of the waste-site reports. The areas with less than 20 points are filtered out because in these cases clustering does not confer any significant advantage (Fig. 4).

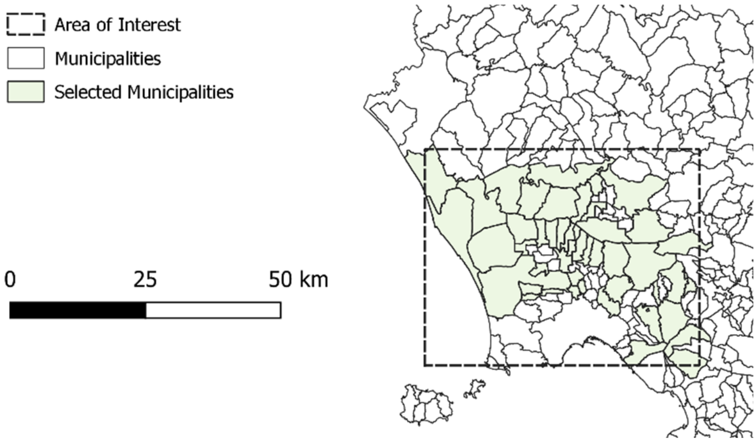

Figure 4: Selected Municipalities.

For example, Fig. 5 shows one of the selected municipalities (Villa Literno), its 204 waste sites (points) and the depot (diamond). The roads are also shown and it can be clearly seen that the waste sites are often located along the roads.

Figure 5: Waste sites, depot and roads (municipality of Villa Literno).

4.3 Clustering analysis

For each selected municipality, the set of waste sites is processed using the HDBSCAN clustering algorithm described in Section 3.2, considering the road distances instead of the line-of-sight (Euclidean) ones.

In this process, an operative requirement is imposed: the total inspection time for each cluster must be the *max*(45 min, 2**k* min) where k is the number of waste sites within the cluster. Imposing this constraint, the two HDBSCAN parameters are chosen consequently.

In Fig. 6 the results of the HDBSCAN clustering are shown for San Tammaro municipality. The points not included in any cluster represent the "noise", i.e., not considered in the outcome of the algorithm. From an operational perspective, these points can be later re-analysed repeating the same two-step procedure described in Section 2.2.

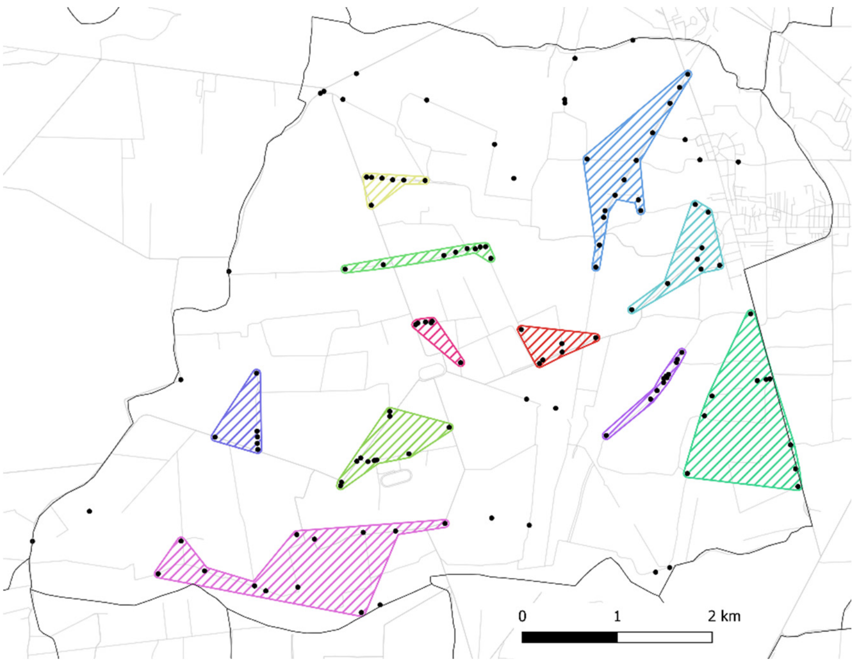

Figure 6: HDBSCAN clustering (municipality of San Tammaro).

As already noted, the obtained clusters are generally long-shaped, reflecting the fact that the micro-dumps are formed mainly along the roads and to group them in this way leads obvious practical advantages for the operators.

In Table 1 some relevant data about the clustering process are reported:

- Municipality ID;
- Name of municipality;
- Total number of the waste sites within the municipality borders;
- Number of clusters;
- Priorities of clusters;
- Percentage of the unclassified waste sites.

Table 1: Clustering data.

ID	Municipality	Total waste sites	Clusters	Cluster priorities (average volume, m^3)	Unclassified waste sites (%)
1	Casal di Principe	95	9	4, 8, 2, 87, 2, 12, 9, 20, 16	27
2	Teverola	37	11	3, 2, 12, 10, 2, 2, 5, 2, 13, 3, 1	11
3	Afragola	131	9	8, 3, 6, 6, 15, 3, 10, 5, 8	25
4	Sant'Antimo	46	9	2, 3, 2, 4, 1, 3, 2, 3, 8	4
5	Casaluce	48	11	4, 9, 4, 2, 9, 3, 140, 8, 8, 4, 4	15
6	Somma Vesuviana	48	6	8, 4, 12, 4, 11, 4	8
7	Castel Volturno	235	11	4, 4, 6, 13, 11, 5, 32, 10, 6, 3, 9	22
8	Macerata Campania	66	10	10, 4, 8, 4, 6, 4, 4, 7, 7, 6	9
9	Parete	21	7	3, 9, 4, 13, 5, 3, 12	19
10	Casoria	40	11	4, 3, 4, 4, 7, 2, 6, 2, 4, 18, 2	15
11	Carinaro	53	7	2, 2, 2, 5, 2, 8, 6	32
12	Succivo	72	11	3, 2, 2, 2, 3, 5, 2, 2, 4, 6, 6	17
13	Casandrino	27	9	10, 2, 3, 5, 2, 6, 4, 4, 2	11
14	Santa Maria Capua Vetere	84	8	2, 10, 6, 10, 5, 8, 5, 4	23
15	Grazzanise	53	9	14, 7, 10, 4, 18, 2, 101, 2, 30	17
16	Villa di Briano	44	8	9, 5, 20, 12, 7, 7, 8, 6	16
17	San Felice a Cancello	29	9	5, 5, 2, 4, 2, 3, 2, 8, 4	17
18	Cancello ed Arnone	35	9	7, 6, 59, 2, 8, 2, 4, 15, 2	17
19	Aversa	24	8	12, 2, 12, 18, 2, 2, 5, 16	4
20	Castello di Cisterna	20	5	8, 10, 34, 15, 8	30
21	Terzigno	36	10	11, 3, 4, 8, 5, 6, 2, 2, 5, 11	11
22	San Vitaliano	23	6	13, 4, 7, 8, 5, 2	17
23	Mondragone	99	11	20, 8, 30, 9, 4, 12, 4, 3, 42, 14, 3	15
24	Ercolano	28	8	3, 8, 2, 8, 3, 2, 4, 4	14
25	San Cipriano d'Aversa	29	9	5, 8, 13, 2, 10, 2, 9, 3, 51	14
26	San Prisco	29	8	16, 4, 4, 5, 11, 52, 4, 6	14
27	Giugliano in Campania	345	11	4, 4, 6, 5, 5, 4, 7, 6, 14, 6, 5	21
28	Caserta	54	10	5, 2, 5, 2, 2, 3, 2, 1, 3, 7	20
29	Qualiano	30	9	252, 3, 2, 7, 1, 2, 2, 10, 1	20
30	Ottaviano	32	5	12, 7, 6, 2, 12	9
31	Santa Maria la Fossa	42	7	1, 2, 21, 26, 8, 4, 3	26
32	Villa Literno	204	11	12, 11, 7, 36, 5, 11, 3, 2, 7, 5, 5	31
33	Orta di Atella	92	10	7, 23, 11, 5, 85, 21, 20, 3, 4, 4	33
34	San Tammaro	125	11	6, 3, 7, 5, 3, 3, 8, 3, 7, 5, 4	20
35	Frignano	62	11	3, 2, 13, 9, 11, 5, 2, 12, 7, 11, 1	10
36	Caivano	139	11	6, 6, 7, 15, 10, 11, 6, 8, 5, 24, 31	19
37	Marcianise	217	10	5, 2, 3, 3, 4, 3, 3, 3, 7, 11	39
38	Marigliano	102	10	8, 5, 7, 6, 7, 14, 7, 6, 10, 5	28
39	Acerra	203	9	8, 16, 14, 10, 15, 9, 7, 4, 3	23
40	Gricignano di Aversa	106	8	3, 4, 3, 6, 2, 10, 4, 9	24
41	Capua	60	11	7, 5, 3, 8, 9, 4, 13, 10, 7, 12, 4	12
42	Sant'Anastasia	61	6	4, 20, 6, 4, 32, 11	13
43	Villaricca	26	7	4, 7, 5, 5, 1, 16, 3	12

4.4 UCT performance evaluation

In this section, UCT performance will be compared to the results of the above-cited MAXIMP algorithm, which is used in the same scenario conditions, selecting an average value of 0.5 for its α parameter (see [3]). UCT performance cannot be compared to the exact optimal solutions as the time required for exhaustive route enumeration is not feasible even for few nodes (for example, just for 9 nodes, 9! = 362880 permutations imply a computational time of about 20 min, with a state-of-art personal computer). Instead, the computational time required by MAXIMP is estimated below one second.

Finally, in our scenario, the UCT computational time, although around an average value of about 70s, is fully compliant with an operational usage on a daily basis.

The problem settings are hereafter reported:

- T = 8 hours, typical Italian work shift;
- m = 3, maximum number of the vehicles starting from the same unique depot;
- V = 20 km/h, constant vehicles speed;
- b_i = 2 min equal for each i-th site, average time needed to perform a site inspection.

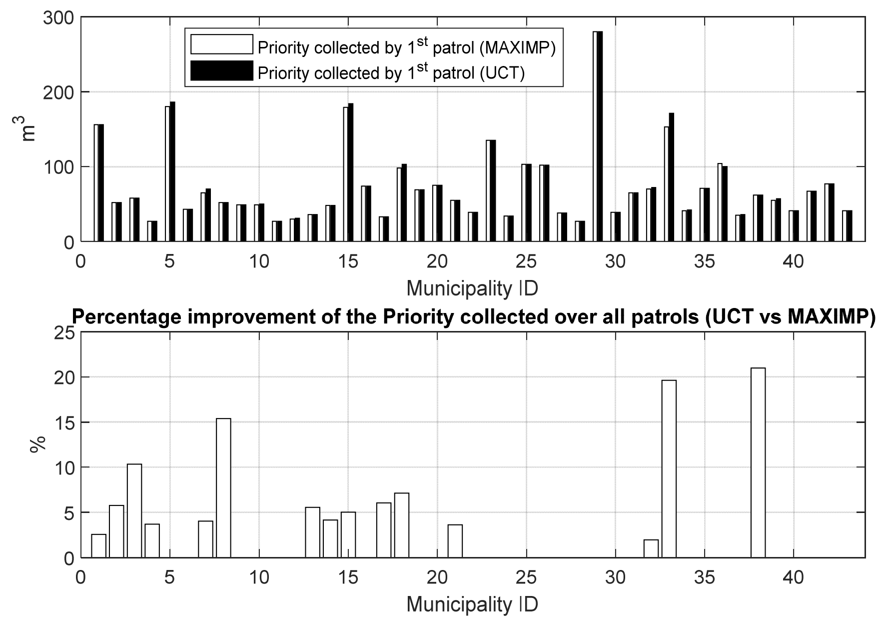

Figure 7: Collected priorities by MAXIMP and UCT (1st patrol only) and UCT improvement.

In Fig. 7 two different information are shown:

- the top chart shows the collected priority (the average cluster volume weighted with the materials dangerousness) of the first patrol only, as planned by both MAXIMP and UTC. It is possible to see that, already using only one patrol, UCT always outperforms the first one, except for the municipality number 36 “Frignano” (104 vs 100).

- the bottom chart shows the percentage improvement of the overall collected priority by UCT with respect to MAXIMP (+2.7% mean, 5.1% std), using all patrols and over all the municipalities. It should be noted that the overall priority collected using UCT always matches the theoretical one (sum of the priority values per municipality (see Table 1)).

For the first patrol only, the mean and the standard deviation values of the mean residual time with respect to the deadline *T*, are [0.79 0.64] hours and [0.41 0.62] hours, respectively for MAXIMP and UCT: this means that the UCT generally exploits better the available time interval *T*.

In terms of the number of the patrols involved, MAXIMP often stops when few sites are left, while UCT always uses the same number or more. However, for the second or third patrol, the solution is sometimes trivial, e.g., only one or two sites remain to be visited (Fig. 8).

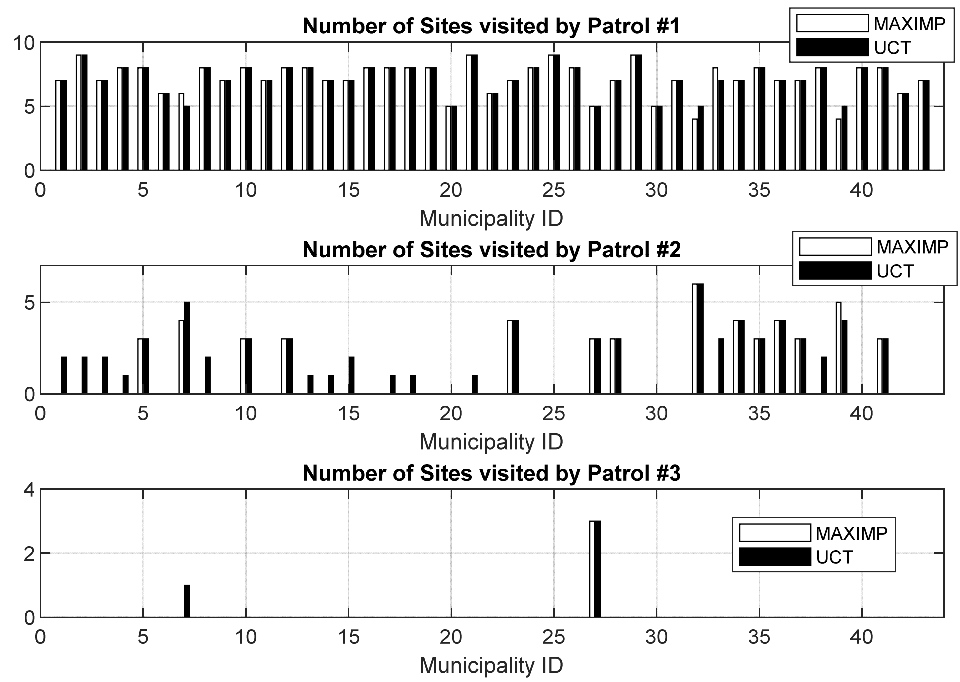

Figure 8: Number of waste sites visited by each patrol (MAXIMP and UCT comparison).

5 CONCLUSIONS

In this work some machine learning (clustering and Monte Carlo Tree Search) techniques are applied to optimize the use of resources for waste-sites inspections. More in detail, these techniques are used to perform an optimal multiple-tour assignment for the available patrols in order to globally maximize the number of visited sites that have the highest priority (which is related to their volume and the dangerousness of their waste).

The solutions obtained using the Monte Carlo Tree Search with upper confidence bounds (UCT) are compared to those obtained using a heuristic algorithm taken from literature (MAXIMP), often used to solve this kind of VRP. For the specific scenario here considered, the overall priority collected using UCT always matches or outperforms MAXIMP, with a better exploitation of the time interval given for the inspections. In addition, the UCT computational time (~1 min) enables also practical applications.

Besides the demonstrated performance, the other advantage of the presented approach is the fact that, once the considered problem is mapped into a game tree formalism, the UCT can operate effectively without any knowledge of the particular domain, apart from the game rules and the end conditions. Possible future improvements may focus on investigating modifications to the tree policy in the MCTS algorithm. This may improve its performance in finding optimal solutions as the state space grows (graphs with more nodes).

ACKNOWLEDGEMENTS

The proposed work has been developed within the Italian national funded project C4E "Crowd for Environment" from 2018 to 2021, aiming to develop a support decision system for the local authorities in the monitoring and removal of illegal waste. Founded by Italian Ministry of Research Grant Number RNA-COR 896199.

REFERENCES

[1] Mazza, A., Piscitelli, P., Neglia, C., Rosa, G. & Iannuzzi, L., Illegal dumping of toxic waste and its effect on human health in Campania, Italy. *International Journal of Environmental Research and Public Health*, **12**(6), pp. 6818–6831, 2015.

[2] Nebula, F., Gargiulo, F., Gigante, G., Pascarella, D. & Cicala, L., Multiple-tour constrained optimization for waste sites inspections. *International Conference on Optimization and Learning – OLA2021,* pp. 93–95, 2021.

[3] Butt, S.E. & Cavalier, T.M., A heuristic for the multiple tour maximum collection problem. *Computers and Operations Research,* **21**, pp. 101–111, 1994.

[4] Toth, P. & Vigo, D., An overview of vehicle routing problems. *The Vehicle Routing Problem*, eds P. Toth & D. Vigo, SIAM: Philadelphia, pp. 1–26, 2002.

[5] Vansteenwegen, P. & Souffriau, W., The orienteering problem: A survey. *European Journal of Operational Research*, **209**, pp. 1–10, 2011.

[6] QGIS Geographic Information System, QGIS Development Team. www.qgis.org.

[7] OpenStreetMap, OpenStreetMap contributors. www.openstreetmap.org. Accessed on: 18 Sep. 2021.

[8] Boeing, G., OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers, Environment and Urban Systems*, **65**, pp. 126–139, 2017.

[9] QNEAT3, QGIS Network Analysis Toolbox 3. root676.github.io. Accessed on: 28 Feb. 2022.

[10] McInnes, L., Healy, J., Astels, S., HDBSCAN: Hierarchical density based clustering. *Journal of Open Source Software*, **2**(11), 2017.

[11] McInnes, L. & Healy, J., Accelerated hierarchical density clustering. *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pp. 33–42, 2017.

[12] Browne, C., Powley, E., Whitehouse, D., Lucas, S., Cowling, P.I., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S. & Colton, S., A survey of Monte Carlo Tree Search methods. *IEEE Transactions on Computational Intelligence and AI in Games*, **4**(1), pp. 1–43, 2012.

[13] Auer, P., Cesa-Bianchi, N. & Fischer, P., Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, **47**, pp. 235–256, 2002.

[14] Mańdziuk, J. & Nejman, C., UCT-based approach to capacitated vehicle routing problem. *Artificial Intelligence and Soft Computing*, pp. 679–690, 2015.