A Distributed Online Heuristic for a Large-scale Workforce Task Assignment and Multi-vehicle Routing Problem

Diego Deplano, *Member, IEEE*, Carla Seatzu, *Senior, IEEE*, Mauro Franceschelli, *Senior, IEEE*

***Abstract*— This paper presents a novel formulation for a workforce routing, task assignment, and scheduling problem with privacy by design , drawing inspiration from multi-vehicle routing problems. We examine a real case study involving a large number of technicians tasked with refurbishing and repairing a large number of photo booth machines spread across a wide geographic area, spanning a country. We then introduce a novel heuristic distributed online optimization algorithm, based on gossiping, to: i) assign daily refurbishing and repair tasks to technicians; ii) plan optimal routes for each technician to execute the assigned tasks; iii) dynamically update task assignments and routes in real-time to accommodate delays and unforeseen impediments encountered by technicians (such as traffic jams). The objective is to maximize enterprise profit by effectively managing the workforce. The proposed method inherently safeguards the privacy of real-time geolocation data for the entire workforce, ensuring it remains undisclosed and inaccessible to the company's ICT infrastructure. We provide a numerical simulation utilizing real data, supplied by DEDEM S.p.A., demonstrating the performance of the proposed heuristic in terms of expected net profit for the company.**

I. Introduction

In this paper we consider one of the most complex scenarios of large-scale workforce management which can in part be modeled as a dynamic heterogeneous multi-vehicle routing problem (dynamic HMVRP).

The popular vehicle routing problem (VRP) [1], [2] is an extension of the famous Travelling Salesman Problem (TSP) [3] where a salesman (or vehicle) need to visit in minimum time or distance a set of locations or cities and go back to the starting point. In its general formulation the TSP is NP-hard, and several heuristic approaches with guarantees on the worst-case performance have been proposed [4]–[6]. In the VRP usually additional constraints are considered with respect to the TSP problem, such as capacity constraints for the vehicle, time windows of availability, etc. The VRP has received significant attention in the past decades and very efficient formulations and heuristic approaches exist to address its basic formulations: the study by Eksioglu, Vural, and Reisman [7] revealed more than 1 thousand journal articles with VRP as the main topic, published between 1959 and 2008, with an increasing rate of 6% every year.

Work supported by project e.INS- Ecosystem of Innovation for Next Generation Sardinia (cod. ECS 00000038) funded by the Italian Ministry for Research and Education (MUR) under the National Recovery and Resilience Plan (NRRP) - MISSION 4 COMPONENT 2, "From research to business" INVESTMENT 1.5, "Creation and strengthening of Ecosystems of innovation" and construction of "Territorial R& D Leaders".

Diego Deplano, Carla Seatzu, Mauro Franceschelli are with the Department of Electrical and Electronic Engineering, University of Cagliari, 09123 Cagliari, Italy (e-mails: diego.deplano@unica.it, seatzu@unica.it, mauro.franceschelli@unica.it). Mauro Franceschelli is corresponding author.

Fig. 1: A photo booth operated by DEDEM S.p.A.

In the MVRP the VRP is complicated by the existence of more than one vehicle which needs to be routed across the set of locations. This problem is generally more difficult to solve because not only the computation of each route is an NP-hard problem but also because the different sets of locations need to be assigned to different vehicles. Usually, in the MVRP identical vehicles are considered to simplify the problem and reduce the number of variables and constraints. Several variations on these problems have appeared recently such as [8], [9] for the pick and delivery in multi-robot systems, shared locations with collision-avoidance [10]–[13], or applications of multi-robot systems in precision agriculture [14]–[16] which employ ad hoc Mixed-Integer Linear Programming (MILP) formulations for fair workload distribution among robots and general optimization of the system behavior.

In the heterogeneous MVRP (HMVRP) each vehicle is characterized by different parameters such as motion speed, capacity etc., thus resulting in its most complex formulation which can be addressed for large problem instances only by heuristics [17], [18]. A way to address this issue is by employing the so-called gossip-based approaches where each vehicle or robot in the network updates its own task assignment and routing with a better one by exchanging tasks iteratively with one other vehicle at a time, i.e., *gossiping*. Gossiping has been successfully used to solve distributed load balancing [19]–[21] and task assignment problems [22] in large networks, and the HRMVRP also with performance guarantees and constant factor approximations of the objective value of the solution with respect to the optimal objective value exploiting the gossip-based interaction framework [18]. Additionally, when tasks and vehicles can enter and leave dynamically as real-time advances, the problem is called *dynamic*. Indeed, the optimal solution is now time-varying because every time there is a change in the set of tasks or set of vehicles, the problem changes and so does its solution.

In this work, we examine a real case study provided by the company DEDEM S.p.A, which requires managing a large workforce of technicians responsible for refurbishing and repairing numerous photo booths (see Fig. 1) across the entire Italian territory. The core of the problem involves a dynamic HMVRP. However, the various extensions and modifications necessary to practically address their workforce management issues make the scenario particularly challenging from a computational perspective. Considering its current workforce of approximately 1 thousand technicians, each tasked with $10 \div 20$ daily assignments distributed geographically, there are up to 20 thousand daily tasks to be addressed. Consequently, this necessitates the development of ad-hoc and novel heuristics. In particular, in our complex scenario, we consider multiple technicians departing from different depots and encountering tasks at each visited position, each with a specified duration and an hourly profit upon completion. Thus, the round trips to be found must maximize the net profit, factoring in both travel expenses and the revenue generated from task completion. Tasks are naturally prioritized based on their proximity to the technician's depot and their potential profitability. To address the company's requirements, we additionally incorporate the following features, further intensifying the complexity of the problem: (i) not all tasks are mandatory; instead, tasks should only be undertaken if they contribute to the net profit; (ii) tasks require specific skills for execution, which should be possessed by the assigned technician; (iii) tasks must be executed within specific time windows and also within the working hours of the technician that performs it, accounting for the time needed to return to their depot.

The above-described problem is threefold: 1) an assignment problem, because tasks must be assigned to the technicians; 2) a routing problem, because it must be provided the optimal route to visit the positions where the tasks must be performed; 3) a scheduling problem, because the time of the day in which tasks are executed changes the profit.

Lastly, we extend the problem horizon beyond a single day. This entails that, for a given task assignment, multiple routes along with their scheduling must be devised for each technician, spanning across multiple days. Summarizing, **the main contributions** are:

- A MILP formulation of the complex optimization scenario of the DEDEM S.p.A. case study, with a dynamic HMVRP at its core, extended to include all the practical details of the actual scenario.
- A real-time heuristic algorithm developed ad hoc for the case study characterized by randomization, scalability, and ability to dynamically reassign tasks and routes depending on delays on task execution. The heuristic is distributed and by design it does no exploit the real-time position of the technicians, thus preserving their privacy in the workplace.
- Numerical tests to validate the approach by using real data drawn from DEDEM S.p.A.'s data. It is shown how the proposed heuristic allows to improve the profit for the company up to 15% while dealing with undesired events such as delays and unexpected unavailability of the technicians.

Structure of the paper: Section II formalizes the problem at hand as a Mixed-Integer Linear programming, while Section III presents the proposed heuritic approach. Section IV provides an in-depth discussion of numerical simulations performed in a realistic scenario constructed from real data retrieved by DEDEM S.p.A., then Section V gives some concluding remarks.

II. A Mixed-Integer Linear Programming Formulation

Let $s \in \mathbb{N}$ be the number of technicians, $p \in \mathbb{N}$ be the number of positions to be visited by the technicians, each of which corresponds to a task to be executed, and $d \in \mathbb{N}$ be the number of days available for executing the tasks. Accordingly, we denote by $S = \{1, \ldots, s\}$ the set of technicians' starting positions which we call depots (we overload the notation $i \in S$ to also denote the technicial itself starting from the depot $i \in S$) , by $P = \{ds+1, \ldots, ds+p\}$ the set of tasks' positions, and by $D_\ell = \{\ell s+1, \ldots, (\ell+1)s\}$ the set of intra-day depots' position, one for each extra day $\ell = 1, \ldots, d-1$, which are needed to model the return of the technicians to their depots at the end of each working day: in other words, the ℓ-th night of the i-th technician is $\ell s+i \in D_\ell$. Note that for a single day $d = 1$ it holds $D_\ell = \emptyset$. Such a description of the sets facilitates a clear mathematical representation of the potential routes that the technicians may travel, as detailed next. Construct an undirected graph $G = (V, E)$ describing the viable paths $(i, j) \in E$ for each pair of locations $i, j \in V$:

$$V = S \cup D \cup P, \quad \text{where} \quad D = \bigcup_{\ell=1}^{d-1} D_\ell,$$

$$E : \begin{cases} (i,j),\ (j,i) \in E & i,j \in P, \\ (i,j),\ (j,i) \in E & i \in S \cup D, j \in P, \\ ((\ell-1)s+i, \ell s+i) \in E & i \in S,\ \ell = 1, \ldots, d-1, \\ (i,j) \notin E & \text{otherwise,} \end{cases} \tag{1}$$

where the first set of edges represent the paths between any pair of tasks, the second set of edges represent the paths between any technician's depot and any location, and the third set of edges represent the (fake) paths of each technician from one night to the next night. Associated with each technician $i \in S$ are

- a starting time $T_i^s \in [0, 24]$ (*hour*) and an ending time $T_i^e \in [0, 24]$ (*hour*) describing the time window $[T_i^s, T_i^e]$ within which the technicians can work;
- a fixed cost $q_i > 0$ (*eur*) that represents the cost of the technician for one day.

Associated with each position $i \in P$ are:

- a starting time $T_i^s \in [0, 24]$ (*hour*) and an ending time $T_i^e \in [0, 24]$ (*hour*) describing the time window $[T_i^s, T_i^e]$ within which the task in that position can be executed;
- a duration $\delta_i > 0$ (*hour*) denoting the amount of time needed to execute the task in that position;

- an hourly profit $g_i > 0$ (*eur/hour*).

Associated with each intra-day depot $i \in D_\ell$ are:

- a technician $i - \ell s \in S$;
- a time-window $[T_i^s, T_i^e] = [T_{i-\ell s}^e + 24(\ell-1), T_{i-\ell s}^s + 24\ell]$ (*hour*) within which the technician must go back;
- a duration $\delta_i = 24 - T_{i-\ell s}^e + T_{i-\ell s}^s$ (*min*) denoting the amount of resting time.

Associated with each viable path $(i,j) \in E$ are:

- a distance $d_{ij} \geq 0$ (*km*) between locations $i, j \in V$;
- a travel time $t_{ij} \geq 0$ (*hour*) between positions $i, j \in V$;
- a fuel cost $f_{ij} \geq 0$ (*eur/km*) for the path from $i \in V$ to $j \in V$, which takes into account the average speed on that path.

A wide variety of formulations exist for the standard TSP [23]. In the following, we adapt the multi-commodity flow formulation of Claus [24] – which we are going to call CMFC in the reminder of the manuscript – for the standard TSP to work with the more complex scenario considered in this paper, which consists of the following additional challenges:

- Multiple technicians starting from different depots;
- Time windows for both technicians and tasks;
- Hourly profit generated by the tasks' execution;
- Tasks may remain unassigned;
- Multi-day scheduling.

Due to this more complex scenario, both the objective functions and the constraints derived in this section constitute an original contribution of this manuscript.

Our choice to use the CMCF formulation as a base start for the development of the problem formulation in our scenario stems from the fact that it is equivalent (in terms of tightness of the LP relaxations) to the seminal formulation provided by Dantzig et. al [25], as shown in [23, Figure 2], it enjoys a polynomial number of constraints instead of an exponential number ([23, Table 1]), and it has a more convenient interpretation of the variables. To get the idea behind the CMCF formulation, one should imagine that when the technician passes through a position $k \in P$, a commodity is collected (think about a receipt) and it must be brought back to the depot. Thus, we have two kinds of variables:

- Boolean variables $X_{i,j} \in \{0,1\}$ denoting the motion of any technician from location $i \in V$ to location $j \in V$, such that $(i,j) \in E$.
- Continuous[1] variables $F_{i,j,k} \in [0,1]$ denoting the motion of the commodity $k \in P \cup D$ from position $i \in V$ to location $j \in P \cup D$, such that $(i,j) \in E$.

Differently from the standard Claus formulation, we introduce additional continuous variables $U_{i,j}$ to account for time windows and hourly profit generated by tasks' execution:

- Continuous variables $U_{i,j} \geq 0$ denoting the time at which the motion of any technician from location $i \in V$ to location $j \in V$ occurs, such that $(i,j) \in E$.

[1]Even though these variables will only take boolean values due to the subsequent set of constraints, declaring them continuous highly speed up the execution of the solver.

The cost function to be minimized consists of three terms: 1) The cost associated with the fuel consumed by the technicians during the scheduled trip:

$$\text{FUEL}(X) = \sum_{(i,j)\in E} f_{ij} d_{ij} X_{i,j}.$$

2) The cost associated with the technicians' activities:

$$\text{TECH}(F) = \sum_{i \in S} q_i \sum_{j \in P} F_{i,j,i} + \sum_{\ell \in 1}^{d-1} \sum_{i \in D_\ell} q_{i-\ell s} \sum_{j \in P} F_{i,j,i},$$

where the first term is the cost associated with the first day, and the second term is the cost associated with subsequent days.

3) The profit associated with the tasks' execution:

$$\text{TASKS}(X,U) = \sum_{j \in P} g_j \sum_{i \in V} \left[(24 \cdot d - t_{ij} - \delta_j) X_{i,j} - U_{i,j}\right].$$

Thus, the cost function to be minimized is given by:

$$f(X,F,U) = \text{FUEL}(X) + \text{TECH}(F) - \text{TASKS}(X,U). \quad (2)$$

We now introduce all the constraints that are needed to solve the problem under study.

Constraint 1: Each intra-day depot $k \in D$ must be visited:

$$\sum_{j \in V} X_{j,k} = 1, \quad \forall k \in D.$$

Constraint 2: Each position $i \in V$ is left at most once:

$$\sum_{j \in V} X_{i,j} \leq 1, \quad \forall i \in V.$$

Constraint 3: Each position $i \in V$ is entered if it is left:

$$\sum_{j \in V} (X_{i,j} - X_{j,i}) = 0, \quad \forall i \in V.$$

Constraint 4: Each commodity $k \in P \cup D$ can flow along the edge $(i,j) \in E$ only if the edge belongs to a tour:

$$0 \leq F_{i,j,k} \leq X_{i,j}, \quad \forall j \in V, \forall i,k \in P \cup D, : i \neq j.$$

Constraint 5: Each commodity $k \in P \cup D$ must leave its position if it belongs to some route:

$$\sum_{j \in V} (F_{k,j,k} - X_{k,j}) = 0, \qquad \forall k \in P \cup D.$$

Constraint 6: Each commodity $k \in P \cup D$ must return to a depot if it belongs to some route:

$$\sum_{j \in P \cup D} \sum_{i \in S} F_{j,i,k} \geq \sum_{j \in V} X_{j,k}, \quad \forall k \in P \cup D.$$

Constraint 7: Each commodity $k \in P \cup D$ that enters a location $i \in P \cup D$ (not a final depot) must also leave it:

$$\sum_{j \in P \cup D} F_{j,i,k} = \sum_{j \in V} F_{i,j,k}, \quad \forall i,k \in P \cup D.$$

Constraint 8: For each technician $i \in S$ going to position $k \in P \cup D$, commodity k must return to its depot:

$$X_{i,k} \leq \sum_{j \in P \cup D} F_{j,i,k}, \quad \forall i \in S, \forall k \in P \cup D.$$

Constraint 9: Each technician $i \in S$ must leave its depot within its time window:

$$T_i^s X_{i,k} \le U_{i,k} \le (T_i^e + 24(d-1)) X_{i,k}, \quad \forall i \in S, \forall k \in P \cup D.$$

Constraint 10: Each technician must return to its depot $i \in S$ within its time window:

$$T_i^s X_{i,k} \le U_{k,i} \le (T_i^e + 24(d-1) - t_{k,i}) X_{i,k}, \quad \forall i \in S, \forall k \in P \cup D.$$

Constraint 11: Each task's position $k \in P$ must be visited within its time-window:

$$(T_k^s - t_{i,k}) X_{i,k} + 24 \sum_{\ell \in D} F_{i,k,\ell} \le U_{i,k}, \quad \forall i \in V, \forall k \in P,$$
$$(T_k^e - t_{i,k} - \delta_k) X_{i,k} + 24 \sum_{\ell \in D} F_{i,k,\ell} \ge U_{i,k}, \quad \forall i \in V, \forall k \in P.$$

Constraint 12: Each intra-day depot $k \in D$ must be executed within its time-window:

$$(T_k^s - t_{i,k}) X_{i,k} \le U_{i,k}, \quad \forall i \in V, \forall k \in D.$$
$$(T_k^e - t_{i,k} - \delta_k) X_{i,k} \ge U_{i,k}, \quad \forall i \in V, \forall k \in D.$$

Constraint 13: Each position $k \in P \cup D$ must be executed after the previous position $i \in P \cup D$ has been reached and the task has been completed:

$$\sum_{j \in V} U_{k,j} \ge \sum_{j \in V} \left(U_{j,k} + (t_{j,k} + \delta_k) X_{j,k} \right), \qquad k \in P \cup D.$$

Constraint 14: Task in position $k \in P$ cannot be executed by technician $j \in S$:

$$\sum_{i \in P \cup D} F_{i,j,k} = 0, \qquad \forall j \in S, \forall k \in P.$$

Constraint 15: Technician $j \in S$ is not available on day ℓ:

$$X_{p,q} = 1, \quad \text{where} \quad \begin{cases} p &= (\ell - 1)s + i, \\ q &= (\ell \,\%\, d)s + i, \end{cases} \text{ and } \ell = 1, \dots, d,$$

where $a \,\%\, b$ returns the reminder after the division a/b.

Thus, the optimization problem reads as:

$$\begin{aligned} &\underset{X,F,U}{\text{argmin}} \;\; f(X, F, U) \\ &\text{s.t.} \quad \text{Constraints 1-15} \end{aligned} \tag{3}$$

III. A GOSSIP-BASED HEURISTIC ALGORITHM

In this section, we describe a heuristic sub-optimal approach to solve the problem detailed in Section II when the complete problem becomes too large to be solved optimally. The proposed approach can be used both offline, to compute an initial solution to the problem, and online, to continue improving the initial solution while taking into account the following realistic occurrences:

- Delays due to traffic jams or unexpected complications during the execution of the tasks or sudden unavailability of some technicians: in this case, the algorithm is able to re-arrange the schedules of the technicians in order to minimize the loss;
- Availability of new technicians or new tasks: in this case, the algorithm is able to compute new solutions in real time in order to increase the profit.

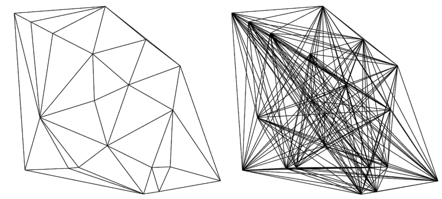

Fig. 2: Graphical comparison between the Delaunay graph (left) and the complete graph (right) of a given set of points.

The proposed approach, detailed in Algorithm 1, is based on *gossiping*, a distributed computing technique used to disseminate information efficiently across a network of nodes. In this scenario, we let the nodes represent the technicians and let the edges between nodes represent possible pairs of technicians for which a local optimal solution can be seek. Algorithm 1 implements an edge-based gossiping technique by picking at random one of these edges (i.e., selecting a pair of technicians) and computing a local optimal solution by considering the tasks already assigned to them, if any, together with some unassigned tasks. Once the optimal solution is found, it moves to another pair of technicians, until it is not possible anymore to improve any of the schedule for each pair of technicians. In so doing, one can be sure that when Algorithm 1 ends, it is not possible anymore to take a pair of technicians and improve their schedules. This approach is quite efficient for mainly two reasons: 1) each local optimization problem is much easier to solve if compared to the global problem; 2) local optimizations involving different technicians can be run simultaneously in parallel.

Since the number of possible pairs in a set of $n \in \mathbb{N}$ elements is given by $n(n-1)/2$, i.e., it increases quadratically with n, we decide to performe the above described gossip technique over a subset of all these pairs by exploiting their geographical distribution. We do so by means of the so-called "*Delaunay graph*" [26], [27].

Definition 1: *Consider a set of points scattered over a geometric space and let $G = (V, E)$ be a graph where V is the set of nodes representing the points and E is a set of edges connecting them. The graph G is the (unique) Delaunay graph of points V if the edges in E form a triangulation such that no point in V is inside the circumcircle of any triangle formed by the edges of E.*

The Delaunay graph provides a nice connectivity structure among the input points because for any given node, its incident edges are connected to the nodes that are closer to it than any other nodes in the set. More importantly, the maximum number of edges in a Delaunay graph is equal to $3n - 6$ for $n \ge 3$, i.e., it increases linearly with the number of nodes, which scales much better than a complete graph with a number of edges equal to $n(n-1)/2$, which increases quadratically with the number of edges (see Fig. 2).

In Algorithm 1, the Delaunay graph is constructed based on the technicians' current tasks, thereby circumventing in-

Algorithm 1 Online Task Assignment and Scheduling with Multi-Vehicle Routing

Input: The sets of technicians S, days D, and tasks P
The initial Delaunay graph $G_D = (S, E_D)$
The number $m \in \mathbb{N}$ of unassigned tasks to be included in the local optimizations

Initialize: Mark all edges as active $E_A = E_D$

1: **repeat** indefinitely steps 2-24:
2: **while** $E_A = \emptyset$ **do**
3: **if** technician $i^\star$ is late or changes task
4: Mark incident edges of agent $i^\star$ as active: $E_A = E_A \cup \{(i,j) \in E_D \mid i = i^\star \text{ or } j = i^\star\}$
5: **end if**
6: **end while**
7: Update the positions of technicians based on the tasks they are currently executing (if any)
8: Keep the memory of previous edges: $E_D^- = E_D$
9: Construct the new Delaunay Graph: $\mathcal{G}_D = (S, E_D)$
10: Mark old edges as inactive: $E_A = E_A \cap E_D$
11: Mark new edges as active: $E_A = E_A \cup (E_D \setminus E_D^-)$
12: Pick an edge $(i^\star, j^\star)$ at random from E_A
13: Solve the local optimization problem (3) where: (i) only technicians $i^\star$ and $j^\star$ are considered (ii) only tasks currently assigned to technicians $i^\star$ or $j^\star$ are considered, along with $m \in \mathbb{N}$ randomly picked unassigned tasks in P (iii) the scheduling of the current and previous tasks must remain unchanged
14: **if** the new solution improves the objective of (3)
15: **if** agent $i^\star$ changes assignment
16: Mark its incident edges as active: $E_A = E_A \cup \{(i,j) \in E_D \mid i = i^\star \text{ or } j = i^\star\}$
17: **end if**
18: **if** agent $j^\star$ changes assignment
19: Mark its incident edges as active: $E_A = E_A \cup \{(i,j) \in E_D \mid i = j^\star \text{ or } j = j^\star\}$
20: **end if**
21: **return** the new assignment, schedule, and routing for technicians $i^\star$ and $j^\star$
22: **end if**
23: Mark edge $(i^\star, j^\star)$ as inactive $E_A = E_A \setminus \{(i^\star, j^\star)\}$
24: Update the sets of the currently available technicians S and tasks P

trusive real-time geolocation and ensuring technician privacy. Specifically, the initial geographic position of the technicians correspond to their assigned depot, which is subsequently updated to reflect the location of the task they are currently executing. Consequently, during the real-time execution of Algorithm 1, the Delaunay graph must be recomputed every time one or more of the technicians change tasks or when is late on schedule, thus allowing re-assignemnt of the tasks to reduce profit losses. In order to deal with this dynamic scenario, we mark the edges of the real-time Delaunay graph as "*active*" or "*inactive*" to denote whether the local optimization between the corresponding incident nodes has been already performed or not. Algorithm 1 implements the following logic:

- All edges of the Delaunay graph are initially active.
- If the local optimization associated with an edge (i, j) improves the previous solution by changing the schedule of one of the technicians, then all edges incident to nodes i and/or j are marked as active. In any case, edge (i, j) is marked as inactive at the end of the optimization.
- If the Delaunay graph changes, then all remaining edges keep their marking, while new edges are marked as active.
- If an agent i is late on schedule or changes tasks, the edges incident on node i are marked as active.

This logic ensures that there exists a finite number of steps after which the execution of Algorithm 1 stops because all edges will eventually be marked as inactive.

IV. Numerical simulation: a case study by DEDEM S.p.A.

In this section, we discuss a numerical simulation of the proposed Algorithm 1 using real data provided by DEDEM S.p.A., an international company with a long history in the passport photo sector, starting in 1962 with the installation of the first photo booth machine in Rome, Italy. We have collected anonymized data of 16 technicians' tours in a working week of 5 days for a total of about 1000 tasks to be executed across 5 Italian regions (Lazio, Campania, Toscana, Umbria, Abruzzo). We provide next a high-level description of the real parameters collected by DEDEM S.p.A. that we used to formalize our set-up as described in Section II:

- Technicians start working at 7 : 00 AM ($T_i^s = 7$) and finish working at 6 : 30 PM ($T_i^e = 18.5$ for all $i \in S$);
- The cost of technicians for one day of work is € $50, 00$ ($q_i = 50$ for all $i \in S$);
- Tasks may be executed at any time ($T^s = 7$ and $T_i^e = 18.5$ for all $p \in P$);
- The duration of the tasks ranges from 1 minute to 5 hours ($d_p \in [0.02, 5]$ for all $p \in P$), while the average duration is about 22 minutes;
- The profit of the tasks ranges from € 0.50 to € 11.40 per hour ($g_p \in [0.5, 11.4]$ for all $p \in P$), while the average profit is about € 1.80;
- The area containing all tasks is contained within a squared area of side length of 250 km;
- Travel times have been computed by means of Google Maps APIs, while the fuel cost has been considered fixed for all roads and equal to about € 0.10 per kilometer ($f_{ij} = 0.10$ for any $i, j \in V$).

The actual routes taken by the technicians are illustrated in Fig. 3, encompassing 814 tasks (approximately 88% of the total available) completed over a span of 5 workdays. The expected profit, computed as per Equation (2) over a complete week (considering also the weekend), amounts to € $281.695, 14$. A closer inspection of Fig. 3 reveals that the company currently assigns tasks to technicians based

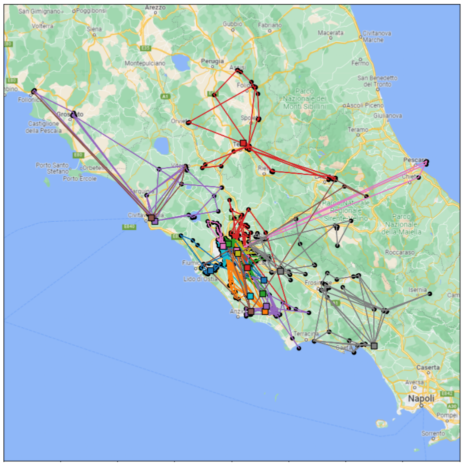

Fig. 3: Real routes retrieved from DEDEM S.p.A.'s data. Some tasks are not executed due to time window constraints.

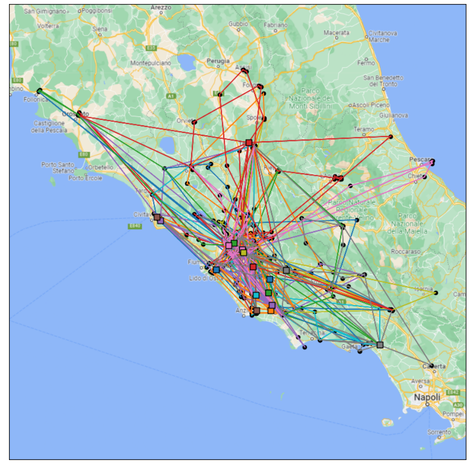

Fig. 4: Routes obtained by Algorithm 1. Improved routing and assignments allow more tasks to be executed.

on their geographic proximity. Specifically, in the case of DEDEM S.p.A., the tasks under consideration involve the refurbishment and repair of photo booth machines situated at fixed geographical locations. Each technician is exclusively responsible for tasks associated with machines assigned to them, precluding them from working on other machines.

A. *Heuristic offline solution*

The complete optimization problem formulation, when considering all 16 technicians and all 932 tasks, becomes really hard to solve since it consists of about half billion variables (half million are boolean). Instead, when implementing Algorithm 1, the problem's complexity decreases significantly. This is due to the fact that only 2 technicians and a subset of the total tasks are considered. Initially, at the point where none of the tasks are assigned and only $m = 10$ unassigned tasks are taken into account, the optimization problem consists of:

- $\sim$ 6 thousand variables (200 are boolean);
- $\sim$ 12 thousand constraints.

As Algorithm 1 progresses through subsequent iterations, more tasks are incorporated into the problem. However, it remains constrained to only 2 technicians and a fraction of the total tasks. This is because tasks are gradually assigned to technicians in a distributed manner during execution. Consequently, the initial local optimizations are comparatively simpler to solve optimally, while complexity increases as more tasks are assigned to technicians. Indeed, at the end of the process, on average, the local optimizations consist of:

- $\sim$ 1.5 million variables (13 thousand are boolean);
- $\sim$ 3 million constraints.

Nevertheless, the solver benefits from the previous solution, which serves as a warm start, speeding up the algorithm.

We have implemented Algorithm 1 in Python programming language, utilizing Gurobi 11.0 (with Academic Licence) as the optimization solver. We simulate the execution of Algorithm 1 during the weekend to compute a solution for the next working week. We impose the following constraints on each local optimization:

- an optimality gap of 10%;
- a time limit of 15 minutes;
- an horizon time of 5 days.

The algorithm ran for a total of about 36 hours, from Saturday at 1 AM to Sunday at 1 PM. The routes of the heuristic solution are illustrated in Fig. 4, encompassing 832 tasks (approximately 90% of the total available) completed over a span of 5 workdays. The expected profit, computed as per Equation (2) over a complete week (considering also the weekend), amounts to € $326.318, 66$. When compared with the real routes, the solution found by the proposed heuristic allows to perform 18 more tasks ($> 2\%$) with an extra profit of € $44.623, 52$ ($> 15\%$).

B. *Heuristic online solution*

We now consider the scenario in which an unexpected event does not allow the normal execution of the tasks in the schedule and show how the online execution of Algorithm 1 allows to mitigate the negative effects of the event, i.e., it allows to reduce the loss of profit for the company. The unfortunate event taken into consideration is the sudden unavailability of one technician on the first day, to which were assigned 19 tasks, which would cause a profit loss of € $11.094, 52$ if no action is taken. We simulate the real-time execution of Algorithm 1 during the first day of the workweek, where the 19 tasks originally assigned to the technician who is no longer available are now available for assignment to other technicians, provided it results in increased profit. The objective is that of trying to re-assign these tasks to other technicians during the working day. We impose the following constraints on each local optimization performed by the solver:

- an optimality gap of 0%;
- a time limit of 2 minutes;
- an horizon time of 2 days.

With these parameters, the algorithm ran for approximately 2 hours – i.e., at least 60 local optimizations – assigning 17 out of 19 tasks. In other words, the algorithm ran at the same time the technicians were executing their tasks and, progressively, exchanged or added tasks to the technicians by modifying their future schedule. After the first two hours, it continued to optimize the routes for the remainder of the day without assigning any additional tasks but still changing them if convenient. At the end of the day, the profit loss is € $1.055,41$, less than 10% of the maximum possible loss in the case no action would have been taken. This saved loss can be attributed to two main factors: 1) the reassignment of tasks for an employee who became suddenly unavailable on the first day; 2) the local optimization between new pairs of technicians resulting from the time-varying Delaunay graph constructed throughout the day, which is informed by their latest task executions.

V. Conclusions

This paper presents a distributed heuristic algorithm designed to address a problem encompassing task assignment, route planning, and scheduling for a large workforce. The algorithm accounts for unique complexities arising from a real-world case study provided by DEDEM S.p.A., aimed at maximizing profit. It operates in real-time during tasks execution, automatically adjusting scheduling in response to new tasks, technician unavailability, or delays. Numerical simulations validate the effectiveness of the proposed approach, demonstrating improved expected profit compared to the company's current scheduling and highlighting its ability to mitigate potential profit loss.

Future work will focus on characterizing the scalability of the approach with respect to workforce size and task number, and exploring the introduction of machine-learning in the light of [28], [29].

Acknowledgments: We gratefully acknowledge and thank DEDEM s.p.a. and Federico Lama for the data and support.

References

[1] G. Laporte, "Fifty years of vehicle routing," *Transportation Science*, vol. 43, no. 4, pp. 408–416, 2009.

[2] K. Braekers, K. Ramaekers, and I. Van Nieuwenhuyse, "The vehicle routing problem: State of the art classification and review," *Computers & industrial engineering*, vol. 99, pp. 300–313, 2016.

[3] K. L. Hoffman, M. Padberg, G. Rinaldi, *et al.*, "Traveling salesman problem," *Encyclopedia of operations research and management science*, vol. 1, pp. 1573–1578, 2013.

[4] N. Christofides, "Worst-case analysis of a new heuristic for the travelling salesman problem.," DTIC Document, Tech. Rep., 1976.

[5] L. Few, "The shortest path and the shortest road through n points," *Mathematika*, vol. 2, no. 2, pp. 141–144, 1955.

[6] H. J. Karloff, "How long can a Euclidean traveling salesman tour be?" *SIAM Journal on Discrete Mathematics*, vol. 2, no. 1, pp. 91–99, 1989.

[7] B. Eksioglu, A. V. Vural, and A. Reisman, "The vehicle routing problem: A taxonomic review," *Computers & Industrial Engineering*, vol. 57, no. 4, pp. 1472–1483, 2009.

[8] A. Camisa, A. Testa, and G. Notarstefano, "Multi-robot pickup and delivery via distributed resource allocation," *IEEE Transactions on Robotics*, vol. 39, no. 2, pp. 1106–1118, 2022.

[9] L. Abbatecola, M. P. Fanti, G. Pedroncelli, and W. Ukovich, "A distributed cluster-based approach for pick-up services," *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 2, pp. 960–971, 2018.

[10] D. Deplano, S. Ware, R. Su, and A. Giua, "A heuristic algorithm to optimize execution time of multi-robot path," in *2017 13th IEEE International Conference on Control & Automation (ICCA)*, IEEE, 2017, pp. 909–914.

[11] S. Karaman and E. Frazzoli, "Vehicle routing problem with metric temporal logic specifications," in *2008 47th IEEE conference on decision and control*, IEEE, 2008, pp. 3953–3958.

[12] C. I. Vasile and C. Belta, "An automata-theoretic approach to the vehicle routing problem.," in *Robotics: Science and Systems*, Berkeley, CA, 2014, pp. 1–9.

[13] D. Deplano, M. Franceschelli, S. Ware, S. Rong, and A. Giua, "A discrete event formulation for multi-robot collision avoidance on pre-planned trajectories," *IEEE Access*, vol. 8, pp. 92 637–92 646, 2020.

[14] M. Lippi and A. Marino, "A mixed-integer linear programming formulation for human multi-robot task allocation," in *2021 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN)*, 2021, pp. 1017–1023.

[15] M. Lippi, J. Gallou, A. Gasparri, and A. Marino, "An optimal allocation and scheduling method in human-multi-robot precision agriculture settings," in *2023 31st Mediterranean Conference on Control and Automation (MED)*, 2023, pp. 541–546.

[16] A. Furchì, M. Lippi, R. F. Carpio, and A. Gasparri, "Route optimization in precision agriculture settings: A multi-steiner tsp formulation," *IEEE Transactions on Automation Science and Engineering*, vol. 20, no. 4, pp. 2551–2568, 2023.

[17] S. Chopra and M. Egerstedt, "Heterogeneous multi-robot routing," in *IEEE American Control Conference*, 2014, pp. 5390–5395.

[18] M. Franceschelli, D. Rosa, C. Seatzu, and F. Bullo, "Gossip algorithms for heterogeneous multi-vehicle routing problems," *Nonlinear Analysis: Hybrid Systems*, vol. 10, pp. 156–174, 2013.

[19] M. Franceschelli, A. Giua, and C. Seatzu, "Load balancing on networks with gossip-based distributed algorithms," in *46th IEEE Conference on Decision and Control*, 2007, pp. 500–505.

[20] M. Franceschelli, A. Giua, and C. Seatzu, "A gossip-based algorithm for discrete consensus over heterogeneous networks," *IEEE Transactions on Automatic Control*, vol. 55, no. 5, pp. 1244–1249, 2010.

[21] E. Gravelle and S. Martínez, "An anytime distributed load-balancing algorithm satisfying capacity and quantization constraints," *IEEE Transactions on Control of Network Systems*, vol. 4, no. 2, pp. 279–287, 2015.

[22] M. Franceschelli, A. Giua, and C. Seatzu, "Fast discrete consensus based on gossip for makespan minimization in networked systems," *Automatica*, vol. 56, pp. 60–69, 2015.

[23] T. Öncan, I. K. Altınel, and G. Laporte, "A comparative analysis of several asymmetric traveling salesman problem formulations," *Computers & Operations Research*, vol. 36, no. 3, pp. 637–654, 2009.

[24] A. Claus, "A new formulation for the travelling salesman problem," *SIAM Journal on Algebraic Discrete Methods*, vol. 5, no. 1, pp. 21–25, 1984.

[25] G. Dantzig, R. Fulkerson, and S. Johnson, "Solution of a large-scale traveling-salesman problem," *Journal of the operations research society of America*, vol. 2, no. 4, pp. 393–410, 1954.

[26] D. P. Dobkin, S. J. Friedman, and K. J. Supowit, "Delaunay graphs are almost as good as complete graphs," *Discrete & Computational Geometry*, vol. 5, pp. 399–407, 1990.

[27] J.-D. Boissonnat, O. Devillers, and S. Hornus, "Incremental construction of the delaunay triangulation and the delaunay graph in medium dimension," in *Proceedings of the twenty-fifth annual symposium on Computational geometry*, 2009, pp. 208–216.

[28] D. J. Hoitomt, P. B. Luh, and K. R. Pattipati, "A practical approach to job-shop scheduling problems," *IEEE transactions on Robotics and Automation*, vol. 9, no. 1, pp. 1–13, 1993.

[29] A. Liu, P. B. Luh, K. Sun, M. A. Bragin, and B. Yan, "Integrating machine learning and mathematical optimization for job shop scheduling," *IEEE Transactions on Automation Science and Engineering*, 2023.