This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

A Decision Support Approach for Postal Delivery and Waste Collection Services

Lorenzo Abbatecola, Maria Pia Fanti, *Senior Member, IEEE*, Agostino Marcello Mangini, and Walter Ukovich, *Member, IEEE*

***Abstract*— This paper presents an urban-decision support system (U-DSS) devoted to manage, in a unified framework, the logistic services of the smart cities, such as postal delivery (PD) and waste collection (WC) services. The U-DSS architecture is proposed by describing its main components. In particular, this paper focuses on the core of the U-DSS, i.e., the model component that provides the solutions of a general vehicle assignment and routing optimization problem with the aim of minimizing the length of the routes and satisfying time and capacity constraints. In order to solve the vehicle routing problems in reasonable time, a two-phase heuristic algorithm is proposed based on a clustering strategy and a farthest insertion heuristic for the solution of a traveling salesman problem. The applicability of the proposed U-DSS is enlightened by comparing the proposed heuristic algorithm solutions with the mixed integer linear programming problem solutions of the PD and WC services. Moreover, the discussion of the real case studies of the city of Bari (Italy) assesses the proposed approach.**

***Note to Practitioners*—The motivation of this paper is to design an urban-decision support system (U-DSS) for use by decision makers (DMs) in the offline optimal planning of the vehicle assignment and routing in different services, such as postal delivery and waste collection. The U-DSS provides solutions to the DMs with threefold important key features. First, the modeled and solved vehicle routing problem is general and can consider several landfills, set of vehicles with different capacities and speeds, several shifts and routes satisfying capacity, and time constraints. Second, the traveling times are obtained by the routes and the distances are determined by Google Map application programming interface, the traffic, and weather forecasts. Third, the solutions are obtained in extremely short times, even if the dimensions of the problem are very large. Future research aims at extending the U-DSS by considering dynamic routing and modifying in real time the planning of the routes with a suitable use of the modern information and communication technologies.**

***Index Terms*— Decision support system (DSS), heuristic algorithms, logistics, vehicle routing problem (VRP).**

Manuscript received November 2, 2015; revised February 21, 2016; accepted May 10, 2016. This paper was recommended for publication by Associate Editor M. Kay and Editor S. Reveliotis upon evaluation of the reviewers' comments.

L. Abbatecola and W. Ukovich are with the Department of Engineering and Architecture, University of Trieste, Trieste 34128, Italy (e-mail: lorenzo.abbatecola@phd.units.it; ukovich@units.it).

M. P. Fanti and A. M. Mangini are with the Department of Electrical and Information Engineering, Polytechnic of Bari, Bari 70126, Italy (e-mail: mariapia.fanti@poliba.it; agostinomarcello.mangini@poliba.it).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASE.2016.2570121

I. Introduction

One emerging and challenging issue in smart cities is the city logistics. Every urban dedicated management system, such as traffic management, waste collection (WC), and postal delivery (PD), is very complex and socially and environmentally sensitive: it requires dedicated and intelligent support from the decision making process point of view. The increasing availability of information and communication technologies (ICTs) allows the development of models and novel decision making strategies to optimize the information process and utilization for the decision makers (DMs).

Among the most critical services to be managed in the smart cities, the PD and WC services are considered core problems in the city logistics [1]. They are often modeled as vehicle routing problems (VRPs) that are largely addressed in the related literature (see, for instance, among the others the review papers [1]–[4].

In particular, the management of the PD system involves locations, network design, vehicle routing, and crew assignment [5]. Multidepot vehicle routing in city VRP is considered by [6] and [7] that study a VRP and crew scheduling problem for the Australian post. However, the authors deal with just some case studies by using a heuristic algorithm. Reference [8] proposes a hybrid method for route generation based on the automatic developing of the delivery scheduling system. The method combines experts know-how and various optimization algorithms, such as genetic algorithms. In addition, Dong and Xiang [9] address the VRP in the mail delivery context: they propose a hybrid algorithm that combines the ant colony optimization metaheuristics with two local optimization heuristics. Reference [10] considers the problem of borough postal transportation and propose a lexicographic order linear programming model to minimize transportation costs based on the minimization of fixed costs: an optimal solution is derived using a genetic algorithm. In a successive work [5], the authors construct a postal express mail network that can realize the fastest transfer of mails with the lowest cost: the problem is solved by a swarm intelligent scheme. In [11], a modular decision support framework is proposed in order to support fleet and logistics managers in the reporting of routing decision, based on shortest path criterion. However, the authors focus on a scenario made of a single vehicle multistops routing problem.

The WC routing problem (WCRP) is one of the critical management problems to be solved in order to obtain an efficient and quick WC. It concerns the optimal design of routes to be performed by a fleet of vehicles with limited capacities to serve a set of customers. In the field of the WCRP, different approaches and strategies are presented in order to deal with the complexity and the large dimensions of the real systems (see, for instance, [1] for a complete and discussed review): 1) genetic algorithms [12]; 2) heuristic algorithms based on arc or node routing problems [13]; 3) swarm intelligence methods [14], [15] and ant heuristics [16]; and 4) simulation approaches [17]. In particular, [12] develops methodologies based on genetic algorithms to find the best route for collecting solid waste in cities. In addition, Singh *et al.* [13] deal with a capacitated arc routing problem and consider several routes assigned to each truck. Reference [16] proposes an ant colony heuristic, based on a node routing problem, to determine the solution of an urban WC problem in the municipality of Sant Boi de Llobregat, within the metropolitan area of Barcelona (Spain). However, the authors assume that each route starts and ends at the depot and no time constraints are considered. On the other hand, [17] presents an innovative modeling framework based on simulation to validate the solution of a WC problem. In addition, [18] proposes an ant colony optimization algorithm that considers time window constraints and several landfills: the authors analyze a WCRP considering a set of bins to be served, a number of facilities where the collected waste must be disposed off, and an infinite number of vehicles stationed in a single depot. Furthermore, a set of contributions considers time window constraints [14], [15], [18], [19], and a more complex problem complicated by several time windows and interarrival time constraints at each customer point is addressed in [20].

Besides the large number of contributions about the VRP problems presented in the related literature, a few authors propose a decision support approach able to deal with various types of urban logistics problems. An intelligent decision support system (DSS) is described in [21] to support the management of urban infrastructures, such as waterworks and sewage. The system defines the requirements and functionalities developed to improve the delivery, performance, and coordination of municipal services. Reference [13] proposes a DSS with several components, including one for prediction of waste amounts and one for arc routing of trucks. Reference [22] presents a user-friendly Web-based spatial DSS aimed at generating optimized vehicle routes for multiple VRPs that involve serving the demand located along the arcs of a transportation network. The DSS incorporates Google Maps, a database, a heuristic, and an ant-colony metaheuristic developed to generate routes and detailed individual vehicle route maps. However, the proposed DSS is devoted only to the WC and optimizes separately each town district and does not face the overall city system.

Hence, the development of efficient solution methods for city logistics that can include traffic data, fast response, and ITS tools is an open and huge problem.

This paper aims at providing contributions in this context and presents an urban-DSS (U-DSS), devoted to manage in a unified framework the PD and WC services. The U-DSS architecture is proposed by describing its main components: the data component (DC), the interface component (IC), and the model component (MC).

In particular, the proposed U-DSS provides the optimal routes for a general vehicle assignment and routing (VAR) problem to minimize the length of the routes and satisfy time and capacity constraints. It is well known that these kinds of problems are NP-hard and can be solved in reasonable time only if the dimension of the system is quite limited. Hence, we solve the considered VAR problems by a two-phase heuristic algorithm: the first phase is based on a clustering strategy and the second phase is based on a farthest insertion heuristic for the solution of the traveling salesman problem (TSP).

The proposed general algorithm is specified for the PD and WC services. Moreover, in this paper, we present the mixed integer linear programming (MILP) models of the two services: comparing the algorithm solutions with the optimal solutions provided by the MILP, we show that the heuristic gives good solutions and can be used to solve large and complex problems in short time.

In order to show the applicability of the proposed U-DSS, we consider the PD and the WC services of the city of Bari (Italy).

This paper is organized as follows. Section II introduces the U-DSS architecture and Section III describes in detail the VAR module of the MC by proposing a general heuristic algorithm. Moreover, Sections IV and V present two instances and customized applications of the VAR module for the PD and WC services, respectively: two case studies enlighten the U-DSS applicability. Finally, Section VI discusses the concluding remarks.

II. Urban-Decision Support System Architecture

In this section, the U-DSS architecture and its main components are briefly described. The U-DSS is developed following user-centered [23] and model-based [24] approaches: the first activity consists on the identification of the user needs and requirements through a series of interviews; then, the second activity consists on building the U-DSS modules giving emphasis on mathematical models and optimization techniques.

The structure of the proposed U-DSS consists of three main components: the DC, the MC, and the IC. Fig. 1 shows a simple scheme of the U-DSS architecture by enlightening the connections among them.

The DC collects the data and the information necessary to the U-DSS services: historical data and real-time data. In the cases of the VAR problems, the historical database collects the sites to be served, the demand of each site, the number and the dimension of the vehicles, and the time length of the shift. The real-time data are provided by the IC: real-time information is received by the modern ICT tools about the traffic, the weather conditions, peaks of the demand, and unusual situations.

Combining historical data with real-time data makes it possible to refine inputs for the next decisions or select inputs that better fit with a particular scenario. Real-time information, for instance, travel times along routes detected by GPS, can be stored in the DC in order to update the historical data

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

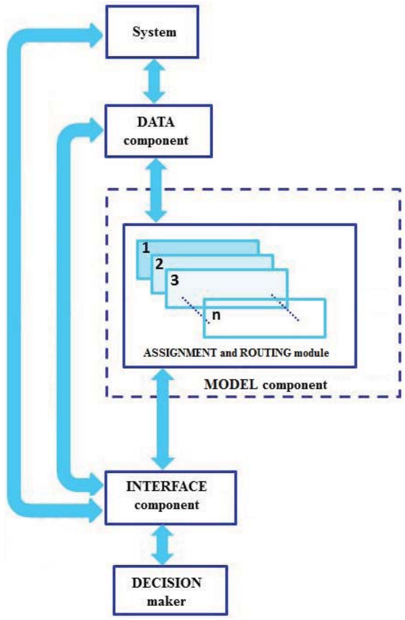


Fig. 1. Scheme of the U-DSS architecture.

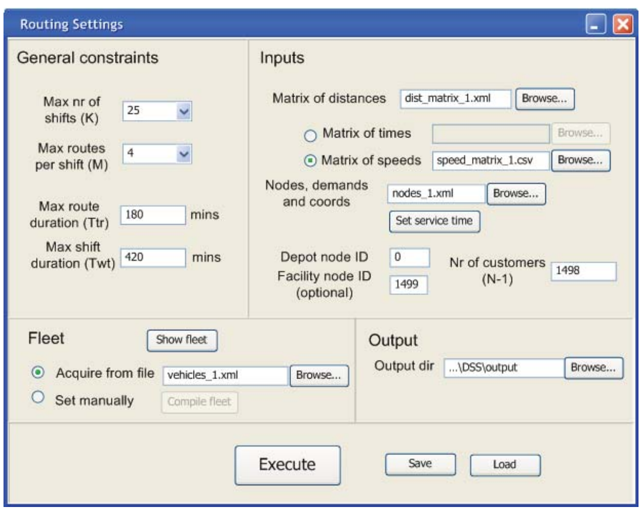


Fig. 2. IC input display.

and forecast more suitable system parameters for different scenarios.

On the other hand, the MC is the core of the U-DSS and contains the models, algorithms, and rules needed to provide decision support for the users. In particular, in the U-DSS, the MC is composed by a module dedicated to solve the VAR problems. In Section III, the module is described in detail by proposing a general heuristic algorithm that can be customized to be applied to different services based on VRPs.

Finally, the IC allows the effective interaction of the user with the real system: it is responsible of the communication and interaction among the MC, the DC, the real world, and the users. The IC main tasks are: 1) setting the information needed for the MC to operate and determine the optimal solution; 2) displaying the solutions provided by the MC; and 3) connecting the U-DSS with the real-time data sources. For instance, Fig. 2 shows the panels that the user has to fill in order to provide the necessary information, and Fig. 3 shows some possible U-DSS text and graphic outputs of the IC.

Shift	Route	Route duration (mins)	Route length (km)	Load
1	1	173,57	47,85	33
	2	179,54	53,49	35
2	1	179,23	54,49	44
	2	173,21	41,81	55
3	1	174,53	53,78	49
	2	176,76	47,60	60
4	1	176,86	36,97	86

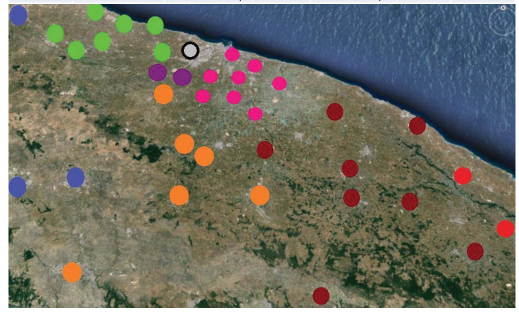


Fig. 3. IC solution display.

For application in real-time context, U-DSS receives unpredictable events, such as street interruptions, accidents, blocking, and so on, and the input data of the MC can be modified accordingly. Two cases may occur: 1) the event happens before the daily service; hence a new solution of the complete VAR problem is determined and 2) the event happens when some shifts have already started; then, the system description is modified in order to set up the new scenario by considering only the not yet traveled routes.

III. Vehicle Assignment and Routing Module

In this section, the MC of the proposed U-DSS is described: it considers the general problem of optimizing the routes of a set of vehicles that have to visit a set of customers by minimizing the total traveled distance, under capacity and time constraints. Our effort is focused on presenting a general model that can provide solutions for different urban logistics problems: suitable instances and customized applications of the proposed module can be realized in the MC.

A. Vehicle Assignment and Routing Problem Description

Let us consider a set of vehicles U that operate collection or delivery activities subject to time and capacity constraints. During a work shift, each vehicle can follow several routes that may be limited by a maximum of M routes per shift. In order to associate a shift to a vehicle, we simply denote by k the shift performed by vehicle $u_k \in U$ of capacity C^k. In addition, each route starts from a depot, follows the assigned path, and, when the vehicle attains capacity or time limits, it goes to an intermediate or final destination that can coincide with the depot. In this site, the workers can perform an operation and/or make a stop before starting a new route. Each route time has to be less than or equal to T_r time units, and the shift, composed of a set of routes performed by the same vehicle, is limited by the working time of T_w time units.

Let $G = (V, A)$ be a directed fully connected graph, where $V = \{v_0, v_1, \ldots v_N\}$ is the set of nodes and A is the set of arcs. In particular, node v_0 is the depot and node v_N represents a facility that can be a node with specific role, such as a landfill in the WC system or the depot itself in typical VRPs. Moreover, each node v_i with $1 \le i \le N-1$ is a customer characterized by a demand $q_i > 0$ and a service time $p_i > 0$. Then, $\overline{V} = \{v_1, v_2, \dots v_{N-1}\}$ denotes the customer set. In addition, each arc $a_{ij} \in A$ corresponds to the shortest path from v_i to v_j of length c_{ij}, and t_{ij}^k is the time spent by vehicle u_k to perform the path of length c_{ij}.

The mth route of the kth shift is described by the following elements.

1) Set $V_m^k \subseteq \overline{V}$ contains the customers visited in the mth route of the kth shift.
2) Function $\vec{V}_m^k$: $\{1, 2, \dots, |V_m^k| + 2\} \rightarrow V_m^k \bigcup \{v_0, v_N\}$, where $\vec{V}_m^k(j) = v_i$ means that the jth node visited during the kth shift in route m is $v_i \in V_m^k \bigcup \{v_0, v_N\}$, for $j = 1, \dots, |V_m^k| + 2$. Note that $\vec{V}_m^k(2) \in \overline{V}$ denotes the first node of route m visited after v_0 or v_N, and $\vec{V}_m^k(1) \in \{v_0, v_N\}$ and $\vec{V}_m^k(|V_m^k| + 2) \in \{v_0, v_N\}$ are the first and the last nodes of the route, respectively.

Symbol $|(.)|$ denotes the cardinality of set (.).

Moreover, we denote $\bar{m} \ge 2$ the last route in the shift and $\bar{j}$ the total number of nodes visited in the mth route (i.e., $\bar{j} = |V_m^k| + 2$).

Now, the overall duration of the route described by $\vec{V}_m^k$ is computed as follows:

$$B_m^k = \sum_{j=1}^{\bar{j}-1} \left(p_{\vec{V}_m^k(j)} + t^k_{\vec{V}_m^k(j), \vec{V}_m^k(j+1)} \right). \tag{1}$$

We remark that B_m^k includes the overall traveling times and the service times of the mth route in the kth shift.

In addition, we define Q_m^k the overall load of vehicle u_k at the end of the mth route

$$Q_m^k = \sum_{j=1}^{\bar{j}} q_{\vec{V}_m^k(j)}. \tag{2}$$

At this point, it is possible to define the feasible routes as follows.

Definition 1: We say that route m performed by vehicle u_k is feasible if $Q_m^k \le C^k$ and $B_m^k \le T_r$.

Moreover, the total duration of the routes in a shift must be less than or equal to T_w, that is

$$\hat{B}^k = \sum_{i=1}^{\bar{m}} B_i^k + t_{N0}^k \le T_w. \tag{3}$$

In other words, the duration of the shift considers the durations of the shift routes B_i^k and of the final return to the depot t_{N0}^k.

The aim of the VAR problem consists of designing a set of feasible routes with minimum total length, such that the following holds.

1) $\vec{V}_m^k(j)$

$$= \begin{cases} v_0, & \text{if } m = 1 \text{ and } j = 1 \quad (4.1) \\ v_0, & \text{if } m = \bar{m} \text{ and } j = \bar{j} \quad (4.2) \\ v_N, & \text{if } 1 \le m < \bar{m} \text{ and } j = \bar{j} \quad (4.3) \\ v_N, & \text{if } 1 < m \le \overline{m} \text{ and } j = 1 \quad (4.4) \\ v_i \in \overline{V}, & \text{otherwise} \quad (4.5) \end{cases} \tag{4}$$

for each k.

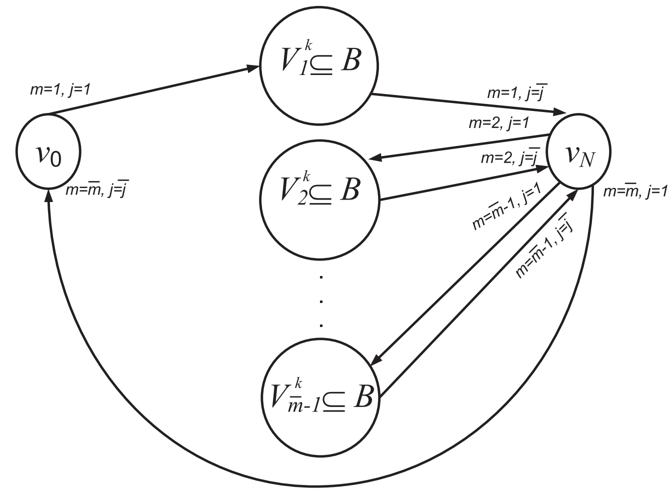


Fig. 4. Routes structure.

2) $\forall v_i \in \overline{V}$ $\exists$ only one triple of values (m, k, and j), such that $\vec{V}_m^k(j) = v_i$, i.e., each node is visited exactly once.
3) $\overline{m} \le M$, i.e., the number of routes performed by each vehicle is less than or equal to M.

More clearly, two conditions (4.1) and (4.2) impose that the first route of the each shift departs from the depot and the last route of each shift returns to the depot, respectively. On the other hand, two conditions (4.3) and (4.4) constraint each route, but the last one, to end to v_N and each route, but the first one, to start from v_N. In all the other cases, the jth visited node is a customer.

Fig. 4 shows the described conditions about the routes structure and the necessary notations are summarized as follows.

U	Set of vehicles.
$\overline{V}$	Set of customers.
$N-1$	Number of customers.
K	Maximum number of shift.
q_i	Demand of node $v_i \in V$.
p_i	Service duration at node $v_i \in V$.
C^k	Capacity of vehicle $u_k \in U$ performing the kth shift.
m	The route index in the shift.
c_{ij}	Length of the shortest path from v_i to v_j.
t_{ij}^k	Time to go from v_i to v_j by vehicle u_k.
T_r	Maximum route time of each vehicle.
T_w	Maximum working time of each vehicle.
V_m^k	Set of nodes visited in route m of the kth shift.
$\vec{V}_m^k$	Function describing the mth route of the kth shift: $\vec{V}_m^k(1), \dots, \vec{V}_m^k(\bar{j})$ gives the sequence. in which the nodes of V_m^k are visited.
Q_m^k	Total load of cluster V_m^k defined by (2).
B_m^k	Duration of route $\vec{V}_m^k$ defined by (1).
$\hat{B}^k$	Duration of the kth shift defined by (3).

B. Vehicle Assignment and Routing Algorithm

The VAR module of the U-DSS is based on a heuristic algorithm able to solve the problems

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

characterized by the assumptions and conditions described in Section III-A.

In particular, a two-phase heuristic algorithm is proposed for each shift: the first phase is based on a clustering strategy and the second phase is based on a farthest insertion heuristic for the solution of the TSP. More precisely, farthest insertion heuristic searches for the not yet inserted node whose minimum distance to all the inserted nodes is minimum. Then, the selected node is placed into the tour at the point causing the shortest increase in the total length [25]. The presented strategy, known in the related literature as cluster first, route second, is successfully applied in the considered cases where the capacity constraints are more stringent than the time constraints.

Let us consider the mth route of the kth shift. Before the application of the algorithm, the vehicles $u_k \in U$ are sorted in the descending order of their capacity C^k.

Phase 1 (Clustering Strategy): This phase is devoted to build the node set V_m^k that collects customers positions to be visited in the kth shift in its mth route. In this phase, the cluster corresponding to V_m^k is determined by considering only the vehicle capacity constraints: the cluster is built starting from the farthest node from the garage or the fulfill not served in the kth shift (the seed node). Successively, the nearest not served node to the seed is added to the cluster and so on. Hence, the cluster is completed when no other node can be added because of the capacity limit of the vehicle u_k.

The basic idea behind the clustering strategy is to first consider the farthest nodes, which require a long time to be reached. Indeed, starting from the farthest node in the clusters, construction gives a solution in which residual nodes (to be covered by the last clusters) are close among themselves and close to garage, so they can be served with a short route. This intuition is supported by the literature (see, for instance, [25]).

Phase 2 (Routing Strategy): Considering the built cluster V_m^k, the routing is determined by the farthest insertion heuristic [25]. If the constraints about the times T_r and T_w are satisfied, then the mth route of the shift k is completed. Otherwise, the last customer position inserted in the cluster is dropped and the TSP is solved again. Such a procedure is iteratively executed until the constraints on T_r and T_w are satisfied. Then, at this point, the cluster is closed. If the constraints on T_w or T_r do not permit to create a new cluster (and the related route), then we close the kth shift.

Such a procedure is iterated for each shift until all the nodes of $\overline{V}$ are included in one and only one route.

Now, the heuristic algorithm is presented by specifying the steps of the two phases. The following additional sets are introduced.

$U^U \subseteq U$	Set of used vehicles.
$U^{NU} \subseteq U$	Set of not used vehicles.
$V^S \subseteq \overline{V}$	Set of served nodes.
$V^{NS} \subseteq \overline{V}$	Set of not served nodes.

Phase 1 (Clustering Strategy):

Step 0. *Initialization.*
  Set $U^{NU} = U$, $U^U = \emptyset$, $V^S = \emptyset$, $V^{NS} = \overline{V}$, $k = 0$.

Step 1. *New shift*
  Set $k = k + 1$
  If $k \le K$ **then set** $m = 1$.
  else go to Step 9
  End

Step 2. *Choosing vehicle*
  Select $u_k \in U^{NU}$ s.t. $C^k = \max_{u_l \in U^{NU}} C^l$.

Step 3. *Determinig the seed of the mth cluster*
 Let $v_s \in V^{NS}$ be the farthest node from the depot node v_0 not yet included in any cluster (not yet served).
  Select $v_s \in V^{NS}$ $s.t.$ $c_{0s}^k = \max_{j:(v_j \in V^{NS})}\{c_{0j}^k\}$
  set $V_m^k = \{v_s\}$

Step 4. *Populating the cluster with the nearest node*
 Select the nearest not served node $v_n \in V^{NS}$ to the seed v_s:
  Select $v_n \in V^{NS} \setminus V_m^k$ $s.t.$ $c_{sn}^k = \min_{j:(v_j \in V^{NS})}\{c_{sj}^k\}$

Step 5. *Checking capacity constraints*
  If $q_n + Q_m^k \le C^k$ **then**
    set $V_m^k = V_m^k \cup \{v_n\}$ and **go to Step 4**.
  End if

Phase 2 (Solve TSP Considering the Nodes of the Cluster and Checking Time Constraints):

Step 6. *Solving TSP considering the nodes of the cluster*
  Set $\bar{j} = |V_m^k| + 2$,
  Select $\vec{V}_m^k(1)$ and $\vec{V}_m^k(\bar{j})$ according to equations (4.1) and (4.2).
  Determine $\vec{V}_m^k(2), \cdots, \vec{V}_m^k(\bar{j}-1)$ and B_m^k as solution of the *farthest insertion heuristic* described in [25] and applied to the set V_m^k.

Step 7. *Checking time constraints.*
  Determine $\hat{B}^k$
  If $\{B_m^k > T_r \vee \hat{B}^k > T_w\}$ **then**
    select $v_z \in V_m^k$ $s.t.$ $v_z = \max_{j:(v_j \in V_m^k)} c_{sj}^k$
    set $V_m^k = V_m^k - \{v_z\}$
    go to Step 6.
  End if

Step 8. *Checking shift closure conditions*
  Set $V^S = V^S \cup V_m^k$
  set $V^{NS} = V^{NS} \setminus V_m^k$
  If $V^{NS} = \emptyset$ **then go to Step 9**.
  End If
  If $\{V_m^k = \emptyset \vee m \ge M - 1\}$ **then**
    If $U^{NU} \ne \emptyset$ **then**
      call POST_FIT_PROCEDURE
      set $U^{NU} = U^{NU} \setminus \{u_k\}$
      set $U^U = U^U \cup \{u_k\}$
      and **go to Step 1**
    else go to Step 9
    End if
  else set $m = m + 1$ and **go to Step 3**.
  End if

Step 9. END

The POST_FIT_PROCEDURE looks for the smallest vehicle for the in-closure shift. Note that the procedure can be disabled whether the fleets are homogeneous.

POST_FIT_PROCEDURE:
select $u_h \in U^{NU}$ $s.t.$ $C^h = \min\{C^l | \ C^l \geqslant \ Q_i^k)$ $\forall i = 1, \ldots, m\ \}$
If $\{\exists u_h \ \wedge C^h < C^k\}$ **then**
switch u_k with u_h
End if

At Step 0, the sets U^{NU}, U^U, V^S, V^{NS}, and k are initialized. If $k \leq K$, then at Step 1, the algorithm considers the first route $m = 1$ of the new kth shift; otherwise, the algorithm stops.

At Step 2, the vehicle u_k with the biggest capacity is assigned to the shift.

Step 3 determines the seed of the cluster, i.e., the customer from which the composition of the cluster starts. The algorithm chooses the farthest customer position from the depot.

At Steps 4 and 5, the cluster is populated with the customer positions nearest to the seed. More precisely, Step 4 chooses the next node v_n and Step 5 checks the vehicle capacity constraint for the considered mth route. If the capacity constraint is satisfied, then the new customer position v_n is included in the cluster and the procedure goes to Step 4 in order to determine a new position to be included; otherwise, the cluster is completed.

When the cluster is formed, the procedure solves the TSP by considering only the cluster nodes and starting and ending according to (4.1) and (4.2). Indeed, Step 6 determines the optimal routing $\vec{V}_m^k$ as a solution of the farthest insertion heuristic.

Step 7 checks both the T_r and T_w constraints. If one of the two constraints is not satisfied, then the algorithm removes from the cluster the farthest node from the seed and goes to Step 6 in order to resolve again the TSP.

Step 8 updates the sets V^S and V^{NS} and decides if there are the conditions to complete the mth route or the shift: if it is not possible to add nodes to the mth route for the time constraints, then the shift is completed.

After the shift construction, Step 8 checks if the vehicle u_k is the best choice for the shift. Indeed, if there is a different vehicle u_h available that can perform the shift with a lower capacity than u_k, then the algorithm associates u_h to the shift instead of u_k. We remark that it is not necessary to give the limit M: if M is very high, then the procedure returns as an output the number $\bar{m}$ of the routes performed by each vehicle.

For the sake of clarity, Fig. 5 shows the UML activity diagram [26] enlightening the steps of the proposed heuristic algorithm.

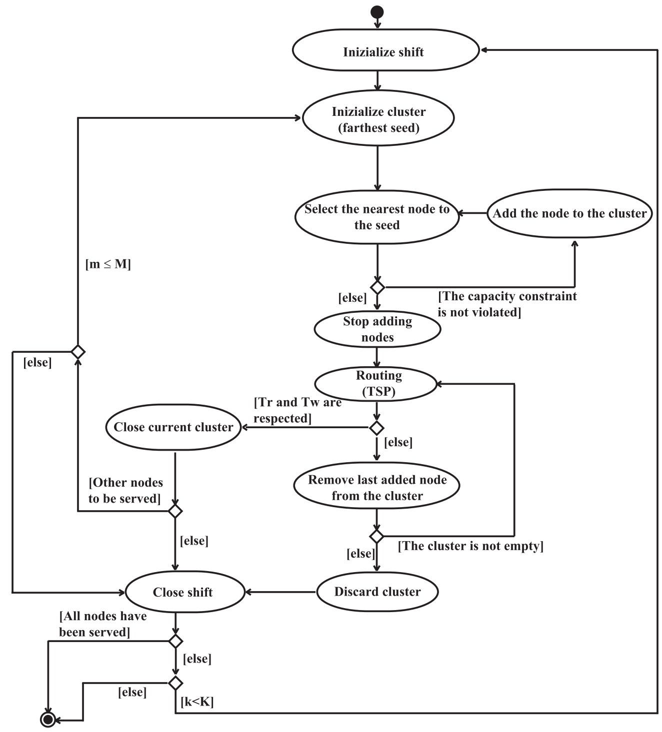


Fig. 5. UML activity diagram of the assignment and routing algorithms.

IV. Postal Delivery Problem

A. Mathematical Formulation

This section describes the configured VAR module of the U-DSS devoted to the PD for the city of Bari (Italy). The main activity of the PD system is the distribution of the postal products from the regional post automation center (PAC) to the borough PD centers (PDCs).

The postal network is modeled by a fully connected directed graph G where the nodes are the PDCs and the depot is the PAC. The delivery service is performed by a single route for each shift ($M = 1$) in which every vehicle starts from and ends to the PAC (i.e., $v_N = v_0$).

Now, the mathematical formulation to solve the VAR is presented in order to assess the proposed heuristic algorithm. The choice of the variables and the constraint structure is inspired by [27].

Decision Variables:

$$x_{ij}^k = \begin{cases} 1, & \text{if vehicle } u_k \text{ visits in sequence nodes } v_i \text{ and } v_j \\ 0, & \text{otherwise.} \end{cases}$$

Dependent Variables: Q_i^k is the total load of vehicle u_k when it leaves node $v_i \in V$. S_i^k is the starting time of the mail delivery of vehicle u_k at node $v_i \in \overline{V}$. S_N^k is the working time of vehicle u_k.

The PD problem can be formulated as the following MILP problem:

$$\min \sum_{k=1}^{K} \sum_{i,j:a_{ij}\in A} c_{ij} x_{ij}^k \tag{5}$$

subject to:
$$\sum_{k=1}^{K} \sum_{j:a_{ij}\in A} x_{ij}^k = 1 \quad \forall v_i \in \overline{V} \tag{6}$$

$$\sum_{j:a_{0j}\in} x_{0j}^k = 1 \quad \forall u_k \in U \tag{7}$$

TABLE I
CHARACTERISTICS OF DATA SETS FOR MAIL SERVICE

Sets	N [positions]	K [shifts]	20 [HU]	15 [HU]	10 [HU]	q_i [bins]	c_{ij} [km]
1	7	4	u_1	u_2,u_3	u_4	1-5	60-70
2	7	4	u_1	u_2,u_3	u_4	2-7	65-80
3	7	4	u_1	u_2,u_3	u_4	3-9	70-100
4	8	4	u_1	u_2,u_3	u_4	1-5	60-70
5	8	4	u_1	u_2,u_3	u_4	2-7	65-80
6	8	4	u_1	u_2,u_3	u_4	3-9	70-100

$$\sum_{j:a_{iN}\in A} x_{iN}^k = 1 \quad \forall u_k \in U \tag{8}$$

$$\sum_{i:a_{i0}\in A} x_{i0}^k = 0 \quad \forall u_k \in U \tag{9}$$

$$\sum_{i:a_{ij}\in A} x_{ij}^k - \sum_{j:a_{ji}\in A} x_{ji}^k = 0 \qquad \forall u_k \in U \quad \forall v_j \in \overline{V} \tag{10}$$

$$x_{ij}^k = 1 \Rightarrow Q_j^k = Q_i^k + q_j \qquad \forall u_k \in U \quad \forall a_{i,j} \in A \tag{11}$$

$$x_{ij}^k = 1 \Rightarrow S_j^k \geq S_i^k + p_i + t_{ij}^k \qquad \forall u_k \in U \quad \forall a_{i,j} \in A \tag{12}$$

$$0 \leq S_N^k \leq T_w \quad \forall u_k \in U \tag{13}$$

$$0 \leq Q_i^k \leq C^k \quad \forall u_k \in U \ \forall v_i \in V \tag{14}$$

$$x_{ij}^k \in \{0, 1\}. \tag{15}$$

The objective function (5) minimizes the length of the routes performed by all the vehicles in each work shift. Constraints (6) impose that each $v_i \in \overline{V}$ is served exactly once. Constraints (7) and (8) guarantee that every vehicle starts the route from the depot and ends it at the node $v_N = v_0$. Constraint (9) avoids that vehicles come back to v_0 after visiting $v_i \in \overline{V}$, and (10) ensures the flow conservation. Constraints (11) and (12) update the vehicle capacity and determine the arrival time of vehicle at each $v_i \in V$, respectively. Finally, (13) and (14) impose time and capacity constraints, respectively.

The outputs of the formulation are the routes assigned to the vehicles and the number of the vehicles necessary to perform the PD.

B. Validation and Results

The performances of the MILP formulation and the heuristic algorithm are compared and tested by large sets of randomly generated instances. In this section, we discuss six sets of them with five instances for each set. The characteristics of the data set are described in Table I, where the values of N and K are chosen so that the MILP formulations can be solved in reasonable time.

The instances are generated by determining at random the values of q_i and c_{ij} according to a uniform distribution into the intervals given in Table I. In addition, we consider a fleet composed by vehicles, respectively, of capacities 20, 15, and 10 handling units (HUs), which is a capacity unit represented by the yellow box normally used for the mail delivery in Italy. In particular, the fourth, fifth, and sixth column of Table I indicate the name of the vehicles associated with the corresponding capacities.

TABLE II
COMPARISON BETWEEN MILP PROBLEM AND HEURISTIC ALGORITHM

Inst.	OPT	UB	CPU-OPT	CPU-UB	*Gap*
1.1	620.44 km	637.50 km	561 s	13 ms	2.75 %
1.2	621.61 km	636.01 km	332 s	11 ms	2.32 %
1.3	636.54 km	641.10 km	464 s	14 ms	0.72 %
1.4	632.14 km	632.14 km	527 s	12 ms	0.00 %
1.5	637.89 km	643.76 km	180 s	12 ms	0.92 %
2.1	775.75 km	778.48 km	340 s	14 ms	0.35 %
2.2	778.10 km	790.50 km	75 s	14 ms	1.59 %
2.3	762.09 km	772.69 km	480 s	15 ms	1.39 %
2.4	779.70 km	779.70 km	110 s	13 ms	0.00 %
2.5	775.59 km	776.94 km	68 s	14 ms	0.17 %
3.1	888.86 km	931.50 km	38 s	13 ms	4.80 %
3.2	861.79 km	873.17 km	23 s	11 ms	1.32 %
3.3	891.83 km	932.36 km	38 s	14 ms	4.54 %
3.4	854.98 km	862.17 km	62 s	13 ms	0.84%
3.5	853.14 km	869.85 km	66 s	14 ms	1.96%
4.1	695.26 km	708.22 km	1000 s	16 ms	1.86 %
4.2	696.62 km	699.42 km	1000 s	15 ms	0.40 %
4.3	686.71 km	693.95 km	1000 s	14 ms	1.05 %
4.4	687.75 km	707.89 km	796 s	13 ms	2.93 %
4.5	688.46 km	698.96 km	1000 s	14 ms	1.53 %
5.1	837.12 km	848.61 km	1000 s	15 ms	1.37 %
5.2	846.69 km	856.20 km	1000 s	16 ms	1.12 %
5.3	834.72 km	861.66 km	895 s	16 ms	3.23 %
5.4	828.48 km	828.48 km	923 s	13 ms	0.00 %
5.5	853.73 km	838.62 km	1000 s	12 ms	0.35 %
6.1	967.71 km	975.00 km	1000 s	13 ms	0.75 %
6.2	939.35 km	989.21 km	763 s	13 ms	5.31 %
6.3	956.05 km	981.45 km	1000 s	14 ms	2.66%
6.4	933.38 km	971.98 km	273 s	12 ms	4.14%
6.5	943.87 km	974.96 km	553 s	11 ms	3.29%

In order to determine the travel times t_{ij}^k, we assume an average speed of 50 km/h. Moreover, the service time p_i associated with $v_i \in B$ is 15 min and the vehicle working time is $T_w = 360$ min.

The MILP problem and the heuristic algorithm are implemented in the MATLAB environment on a PC equipped by a 3.40-GHz Intel i7-3770 with 8 GB of memory, in the single-thread mode. In particular, the MILP problem is solved by using the Gnu linear programming kit [28].

Table II summarizes the results that are obtained by allowing a maximum of 1000 s of CPU time for the solution of each instance: the objective function values of the MILP formulation (OPT), the objective function values of the heuristic algorithm (UB), and the CPU times to obtain them (CPU-OPT and CPU-UB).

Moreover, the last column of Table II reports the optimality gap (named Gap) expressed as follows:

$$\text{Gap} = (\text{UB} - \text{OPT})/\text{OPT} * 100. \tag{16}$$

In particular, for the instances where the CPU time is equal to 1000 s, the optimal solution is not reached and the value of the best feasible obtained solution is reported. The results

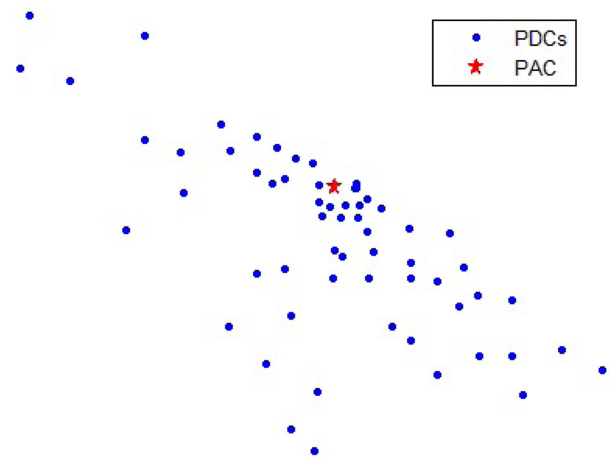


Fig. 6. Node locations of the PD system of the case study.

TABLE III
VEHICLE FLEET

u_k	C^k (HU)
$u_1 - u_6$	20
$u_7 - u_{12}$	15
$u_{13} - u_{15}$	10

of Table II show that the value of the optimality gap averaged on the 30 instances is equal to 1.79%, with a maximum value of 5.31% and a variance of 2.27%. Moreover, for three instances, the two solutions coincide. In conclusion, the proposed heuristic algorithm obtains very good solutions in all considered instances and runs in <17 ms. Hence, the presented algorithm can be used to solve large and complex problems in a short time.

C. Case Study of a Postal Delivery System

In this section, the U-DSS PD module is assessed by considering the case study of Bari. The network is modeled by the graph shown in Fig. 6 that shows the locations of the nodes without showing the arcs for the sake of clarity. Hence, the network is composed by $N = 63$ nodes, where 61 nodes represent the PDCs and the node $v_0 = v_N$ is the PAC that is symbolized by a red triangle in Fig. 6. So, graph G is composed by $(N+1)^2 - (2N) = 3843$ directed arcs.

The values of the elements c_{ij} and t_{ij}^k are determined on the basis of the results obtained by Google Maps Application Programming Interface. The evaluation of these parameters is crucial for the routes optimization: indeed, they can be affected by the traffic congestion, the weather forecast, the number of street lanes, the slope of the streets, and the considered period of the year and of the day.

Moreover, the capacity constraints depend on the vehicle u_k used in the kth shift. In particular, the considered system has $K = 15$ vehicles with the capacities reported in Table III: vehicles $u_1, \ldots, u_6$ have capacity equal to 20 HUs, $u_7, \ldots, u_{12}$ have capacity equal to 15 HUs, and u_{13}, u_{14}, u_{15} have capacity equal to 10 HUs. The service duration in each node $v_i \in \overline{V}$ is

TABLE IV
PDC DEMAND OF THE CASE STUDY (IN HU)

v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	v_{11}
5	3	3	3	2	2	3	2	3	2	2
v_{12}	v_{13}	v_{14}	v_{15}	v_{16}	v_{17}	v_{18}	v_{19}	v_{20}	v_{21}	v_{22}
3	3	6	3	3	5	2	3	7	1	5
v_{23}	v_{24}	v_{25}	v_{26}	v_{27}	v_{28}	v_{29}	v_{30}	v_{31}	v_{32}	v_{33}
6	5	1	6	1	3	1	6	1	2	3
v_{34}	v_{35}	v_{36}	v_{37}	v_{38}	v_{39}	v_{40}	v_{41}	v_{42}	v_{43}	v_{44}
1	2	1	2	3	3	1	1	1	4	1
v_{45}	v_{46}	v_{47}	v_{48}	v_{49}	v_{50}	v_{51}	v_{52}	v_{53}	v_{54}	v_{55}
1	3	3	1	1	1	1	1	1	1	1
v_{56}	v_{57}	v_{58}	v_{59}	v_{60}	v_{61}					
1	5	1	1	2	1					

TABLE V
ROUTE CHARACTERISTICS OF THE SOLUTION

Route	**duration** (min)	**length** (km)	**load** (HU)	**vehicle/capacity**
1	349.45	364.86	14	u_7/15 HU
2	296.92	313.84	18	u_1/20 HU
3	340.28	348.58	15	u_8/15 HU
4	341.53	322.43	19	u_2/20 HU
5	345.57	268.75	13	u_9/15 HU
6	335.52	234.89	18	u_3/20 HU
7	359.37	222.49	15	u_{10}/15 HU
8	317.72	154.65	18	u_4/20 HU
9	291.63	93.34	20	u_5/20 HU
10	66.15	28.10	2	u_{13}/10 HU
Sums	**3044.13**	**2351.92**	**152**	

equal to $p_i = 15$ min, while the service duration in the PAC is $p_0 = p_N = 0$. The demand of each PDC daily changes: the values of the demand of each node that are considered in the case study are reported in Table IV. Since only one route can be performed in each shift, it holds $T_r = T_w = 360$ min.

The solution of the case study is summarized in Tables V and VI. Let us remark that the first route is performed by the vehicle u_7 of capacity 15 HU because of the post fit procedure. Indeed, at first, the heuristic algorithm tries to perform the route with the biggest available vehicle u_1. However, due to time constraints, the vehicle capacity is not used completely (14 HUs of 20); hence, the algorithm switches u_1 with the vehicle u_7 of 15 HUs. In this way, a better utilization of the vehicles is obtained.

Remark that, in route 10 (see Table V), the capacity of the vehicle is underutilized: this happens because there are not vehicles with a capacity smaller than 10 HUs, and route 10 visits the last two residual nodes, near the depot. The DM could decide to assign those nodes to some of the other

TABLE VI
ROUTE NODES OF THE SOLUTION

Route	$\vec{V}_1^k$
1	$v_0\ v_{38}\ v_{60}\ v_{61}\ v_{57}\ v_{39}\ v_0$
2	$v_0 v_{14} v_{26} v_{21} v_{22}\ v_0$
3	$v_0\ v_{33}\ v_{34}\ v_{35}\ v_{16}\ v_{30}\ v_0$
4	$v_0\ v_{23}\ v_{24}\ v_{17}\ v_{58}\ v_{32}\ v_0$
5	$v_0\ v_5\ v_{13}\ v_{27}\ v_{28}\ v_{29}\ v_3\ v_0$
6	$v_0\ v_{20}\ v_{25}\ v_{45}\ v_{43} v_{12}\ v_{44}\ v_{59}\ v_0$
7	$v_0\ v_{50}\ v_{11}\ v_{48}\ v_{31}\ v_{15} v_{49}\ v_{47}\ v_{46}\ v_0$
8	$v_0\ v_6\ v_{36}\ v_{56}\ v_{55}\ v_{19}\ v_4\ v_7\ v_{37}\ v_8\ v_0$
9	$v_0\ v_{53}\ v_{51}\ v_{42}\ v_{40}\ v_{41} v_{10}\ v_{18}\ v_2\ v_9\ v_1\ v_0$
10	$v_0\ v_{52}\ v_{54}\ v_0$

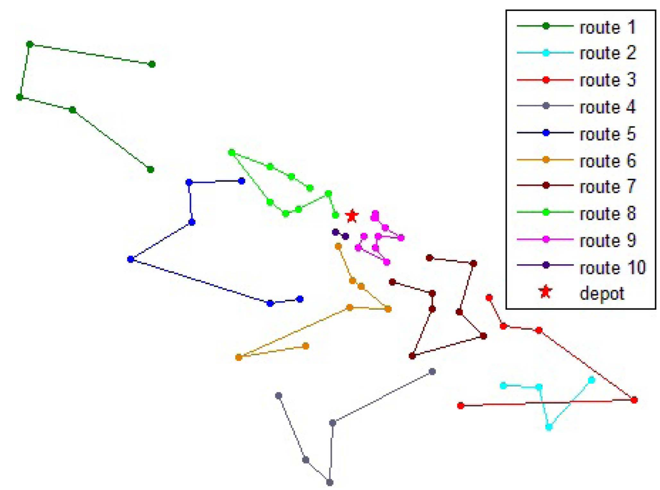


Fig. 7. Clusters obtained by the heuristic algorithm.

routes, by applying them an extra time. The clusters are graphically represented in Fig. 7 with different colors: the red star represents the depot. For the sake of clarity, in Fig. 7, the connections with the depot are not depicted.

V. WASTE COLLECTION PROBLEM

A. Waste Collection System Description and Mathematical Formulation

In this section, the configured VAR module of the U-DSS devoted to a WC system is described. We consider a maximum number K of shifts and a set of K identical vehicles $U = \{u_1, u_2, \ldots, u_K\}$: each vehicle $u_k \in U$ can collect C^k bins. Every day, a team of workers drives the vehicles to perform the WC subject to time and capacity constraints. During a work shift, each driver of the team can perform several routes for a maximum of M.

The first travel starts from the depot, follows the assigned route by collecting the bin waste, and, when the truck is full or the time limit is attained, it has to stop at the landfill. In this site, the workers unload the waste, make a stop, and then start a new route from the landfill. On the basis of the company specifications, the work shift goes to an end by an empty travel from the landfill to the garage. Each route time has to be less than or equal to T_r min and the shift, composed of a set of routes, is limited by the working time of T_w min.

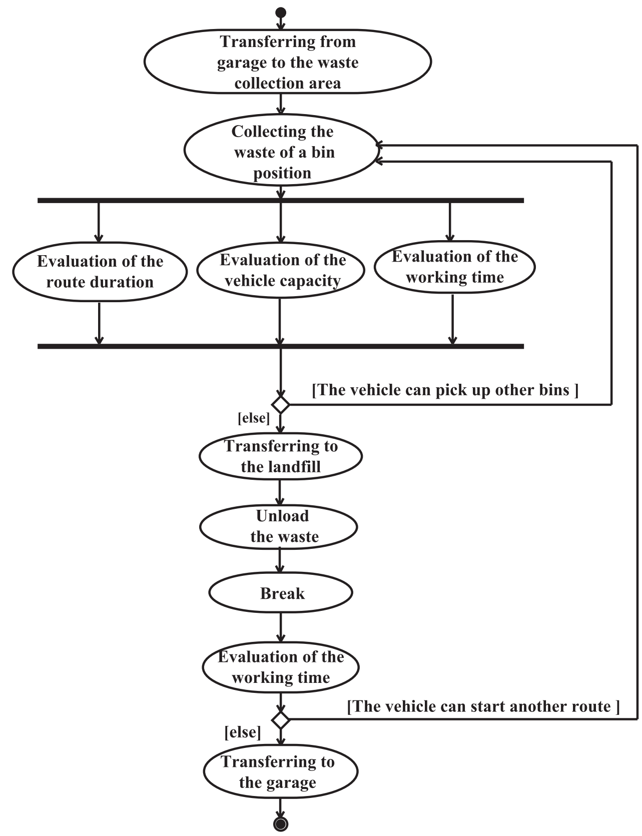


Fig. 8. Activity diagram of the vehicle during a shift.

Fig. 8 shows the UML activity diagram of the vehicle shift highlighting the actions of drivers and vehicles.

The problem is minimizing the length of all the routes performed by the vehicles in each work shift. Each route is given as a sequence of visited nodes with the indication of the arrival times.

Now, a mathematical formulation of the considered VAR is presented on the basis of the model in [27], but also providing several routes in each shift. Hence, the decision and dependent variables are slightly modified with respect to the MILP (5)–(15).

Decision Variables:

$$x_{ij,m}^k = \begin{cases} 1, & \text{if vehicle } u_k \text{ visits in sequence nodes } v_i \text{ and } v_j \\ & \text{during route } m \text{ of shift } k \\ 0, & \text{otherwise.} \end{cases}$$

Dependent Variables:

1) $Q_{i,m}^k$ load collected by vehicle u_k when it leaves node $v_i \in V$ during route m;
2) $S_{i,m}^k$ starting time of the pickup operation of vehicle u_k at node $v_i \in \overline{V}$ during route m;
3) $S_{N,m}^k$ arrival time to node v_N of route m performed by u_k;
4) $S_{0,m}^k$ working time of vehicle u_k during route m.

The considered VAR consists of designing a set of feasible routes with minimum total length, such that the following holds.

1) Each node is visited by exactly one vehicle.
2) The first route starts from v_0 and ends to v_N.
3) The last (empty) route starts from v_N and ends to v_0.

4) The remaining routes start from and end to v_N.
5) The number of routes performed by each vehicles is less than or equal to M.

Now, the considered VAR can be formulated by the following MILP problem:

$$\min \sum_{k=1}^{K} \sum_{m=1}^{M} \sum_{i,j:a_{ij}\in A} c_{ij} x_{ij,m}^{k} \tag{17}$$

$$\text{subject to: } \sum_{m=1}^{M} \sum_{k=1}^{K} \sum_{j:a_{ij}\in A} x_{ij,m}^{k} = 1 \quad \forall v_i \in \overline{V} \tag{18}$$

$$\sum_{j:a_{0j}\in A} x_{0j,1}^{k} = 1 \quad \forall u_k \in U \tag{19}$$

$$\sum_{m=2}^{M} \sum_{j:a_{0j}\in A} x_{0j,m}^{k} = 0 \quad \forall u_k \in U \tag{20}$$

$$\sum_{j:a_{Nj}\in A} x_{Nj,m}^{k} \le 1 \quad \forall u_k \in U,\ m = 2,\ldots,M \tag{21}$$

$$\sum_{i:a_{iN}\in A} x_{iN,m}^{k} \le 1 \quad \forall u_k \in U,\ m = 1,\ldots,M \tag{22}$$

$$\sum_{i:a_{ij}\in A} x_{ij,m}^{k} - \sum_{i:a_{ij}\in A-\{a_{j0}\}} x_{ji,m}^{k} = 0 \quad \forall v_j \in \overline{V} \quad \forall u_k \in U\ \forall m \tag{23}$$

$$\sum_{i:a_{iN}\in A} x_{iN,m}^{k} - \sum_{j:a_{Nj}\in A} x_{Nj,m+1}^{k} = 0 \quad \forall u_k \in U, m = 1,2,\ldots,M-1 \tag{24}$$

$$\sum_{m=1}^{M} x_{N0,m}^{k} = 1 \quad \forall u_k \in U \tag{25}$$

$$x_{ij,m}^{k} = 1 \Rightarrow S_{j,m}^{k} \ge S_{i,m}^{k} + p_i + t_{ij}^{k} \quad \forall u_k \in U \quad \forall a_{i,j} \in A \text{ s.t. } i \ne N\ \forall m \tag{26}$$

$$x_{Nj,m}^{k} = 1 \Rightarrow S_{j,m}^{k} \ge S_{N,m-1}^{k} + p_N + t_{Nj}^{k} \quad \forall u_k \in U, \quad \forall j \text{ s.t. } a_{N,j} \in A\ \forall m > 1 \tag{27}$$

$$x_{ij,m}^{k} = 1 \Rightarrow Q_{j,m}^{k} = Q_{i,m}^{k} + q_j \quad \forall u_k \in U \quad \forall a_{i,j} \in A \text{ s.t. } i \ne N\ \forall m \tag{28}$$

$$x_{Nj,m}^{k} = 1 \Rightarrow Q_{j,m}^{k} = q_j \quad \forall u_k \in U \quad \forall j \text{ s.t. } a_{N,j} \in A\ \forall m > 1 \tag{29}$$

$$0 \le S_{N,1}^{k} \le T_r \quad \forall u_k \in U \tag{30}$$

$$0 \le S_{N,m}^{k} - S_{N,m-1}^{k} - p_N \le T_r \quad \forall u_k \in U\ \forall m > 1 \tag{31}$$

$$0 \le S_{0,m}^{k} \le T_w \quad \forall u_k \in U\ \forall m \tag{32}$$

$$0 \le Q_{i,m}^{k} \le C^{k} \quad \forall u_k \in U \quad \forall v_i \in V\ \forall m \tag{33}$$

$$x_{ij,m}^{k} \in \{0,1\}. \tag{34}$$

The objective function (17) minimizes the length of the routes performed by all the vehicles in each work shift. Constraint (18) imposes that each node $v_i \in \overline{V}$ has to be served exactly once. Constraints (19)–(21) guarantee that every vehicle starts the first route at the garage, while the other routes begin from the disposal site. Moreover, constraint (22) states that each vehicle ends its route at the landfill site, and constraint (23) ensures the flow conservation avoiding that the vehicles come back to the garage after having served a node $v_i \in B$. Constraint (24) connects the end of the route m to the start of the route $m+1$. Constraint (25) imposes that the last route of the shift has to be performed from the disposal site to the garage.

Constraints (26) and (27) determine the arrival time at node $v_i \in V$, and constraints (28) and (29) update the loads $Q_{i,m}^{k}$. On the other hand, (30)–(33) impose the time and capacity constraints.

Note that the nonlinear constraints (26)–(29) can be linearized by a big number formulation [29].

TABLE VII
CHARACTERISTICS OF DATA SETS

Sets	N [positions]	K [vehicles]	q_i [bins]	c_{ij} [kilometers]
1	6	3	30-40	20-30
2	6	3	35-45	15-25
3	6	3	30-50	10-40
4	7	3	30-40	20-30
5	7	3	35-45	15-25
6	7	3	30-50	10-40
7	8	4	30-40	20-30
8	8	4	35-45	15-25
9	8	4	30-50	10-40

B. Validation and Results

The performances of the MILP formulation and the heuristic algorithm are compared and tested by nine sets of randomly generated instances. Table VII reports the data sets chosen to compare the two approaches: the values of N and K are selected, so that the MILP formulations can be solved in reasonable time. Note that the instances of the sets 1–3, 4–6, and 7–9 generate 720, 891, and 1440 variables, respectively. Moreover, the sets 1–3 as well as the sets 4–6 and 7–9 differ for the demand in each pickup position and for the distance between the nodes: these parameters are generated with uniform probabilities between the values shown, respectively, in the fourth and fifth columns of Table VII.

In order to determine the route times t_{ij}^{k} between two consecutive pickup sites, we assume a speed of 40 km/h in the following cases: 1) the vehicle goes from the depot to the first not served pickup point; 2) the vehicle goes from the landfill to the first not served pickup point; 3) the vehicle comes back to the landfill from the last served pickup point; and 4) the vehicle comes back to the garage from the landfill. In all the other cases, the average speed is assumed equal to 20 km/h.

Moreover, the service time p_i associated with $v_i \in B$ is determined on the basis of the number of bins located in the considered pickup position: the vehicle simultaneously loads two bins in 1.5 min. On the other hand, the unloading time

TABLE VIII
COMPARISON BETWEEN MILP PROBLEM AND HEURISTIC ALGORITHM

Inst.	OPT	UB	CPU-OPT	CPU-UB	*Gap*
1.1	324.67 km	337.36 km	24 s	26 ms	3.91 %
1.2	326.62 km	336.44 km	29 s	23 ms	3.01 %
1.3	327.77 km	335.43 km	25 s	28 ms	2.34 %
1.4	296.78 km	302.26 km	12 s	23 ms	1.85 %
1.5	336.92 km	336.92 km	25 s	23 ms	0.00 %
2.1	270.55 km	276.12 km	21 s	25 ms	2.06 %
2.2	274.48 km	277.08 km	18 s	25 ms	0.95 %
2.3	250.38 km	260.09 km	25 s	23 ms	3.88 %
2.4	273.56 km	291.48 km	22 s	25 ms	6.55 %
2.5	260.55 km	281.83 km	18 s	26 ms	8.16 %
3.1	262.48 km	270.42 km	6 s	23 ms	3.02 %
3.2	294.96 km	319.06 km	14 s	26 ms	8.17 %
3.3	228.17 km	228.17 km	6 s	20 ms	0.00 %
3.4	258.15 km	261.31 km	26 s	23 ms	1.25 %
3.5	267.62 km	281.27 km	13 s	23 ms	5.10 %
4.1	376.36 km	386.56 km	102 s	23 ms	2.71 %
4.2	369.89 km	387.03 km	265 s	29 ms	4.63 %
4.3	379.81 km	386.00 km	232 s	26 ms	1.89 %
4.4	364.07 km	375.68 km	84 s	24 ms	3.19 %
4.5	374.37 km	433.83 km	166 s	28 ms	15.77 %
5.1	320.66 km	345.12 km	98 s	31 ms	7.63 %
5.2	314.91 km	342.94 km	120 s	29 ms	8.91 %
5.3	309.72 km	348.84 km	29 s	26 ms	12.63 %
5.4	312.38 km	352.76 km	41 s	27 ms	12.93 %
5.5	323.08 km	363.32 km	137 s	25 ms	12.45 %
6.1	369.20 km	435.14 km	11 s	26 ms	17.86 %
6.2	353.40 km	369.41 km	30 s	26 ms	4.53 %
6.3	338.60 km	393.52 km	32 s	27 ms	16.22%
6.4	305.49 km	336.14 km	8 s	27 ms	10.03%
6.5	314.21 km	349.04 km	13 s	25 ms	11.08%
7.1	417.08 km	434.45 km	1000 s	26 ms	4.16 %
7.2	414.14 km	422.70 km	1000 s	26 ms	1.95 %
7.3	421.70 km	431.66 km	1000 s	27 ms	2.36 %
7.4	441.12 km	462.74 km	843 s	29 ms	4.90 %
7.5	442.35 km	476.02 km	748 s	31 ms	7.61 %
8.1	349.10 km	366.66 km	1000 s	28 ms	5.03 %
8.2	344.15 km	357.50 km	1000 s	29 ms	3.88 %
8.3	367.87 km	377.50 km	1000 s	28 ms	2.62 %
8.4	363.33 km	378.44 km	1000 s	26 ms	4.16 %
8.5	348.70 km	357.70 km	1000 s	28 ms	2.58 %
9.1	351.98 km	378.11 km	1000 s	28 ms	7.42 %
9.2	373.88 km	404.99 km	1000 s	27 ms	8.55 %
9.3	383.16 km	418.24 km	725 s	31 ms	9.15%
9.4	403.37 km	428.82 km	1000 s	31 ms	6.31%
9.5	413.37 km	428.56 km	1000 s	28 ms	3.67%

at the landfill is $p_N = 30$ min and includes the compulsory work break. Moreover, it holds $q_0 = q_N = 0$ and $p_0 = 0$.

Summing up, the route times t_{ij}^k for $a_{ij} \in A$ and the service times p_i for $v_i \in V$ are determined by the following relations:

$$t_{ij}^k = \begin{cases} \dfrac{c_{ij} \cdot 60}{20000} & \forall i, j \text{ with } i \neq 0, N \text{ or } j \neq 0, N \\ \dfrac{c_{ij} \cdot 60}{40000}, & \text{otherwise} \end{cases} \tag{35}$$

$$p_i = \begin{cases} 0, & \text{if } i = 0 \\ 30, & \text{if } i = N \\ 1.5 \left\lceil \dfrac{q_i}{2} \right\rceil, & \text{otherwise.} \end{cases} \tag{36}$$

Table VIII shows the results that are obtained by allowing a maximum of 1000 s of CPU time for the solution of each instance.

In particular, we remember that for the instances where the CPU time is equal to 1000 s, the optimal solution has been not reached and the value of the best feasible obtained solution is reported. The results of Table VIII show that the value of the optimality gap (16) averaged on the 45 instances is equal to 5.93%, with a maximum value of 17.86%. Moreover, it is important to note that *Gap* is <10% for 37 instances, it is <5% for 25 instances and for 2 instances that the two solutions coincide. Considering only the 33 optimal solutions, the *Gap* is <10% for 25 instances. The *Gap* of the instances 4.5 and 6.1 is justified by the fact that the MILP problem solution provides three vehicles, while the heuristic algorithm provides four vehicles: this difference determines the largest gap between the two solutions.

TABLE IX
SOLUTION OBTAINED BY THE HEURISTIC ALGORITHM

u_k	m	duration [minutes]	length [kilometers]	Loaded bins [bins]
1	1	173.57	47.85	33
	2	179.54	53.49	35
2	1	179.23	54.49	44
	2	173.21	41.81	55
3	1	174.53	53.78	49
	2	176.76	47.60	60
4	1	176.86	36.97	86
	2	173.82	44.24	59
5	1	178.45	39.35	72
	2	173.39	43.09	55
6	1	173.15	42.83	57
	2	178.04	37.10	89
7	1	144.14	26.44	89
	2	178.17	34.40	87
8	1	168.82	34.58	90
	2	179.31	37.36	90
9	1	149.13	28.30	90
	2	165.96	34.60	89
10	1	173.79	28.12	90
	2	140.77	25.15	86
11	1	156.42	24.20	90
	2	161.57	29.49	89
12	1	115.67	19.20	89
	2	132.01	24.01	90
	3	74.28	17.55	25
13	1	140.41	23.68	89
	2	118.49	20.63	89
	3	63.54	17.41	26
14	1	140.86	25.06	90
	2	175.04	25.96	84
15	1	136.94	22.34	90
	2	143.10	20.82	90
	3	41.93	12.33	19
16	1	154.19	23.75	90
	2	121.53	19.06	90
	3	45.76	13.33	18
17	1	155.61	26.08	87
	2	152.21	24.93	88
	3	14.15	7.43	4
18	1	81.19	15.70	48
Total		3986.5	923.2	2419

In conclusion, the proposed heuristic algorithm obtains good solutions in most of the considered instances and runs in less than 31 ms.

C. Case Study of a Waste Collection System

In this section, the case study of the WC problem of a neighborhood of Bari is described and solved. The maximum number of shifts is $K = 24$ that is the number of the shifts performed in the current waste management. Moreover, 24 identical vehicles form the fleet with capacity $C^k = 90$ bins for $k = 1, \ldots, 24$. During a work shift, each driver of the team can perform a maximum number of $M = 4$ routes and it holds $T_r = 180$ and $T_w = 420$ min.

The WC system of the considered neighborhood consists of 2790 bins distributed over $N - 2 = 1498$ pickup positions. The graph G is of 1500 nodes and $(N + 1)^2 - (N + 1) = 2\,248\,500$ arcs. In order to perform a realistic evaluation of the routes, each value of c_{ij} is increased of 0.15 and 0.30 km for each left-turn and U-turn, respectively. Each parameter t_{ij} is determined by the sum of the following terms: the vehicle average speed obtained by (35) and some penalties due to weather and traffic conditions, rash hours, condition, and slope of the street.

The solution of the considered WC obtained by the heuristic algorithm is shown in Table IX. It is apparent that 18 shifts and 40 routes are necessary to perform the WC. More precisely, most of the shifts are composed of two routes, not considering the fixed empty route from the landfill to the garage at the end of the shift. Although these end-shift routes are omitted in Table IX, total duration and total length consider them (each of them has a duration of 7 min for a length of 4.5 km).

In order to show the effectiveness of the proposed algorithm, the obtained solution is compared with the current waste management of the case study where 24 shifts composed of two routes (not considering the last route) are now performed. Hence, the proposed solution allows saving six shifts and eight collecting routes with significant economic and environmental benefits.

VI. Conclusion

This paper presents a decision support approach for optimizing delivery and collection services in a metropolitan area. To this aim, a U-DSS is designed to support DMs about the work shifts organization by minimizing the travel times of the vehicle routes in the shifts. Then, in order to deal with real cases, a two-phase heuristic algorithm is proposed based on a clustering strategy and a farthest insertion heuristic for the TSP.

In particular, PD and WC services are considered, each of them with their own peculiarities, but also with common aspects that allow designing a common framework.

While the strategy and the heuristic algorithm are the same for both services, in order to validate the algorithm for different applications, two MILP formulations are presented: in any case the aim is minimizing the total length of all the planned routes.

For the PD service, a heterogeneous vehicle fleet is involved: the applied heuristic algorithm allows not only minimizing the vehicles routing but also using vehicles of suitable dimensions.

About the WC service, a complex and constrained problem is addressed: 1) capacity and time constraints are considered; 2) a network with a large number of bins characterized by different demand and pickup times is considered; and 3) each vehicle can perform several routes during a daily shift.

Comparing the presented approach with the solutions proposed in the related literature, we point out the following features.

1) A unified framework to solve different VAR problems is considered.
2) The problem is general and can consider several landfills, different vehicles with different capacities, and speed.
3) The routes can form several shifts satisfying time and capacity constraints.
4) The traveling times are obtained by the routes and the distances determined by Google Map application program interface, the traffic, and weather forecasts.
5) The solutions are obtained in extremely short times, even if the dimensions of the problem are very large.

Future research will focus on extending the U-DSS by considering dynamic routing in order to modify in real time the planning of the routes.

References

[1] G. Kim, Y.-S. Ong, C. K. Heng, P. S. Tan, and N. A. Zhang, "City vehicle routing problem (City VRP): A review," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 4, pp. 1654–1666, Aug. 2015.

[2] V. Pillac, M. Gendreau, C. Guéret, and A. L. Medaglia, "A review of dynamic vehicle routing problems," *Eur. J. Oper. Res.*, vol. 225, no. 1, pp. 1–11, 2013.

[3] B. Eksioglu, A. V. Vural, and A. Reisman, "The vehicle routing problem: A taxonomic review," *Comput. Ind. Eng.*, vol. 57, no. 4, pp. 1472–1483, 2009.

[4] G. Laporte, M. Gendreau, J.-Y. Potvin, and F. Semet, "Classical and modern heuristics for the vehicle routing problem," *Int. Trans. Oper. Res.*, vol. 7, nos. 4–5, pp. 285–300, 2000.

[5] Q. Song, X. Wang, X. Li, and C. Zhang, "Optimization of postal express mail network based on swarm intelligence," in *Proc. 48th IEEE Conf. Decision Control, Held Jointly With 28th Chin. Control Conf. (CDC/CCC)*, Dec. 2009, pp. 591–596.

[6] H. Yang, Y. Zhou, Z. Cui, and M. He, "Vehicle routing problem with multi-depot and multi-task," *Adv. Inf. Sci. Service Sci.*, vol. 3, no. 6, pp. 320–327, 2011.

[7] B. L. Hollis, M. A. Forbes, and B. E. Douglas, "Vehicle routing and crew scheduling for metropolitan mail distribution at Australia Post," *Eur. J. Oper. Res.*, vol. 173, no. 1, pp. 133–150, 2006.

[8] S. Tsuruta, E. Ohsugi, Y. Toyama, and T. Onoyama, "Hybrid method for a postal delivery route generation," in *Proc. TENCON*, vol. 2, 2000, pp. 550–555.

[9] L. W. Dong and C. T. Xiang, "Ant colony optimization for VRP and mail delivery problems," in *Proc. IEEE Int. Conf. Ind. Inform.*, Aug. 2006, pp. 1143–1148.

[10] Q. Song, C. Zhang, X. Li, and F. Hao, "Genetic algorithm based modeling and optimization of the borough postal transportation network," in *Proc. 46th IEEE Conf. Decision Control*, Dec. 2007, pp. 2850–2855.

[11] X. Li and F. Peng, "The shortest path and spatial decision support system implementation in the context of parcel delivery," in *Proc. 18th Int. Conf. Geoinform.*, 2010, pp. 1–6.

[12] I. von Poser and A. R. Awad, "Optimal routing for solid waste collection in cities by using real genetic algorithm," in *Proc. 2nd Int. Conf. Inf. Commun. Technol. (ICTTA)*, vol. 1, 2006, pp. 221–226.

[13] A. P. Singh, G. Ruhe, S. A. H. Amereei, and S. Banack, "Decision support for capacitated arc routing for providing municipal waste and recycling services," in *Proc. 47th Hawaii Int. Conf. Syst. Sci. (HICSS)*, 2014, pp. 986–993.

[14] J. Liu, D. Liu, M. Liu, and Y. He, "An improved multiple ant colony system for the collection vehicle routing problems with intermediate facilities," in *Proc. 8th World Congr. Intell. Control Autom. (WCICA)*, 2010, pp. 3078–3083.

[15] J. Liu and Y. He, "A clustering-based multiple ant colony system for the waste collection vehicle routing problems," in *Proc. 5th Int. Symp. Comput. Intell. Design (ISCID)*, vol. 2, 2012, pp. 182–185.

[16] J. Bautista, E. Fernández, and J. Pereira, "Solving an urban waste collection problem using ants heuristics," *Comput. Oper. Res.*, vol. 35, no. 9, pp. 3020–3033, 2008.

[17] R. Revetria, A. Testa, and L. Cassettari, "A generalized simulation framework to manage logistics systems: A case study in waste management and environmental protection," in *Proc. Winter Simulation Conf. (WSC)*, 2011, pp. 943–952.

[18] R. Islam and M. S. Rahman, "An ant colony optimization algorithm for waste collection vehicle routing with time windows, driver rest period and multiple disposal facilities," in *Proc. Int. Conf. Inform., Electron. Vis. (ICIEV)*, 2012, pp. 774–779.

[19] B.-I. Kim, S. Kim, and S. Sahoo, "Waste collection vehicle routing problem with time windows," *Comput. Oper. Res.*, vol. 33, no. 12, pp. 3624–3642, 2006.

[20] D. V. Tung and A. Pinnoi, "Vehicle routing–scheduling for waste collection in Hanoi," *Eur. J. Oper. Res.*, vol. 125, no. 3, pp. 449–468, 2000.

[21] A. Quintero, D. Konaré, and S. Pierre, "Prototyping an intelligent decision support system for improving urban infrastructures management," *Eur. J. Oper. Res.*, vol. 162, no. 3, pp. 654–672, 2005.

[22] L. Santos, J. Coutinho-Rodrigues, and C. H. Antunes, "A Web spatial decision support system for vehicle routing using Google Maps," *Decision Support Syst.*, vol. 51, no. 1, pp. 1–9, 2011.

[23] F. Sperandio, C. Gomes, J. Borges, A. C. Brito, and B. Almada-Lobo, "An intelligent decision support system for the operating theater: A case study," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 1, pp. 265–273, Jan. 2014.

[24] V. Boschian, M. Dotoli, M. P. Fanti, G. Iacobellis, and W. Ukovich, "A metamodeling approach to the management of intermodal transportation networks," *IEEE Trans. Autom. Sci. Eng.*, vol. 8, no. 3, pp. 457–469, Jul. 2011.

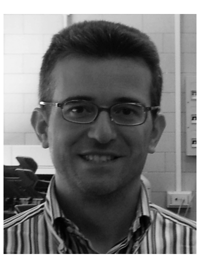
[25] M. Jünger, G. Reinelt, and G. Rinaldi, "The traveling salesman problem," in *Handbooks in Operations Research and Management Science* (Network Models), vol. 7, M. O. Ball, T. L. Magnanti, C. L. Monma, and G. L. Nemhauser, Eds. Amsterdam, The Netherlands: North-Holland, 1995.

[26] R. Miles and K. Hamilton, *Learning UML 2.0*. Sebastopol, CA, USA: O'Reilly Media, 2006.

[27] S. N. Parragh, K. F. Doerner, and R. F. Hartl, "A survey on pickup and delivery problems. Part II: Transportation between pickup and delivery locations," *J. für Betriebswirtschaft*, vol. 58, no. 2, pp. 81–117, 2008.

[28] *GLPKMEX—A MATLAB MEX Interface for GLPK Library*, accessed Sep. 1, 2015. [Online]. Available: http://glpkmex.sourceforge.net/

[29] J.-F. Cordeau, "A branch-and-cut algorithm for the dial-a-ride problem," *Oper. Res.*, vol. 54, no. 3, pp. 573–586, 2006.



Lorenzo Abbatecola received the Laurea (Hons.) degree in electronics engineering from the Politecnico di Bari, Bari, Italy, in 2001. He is currently pursuing the Ph.D. degree in information engineering with the University of Trieste, Trieste, Italy.

He was an ICT Consultant with several Italian companies, such as Fiat SpA, Milan, Italy, and Telecom Italia SpA, Rome, Italy, dealing with data management and systems integration from 2001 to 2007. From 2007 to 2012, he was a Process and Software Engineer with SMA Spa, Bari, where he was involved in the field of transportation and logistics. In 2014, he was involved in the research project LAMRECOR, dealing with routing optimization for mail delivery over the Apulian region, Italy. His current research interests include supply chains and traffic networks, transportation and distribution logistics problems, and routing algorithms.

Maria Pia Fanti (M'92–SM'02) received the Laurea degree in electronics engineering from the University of Pisa, Pisa, Italy, in 1983.

She was a Visiting Researcher with the Rensselaer Polytechnic Institute, Troy, NY, USA, in 1999. Since 1983, she has been with the Department of Electrical and Information Engineering, Polytechnic of Bari, Bari, Italy, where she is currently a Full Professor of System and Control Engineering and the Chair of the Laboratory of Automation and Control. She has authored over 240 papers and two textbooks in her research topics. Her current research interests include discrete-event systems, Petri net, consensus protocols, management and modeling of complex systems, such as automated manufacturing systems, automatic guided vehicle systems, railway and traffic networks, supply chains, and healthcare systems.

Prof. Fanti was the General Chair of the Second IFAC Workshop on Dependable Control of Discrete Systems, the 2010 IEEE Workshop on Health Care Management, and the 2011 IEEE Conference on Automation Science and Engineering. She is an Editor of the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING and was an Associate Editor of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS. She is the Member-at-Large of the Board of Governors of the IEEE SMC Society, the Co-Chair of the Technical Committee on Discrete Event Systems of the IEEE SMC Society, the Chair of the Central and Southern Italy SMC Chapter, and the Chair of the Technical Committee on Automation in Logistics of the IEEE Robotics and Automation Society.

Agostino Marcello Mangini received the Laurea degree in electronics engineering and the Ph.D. degree in electrical engineering from the Politecnico di Bari, Bari, Italy, in 2003 and 2008, respectively.

He has been a Visiting Scholar with the University of Zaragoza, Zaragoza, Spain. He is currently a Post-Doctoral Fellow with the Department of Electrical and Information Engineering, Polytechnic of Bari. He has authored or co-authored over 80 printed publications. His current research interests include modeling, simulation, and control of discrete-event systems, Petri nets, supply chains and urban traffic networks, distribution and internal logistics, management of hazardous materials, management of drug distribution systems, and healthcare systems.

Dr. Mangini was on the program committees of the 2007–2015 IEEE International Conference on Systems, Man, and Cybernetics and the 2009 IFAC Workshop on Dependable Control of Discrete Systems. He is on the Editorial Board of the 2016 IEEE Conference on Automation Science and Engineering.

Walter Ukovich (M'08) received the Laurea degree in electronics engineering from the University of Trieste, Trieste, Italy, in 1970.

He was the Director of the Department of Electrical Engineering, Electronics and Computer Science, University of Trieste, from 2006 to 2010, where he was responsible for the curriculum in management engineering. He is currently a Full Professor of Operations Research with the University of Trieste. He has authored over 210 scientific papers in his research topics. His current research interests include optimization, logistics, transportation, production planning and control, and health management systems.

Prof. Ukovich has been a member of the Evaluation Boards of the National Research Council of Italy, the University of Trieste, the University of Macerata, Macerata, Italy, the Polytechnic of Turin, Turin, Italy, and the University Institute of Architecture of Venice, Venice, Italy. He is member of the Scientific Committee of the Italian Operations Research Council and the Co-Chair of the Technical Committee on Automation in Logistics of the IEEE Robotics and Automation Society. He was the Local Chair of the 2010 IEEE Workshop on Health Care Management, and the Local and Financial Chair of the 2011 IEEE Conference on Automation Science and Engineering.