A Decision Support System for a Multi-trip Vehicle Routing Problem with Trucks and Drivers Scheduling

Nilson F. M. Mendes and Manuel Iori

Departament of Sciences and Methods of Engineering, Università degli Studi di Modena e Reggio Emilia, Reggio Emilia, Italy

Keywords: Decision Support System, Optimization, Vehicle Routing, Scheduling.

Abstract: Many real-world transportation problems can be modeled as variants of the well-known vehicle routing problem (VRP), where a fleet of vehicles based at a central depot is used to deliver freight to clients at a minimum cost. Frequently, the problems defined in the VRP literature and the corresponding solution algorithms do not catch all the problem features incurred by the companies in their every-day activity, and further flexibility is needed during the decision process to make adjustments on the fly. In this paper, we present a decision support system developed for an Italian pharmaceutical distribution company to deal with a Multi-Trip VRP characterized by additional constraints and Truck and Driver Scheduling. The problem is solved in the software with a two-phase algorithm: the first phase consists of an Iterated Local Search metaheuristic that defines the vehicle routes, whereas the second phase invokes a mathematical model to assign trucks and drivers to the routes. The software allows, between the two phases, changes in the solution to better fit the company requirements. Computational results prove the effectiveness of the proposed method.

1 INTRODUCTION

Vehicle Routing Problems (VRP) are a traditional and well studied topic in Operations Research and Management Science. The process of defining efficient and convenient routes is one of the main concerns for a large number of companies, and it can lead to significant losses when it is not performed accurately. In such field, developing a good Decision Support System (DSS), capable of satisfying all requirements and covering all relevant characteristics of the real problem at hand, is not an easy task. Despite the vast literature dealing with different VRP types, finding a variant that exactly describes the operations of a company is also difficult. Most of the times, even state-of-the-art algorithms are not satisfactory in delivering good real routes, due to unrealistic assumptions, lack of information, too optimistic or deterministic approaches or simply because they are not flexible enough to provide alternatives when the best solution is not approved by the decision maker. Conscious of this situation, companies are always looking for alternatives that could give them a better (or more robust) approximation, of even hiring professionals or software houses to developed custom softwares that exactly fit their needs.

In this paper, we describe a branch of a DSS that we are developing for an Italian company specialized in the storage and distribution of pharmaceutical products. This branch provides a routing plan and a driver scheduling and assignment for each route. Furthermore, it provides a set of quantitative reports that allow one to better analyze the quality and fitness of the solution obtained. The problem we solve is a Rich Multi-Trip VRP with Driver and Vehicle Scheduling, where the term "rich" implies that the problem contains a number of additional constraints with respect to the basic VRP. The problem derives indeed from the union of a VRP with Time Windows (VRPTW), see, .g., (Solomon, 1984), and a Multi-Trip Vehicle Routing and Scheduling Problem (MTVRSP), see, e.g., (Brandao and Mercer, 1997), but also includes additional constraints defined by the distribution company. In short, the aim is to create a minimum-cost one-week routing plan to deliver products to a set of clients by using a heterogeneous fleet of vehicles based at a central depot, while satisfying clients service time windows, vehicle and drivers incompatibilities, driving regulation, and presence of intermediate depots.

The contributions of this work are multiple: we clearly describe an optimization problem derived

Mendes, N. and Iori, M.
A Decision Support System for a Multi-trip Vehicle Routing Problem with Trucks and Drivers Scheduling.
DOI: 10.5220/0009364403390349
In *Proceedings of the 22nd International Conference on Enterprise Information Systems (ICEIS 2020) - Volume 1*, pages 339-349
ISBN: 978-989-758-423-7

from a real-world distribution activity; we solve the problem by means of a two-phase algorithm; we present a software architecture that allows for an intuitive and quick man-in-the-middle approach to make the algorithm fully usable within a DSS; we present a large computational evaluation on a set of realistic instances; and we discuss how the approach can be replicated to solve other difficult VRP with operational constraints.

The remainder of the paper is organized as follows. In Section 2, a short review of the related literature is provided. In Section 3, a formal description of the problem and the company vision of it are given. Next, in Section 4, we describe how the system works and how the user participates to the solution process. In Section 5, we describe the solution approach and then, in Section 6, we computationally evaluate it on a case study.

2 LITERATURE REVIEW

As aforementioned, the problem that we study in this paper is a union of a VRPTW, see, e.g., (Solomon, 1984), and a MTVRSP, see, e.g., (Brandao and Mercer, 1997). The VRPTW has a broad and well-studied literature, and it is included in many MTVRSP problems. Consequently, we focus our short review on the MTVRSP literature. We refer the reader interested in VRPTW and other VRP variants to the books by (Golden et al., 2008) and (Toth and Vigo, 2014).

Despite the fact that they well model scenarios where short routes are common or desirable, MTVRSP are relatively new in the literature. While the traditional VRP was introduced sixty years ago by (Dantzig and Ramser, 1959), the multi-trip version appeared only thirty years after with (Fleischmann, 1990). In such article, the term used was not even“multi-trip”, but “VRP with multiple vehicle uses”. Other well known characteristics were also studied in that article, like time windows and heterogeneous fleet. The number of vehicles was also limited, making the use of the same vehicle several times more desirable.

Six years later (Taillard et al., 1996) proposed a three phase heuristic for the problem that started by generating a pool of reasonable routes (first phase) from which a subset was selected (second phase) and then assigned to feasible workdays according to a bin packing-like procedure (third phase).

The first explicit mention to the term “truck drivers” as a part of the problem name occurred, to the best of our knowledge, in (Brandao and Mercer, 1997). Such paper considered the driver maximum working hours in a day as well the needed breaks during a route. Other features, like unloading time, vehicle incompatibilities with some clients and the possibility to hire additional vehicles if the available ones were not enough, were also taken into account. To solve the resulting problem, they proposed a tabu search heuristic. They found good results for instances with approximately 20 vehicles, divided into two types, and 70 clients. The work was then pursued by the same authors in (Brandao and Mercer, 1998), where they discussed a simplified version of an algorithm used in a real-world application, and they reported reductions of about 5% on the delivery costs.

(Campbell and Savelsbergh, 2004) proposed an efficient insertion heuristic to the basic Vehicle Routing and Scheduling Problem (VRSP), a MTVRSP without multi-trips, having polynomial complexity. (Zäpfel and Bögl, 2008) presented a heuristic method to solve an MTVRSP with possibility of drivers outsourcing. (Cattaruzza et al., 2014a) proposed a memetic algorithm and an adaptation of the split procedure, see, e.g., (Duhamel et al., 2011), to segment a chromosome into a MTVRP solutions. (Azi et al., 2014) used an Adaptive Large Neighborhood Search (ALNS) algorithm to solve a variant of the MTVRSP problem where visits to some clients could be avoided.

Just a few papers in the literature developed exact methods to solve the MTVRSP. The first was (Azi et al., 2010), that made use of a column generation approach embedded with a branch-and-cut algorithm. A similar approach was also employed by (Macedo et al., 2011). A more robust method was provided by (Mingozzi et al., 2013), where the authors presented two set-partitioning formulations for the problem, one having a binary variable for each route and the other a binary variable for each schedule. Their solution procedure used bounding methods to reduced the original set of routes and schedules. (Hernandez et al., 2016) also used a similar strategy, but with less refinements in the space solution. In (Tang et al., 2015), a pickup and delivery problem inside an airport was modeled as a trip-chain-oriented set-partitioning, and then solved directly using CPLEX.

In 2006, a new regulation about truck drivers was created in Europe, aiming at giving better job conditions to the workers and rising the safety on the transportation activity. The EC N^o 561/2001 defined a large set of rules about drivers work regime like maximum duration of daily shifts, mandatory breaks, overtimes, minimum rest time between two shifts and weekly rests. Among the new rules we can highlight the minimum rest of 45 minutes after 4.5 hours of uninterrupted activity (that can be replaced by a 15 minutes break after a 30 minutes break within the 4.5 hours) , daily driving time of nine hours (that can be extended to 10 hours two times per week), at least 11 hours of daily rest and a maximum of 56 hours of weekly driving time. This regulation entered in force in April 2007, and right after that digital tacographs became mandatory in the European Union, allowing the authorities to check, a posteriori, the drivers working times. This new scenario motivated several studies where the driver was individually considered.

The first study considering these new constraints was presented, to the best of our knowledge, by (Goel, 2009). In this paper, a Large Neighborhood Search (LNS) is used to create a weekly schedule where all the rest and break rules are respected. To reduce the explored solution space, the authors did not considered overtimes and the 15 minutes more 30 minutes break possibility.

(Kok et al., 2010), on the other hand, was the first to consider all the rules presented in the EC N° 561/2001 regulation and in the Directive 2002/15/EC. To solve an MTVRP incorporating such features, they proposed a restricted dynamic programing framework where clients were sequentially added to the end of partial vehicle routes. Feasibility of such additions was controlled by extra state dimensions.

A few years later, (Drexl et al., 2013) proposed a two-stage method to solve a MTVRSP with the presence of relay stations where the drivers are allowed to change a vehicle. The first stage consists in solving a pickup-and-delivery problem, see, e.g., (Battarra et al., 2014) and (Doerner and Salazar-González, 2014) with time windows and relay stations, whereas the second consists in soling a VRPTW with multiple depots.

Two papers deal with a multi-commodity MTVRSP variant where some commodities cannot be transported together. The former is by (Battarra et al., 2009), who introduce the problem and proposes an adaptive guidance heuristic to solve it, whereas the latter is by (Cattaruzza et al., 2014b), who describe an ILS heuristic.

Vehicle routing and scheduling problems are studied in the context of home healthcare planning by (Algethami et al., 2017), that compares some operators in a genetic algorithm. In the same path (Algethami et al., 2019) proposes an adaptive multiple crossover to the problem.

Some most recent studies in the MTVRSP area are those by (Masmoudi et al., 2016) and (Benkebir et al., 2019). The former tests a set of metaheuristics to solve a multi-trip dial-a-ride problem, and the second proposes a hybrid method composed by a genetic algorithm and a local search for the MTVRSP. Others relevant study are (He and Li, 2019), where is described a two-echelon multi-trip VRP in a context of crop harvesting and (Babaee Tirkolaee et al., 2019) that describes a case study of a multi-trip VRP applied to an urban waste collection. Further information about the MTVRSP can be found in (Cattaruzza et al., 2016). We also refer to (Lahyani et al., 2015) as a recent review of the literature on rich VRP.

3 PROBLEM DESCRIPTION

The Multi-Trip Rich Vehicle Routing Problem with Truck and Driver Scheduling, henceforth refereed to as MTRVRPTDS, describes a one-week products delivery operation. In this problem, trucks and drivers are individually assigned to each route, considering driver regulation and operational constraints. The routes are created trying to minimize the distances, and must respect the client time windows. Deliveries can be anticipated (changing from the required day to the previous day) if it is allowed by the client and useful to improve the costs of the overall plan.

Formally, let D be a set of days in a week, F a set of storage facilities, C a set of clients, T a set of trucks and W a set of drivers. Each client $c \in C$ has a demand of O_{cd} pallets, on day $d \in D$, that must be delivered inside a time window $[\alpha_c, \beta_c]$. A service time of S_{cd} minutes is required to unload the delivery to client c in day d, independently of the truck internal organization of the cargo. Each vehicle $t \in T$ is driven by a unique driver $w \in W$. Vehicles are grouped in types, on the basis of their speed V_t and capacity Q_t. Drivers have different contracts and skills. They are allowed to work at most h hours by day, at most h^{week} hours per week, and they may drive only a subset $T_w \subseteq T$ of the available vehicles, for each $w \in W$.

A unique depot is given. This is due to the fact that the company centralizes all the deliveries of a region to that depot. The depot is supplied of pharmaceutical products by a regional entity, and it is expected that all the products are available in the moment in which they are required by the clients. Each route departs from the unique depot, passes through a subset of clients, and then returns to the depot. An agreement between drivers and the company limits the maximum duration of a route in 8 hours, less than the maximum daily shift allowed by the ECD 56/2006 which is 9 hours. It is also imposed that the total weekly working time cannot exceed 40 hours, and that the daily working extra time is limited to at most 20% of the maximum daily shift.

Some vehicles cannot be sent to visit some clients (because they are too big for the road or they do not have enough power to climb a hill). This is taken into account by considering a set R_t of clients that can be reached by vehicle $t \in T$.

Each vehicle and driver can be assigned to more than one route per day. This is possible if the next route starting time is after the first route ending time plus the time required to reload the truck or the driver have a break.

Additionally to the main depots there are, in some regions, intermediate warehouses that can be supplied by exclusive routes coming from the main depots. Those intermediate warehouses receive loads that will be delivered in the next day to the clients, working like a buffer. The use of an intermediate warehouse has a cost E_f, with $f \in F$ being an intermediate warehouse and F being the set of all intermediate warehouses.

A missing point about driver regulation that is not included in the problem definition concerns the mandatory breaks that drivers should do during the day and the rests between the days. They were not considered in the software specification, because they were considered too operational and quite unpredictable. As reported by one of the employees, as delays or changes in the routes can happen, it could be hard to stop at the moment defined by the specific sequence provided by the DSS, because the driver could be in a non-safe place. Beside this, it could trigger some drivers resistance due to the fact that some of them have already their preferences on where to stop in each place they use to visit. In any case, we discuss how these constraints could be included in our approach below. Regarding the rest between days, no route should start before 5 AM. If a driver arrives late at night, the responsible manager would change the scheduling to avoid this driver to pick a too early route in the next day. This type of on-the-fly changes are usual when using DSS to solve optimization problems.

4 SOFTWARE DATA FLOW

The usability is a very important feature considered by us to develop this software. It is commonly pointed out as one of the main factors for the success of a software in a company. A poor usability can make DSS with advanced algorithms and analytic tools unused due to the resistance of the decision makers to learn and deal with the system complexity.

To ensure that our DSS is used by the company, we decided to create a friendly web interface to deal with the process of loading, visualizing and control of optimization inputs. More than only providing an easy-to-learn tool to run an algorithm, this interface was thought to allow the decision maker to be part of the solving process and have a clear overview of solution quality. The application described in this paper is a module inside a larger software belonging to the same project. This module is divided into two parts, following the strategy proposed by us to solve the problem.

The flow starts with the rich VRP input load. This input consists in three files, one describing the clients, with all the information about time windows, demands, service times and vehicle restrictions; the second describing the type of vehicles available with its respective average speeds, capacity and number of units available, and finally the third with the parameters.

The distances between the clients can be informed by the user in the client file or evaluated in the system using the LibOSM (https://github.com/Marcussacapuces91/LibOsm/) that is part of the Open Street Maps ecosystem and Lemon (https://lemon.cs.elte.hu/trac/lemon), from COIN-OR. These libraries together make possible to calculate real road distance between any pair of points that corresponds a valid address in a given region using only a local geo-spatial database. In this way, it is possible to get all the information needed without the use of Internet or accessing external web services.

After all the data are loaded (or calculated, if we consider distances), the user can start the optimization with a click. To solve this part of the problem we invoke a heuristic algorithm that takes approximately 1 to 3 minutes to converge with instances involving around 200 clients. Such heuristic is described in detail in (Kramer et al., 2019), and is sketched below in the next section. The results obtained are a set of product transfers from a depot to an intermediate warehouse and a set of delivery routes departing from the main depot. Each route or transfer has a departure time and receives a vehicle type assigned to it, but not information about the driver.

Before proceeding to the next step, the decision maker can adjust the solution obtained by changing the departure time, the vehicle type assigned, the clients in the route and the visit order. After each change, a solution automatic validation is performed. If a change leads to an infeasible solution, a rollback procedure is done and the user is informed with a pop-up.

To plan an individual driver and truck to each route the user can proceed in two ways. The first way is to load route information using Excel files like in the previous phase. This method is useful when some other persons or systems created the routes, so it is possible to get some improvement by better using the resources available. The second way is to load the first phase solution as input to the second part, and this can be done with just one click. Next, the user just needs to insert the information about the drivers and the details about each vehicle.

The driver and truck assignment is calculated by solving a mathematical model using a Mixed Integer Linear Programming (MILP) solver. As in the first phase, changes in the solution provided by the system are allowed and checked, as already described. Once all these steps are finished, the solution remains stored in the software database and can be visualized in a dashboard or downloaded in an Excel file.

A last important point to highlight is the tolerance of the DSS with respect to infeasible solutions. In real problems, not satisfying all constraints is common. However, it is not interesting for a decision maker to simply have no answer after it called a solver. Considering this, we show any solution obtained at the end of optimization, treating the typical sources of infeasibility as penalizations in the objective functions, but warning the user about that. A typical situation of this type occurs when not enough trucks or drivers are available for the daily deliveries. In this case, the model incurs in a penalty, the solution is however returned, and the decision maker knows which routes can be directly performed by the company and which ones should be postponed to the next day or given to a third-party logistic operator. The tolerance with infeasible solution is not extended to infeasibilities inserted by the user through solution edition. As aforementioned, no valid modifications are undone to avoid deteriorate a feasible or almost feasible solution.

5 SOLUTION APPROACH

To solve the MTRVRPTDS, we use a two-phase approach. In the first phase, we use the Multi-Start Iterated Local Search (MS-ILS) metaheuristic developed by (Kramer et al., 2019). This MS-ILS was originally developed to solve the same VRP we face in this paper, but without considering multiple-trips neither the presence of a limited umber of trucks and drivers. We thus modified this algorithm in the way described below in Section 5.1 to fit with the new characteristics. In the second phase, the routes created in the previous phase are given in input to a new mathematical model, described in Section 5.2, that assigns drivers and trucks to the routes and defines the effective departing time of each route.

5.1 Multi-start ILS Heuristic

The subproblem solved in the first phase of the MTRVRPTDS defines a set of delivery routes and product transfers from main depots to intermediate warehouses. In this phase, trucks and drivers are not individualized, but we just define how many vehicles of each capacity and average speed are used. The deliveries must be done by satisfying the client time windows and some of them can be anticipated to the previous day at the expenses of opening an auxiliary depot. To solve this problem, we used an adapted version of the MS-ILS by (Kramer et al., 2019), where a penalty is added in the cost function whenever the number of vehicles of a certain type used is greater than the number of available vehicles.

The algorithm can be briefly described as follows. At each iteration, a constructive method creates an initial route connecting the depot to a hospital (that are the clients with larger demands in our instances) and associate the largest allowed vehicle to the route. After this is done for all the hospitals, the remaining clients are inserted in the routes created following a lowest-cost-increase criterion. Time windows violations are accepted, but penalized.

After creating the initial solution, the algorithm starts the ILS loop. In this loop, a Randomized Variable Neighrborhood Descent (RVND) algorithm is used as local search procedure. The RVND selects, at each iteration, an inter-route neighborhood (from a list of four) and executes it. If the solution is not improved, then the neighborhood is removed from the list and the algorithm tries another one. Otherwise, the list is reinitialized and the algorithm tries to improve the current solution using one of three possible intra-route neighborhoods. When the list of available inter-routes neighborhood becomes empty (i.e., after four not improvement iterations) the method stops. After the local search has been performed, a perturbation phase is invoked. In this step, a local search is chosen randomly (from a list of three) to randomly modify the solution. The ILS method is iterated until a given number of iterations without improvements is reached (in our settings, 20 iterations). The multi-start executes the ILS 20 times.

5.2 Mathematical Model

The subproblem solved in the second phase of the MTRVRPTDS defines which truck and driver must execute a route and defines the departure and arrival times at each client, considering the truck average speed, the service start time and the time windows. In some cases, no truck and driver are assigned to a route due to limited resources. In this case, a penalty is applied in the objective function. The penalty roughly corresponds to the cost of assigning the route to an external distribution company. We also apply penalties when a driver works more than her maximum daily working hours or compatibilities are not satisfied. The maximum overtime is modeled as a hard constraint, as well the weekly maximum working hours.

The MILP model uses the following parameters:

- DC_w - Cost of driver $w \in W$
- $DOTC_w$ - Overtime hour cost of driver $w \in W$
- PR - Penalty for route not assigned
- TTP - Total daily time (1440)
- CST_c^r - Service time of client $c \in C$ in the route $r \in R$
- TD_{ab}^r - Total distance between points a and b in the route $r \in R$
- TD_r - Total distance on route $r \in R$
- TST_r - Total service time on route $r \in R$
- TAS_t - Truck $t \in T$ average speed
- (TWB_c, TWE_c) - Limits of time windows of client $c \in C$
- E_r - Set of routes segments in the route $r \in R$
- C_r - Set of clients in the route $r \in R$
- D_r - Day in which the route $r \in R$ is executed
- LT - Truck loading time
- R_d - Routes in the day $d \in D$
- DMH_w - Daily maximum working hours of driver $w \in W$
- WMH_w - Weekly maximum working hours of driver $w \in W$
- FWC_{wc} - Equals to 1 if it is strictly forbidden to assign the driver $w \in W$ to routes containing the client c, equals to 0 otherwise
- FTC_{tc} - Equals to 1 if it is strictly forbidden to assign the truck $t \in T$ to routes containing the client c, equals to 0 otherwise
- FTW_{tw} - Equals to 1 if it is strictly forbidden to assign the driver $w \in W$ and the truck $t \in T$ to the same route, equals to 0 otherwise
- $SOFT\_FWC_{wc}$ - Equals to 1 if it is not desirable to assign the driver $w \in W$ to routes containing the client c, equals to 0 otherwise
- $SOFT\_FTC_{tc}$ - Equals to 1 if it is not desirable to assign the truck $t \in T$ to routes containing the client c, equals to 0 otherwise
- $SOFT\_FTW_{tw}$ - Equals to 1 if it is not desirable to assign the driver $w \in W$ and the truck $t \in T$ to the same route, equals to 0 otherwise

The equations (1) to (31) defines the model proposed. All the variables that represents time instants or intervals as well the temporal parameters are expressed in minutes. Every time the characters f and l appear as a client-index in the model, they represent the first and last clients of the route.

$$\begin{aligned} Min \sum_{w \in W} \sum_{d \in D} (sig_{wd} * DC_w + DOTC_w * pun_{wd}) \\ + PR * ncr_r + penComp \end{aligned} \tag{1}$$

Subject to

$$\sum_{w \in W} x_{wr} + ncr_r = 1 \quad r \in R \tag{2}$$

$$\sum_{i \in T} y_{ir} + ncr_r = 1 \quad r \in R \tag{3}$$

$$max(TWB_c, ac_{cr}) \leq bc_{ir} \quad \forall r \in R \quad c \in C_r \tag{4}$$

$$\begin{aligned} bc_{cr} + CST_c^r \leq dc_{cr} + ncr_r * TTP \\ \forall r \in R, \ \ c \in C_r \end{aligned} \tag{5}$$

$$\begin{aligned} dc_{cr} + \sum_{i \in T} y_{ir} * TD_{c1,c2}^r / TAS_i \leq ac_{c2,r} \\ \forall r \in R, \ \ (c1, c2) \in E_r \end{aligned} \tag{6}$$

$$\begin{aligned} dc_{dep,r} - (bc_{ls} + CST_l^s + \sum_{i \in T} y_{is} * TD_{l,dep}^s / TAS_i) \\ \leq (ro_{rs} - 1) * TTP + LT \quad \forall r, s \in R | D_r \neq D_s \end{aligned} \tag{7}$$

$$x_{wr} + x_{ws} \leq ro_{rs} + ro_{sr} + 1 \quad \forall r, s \in R, \ w \in W \tag{8}$$

$$y_{ir} + y_i s \leq ro_{rs} + ro_{sr} + 1 \quad \forall r, s \in R, \ i \in T \tag{9}$$

$$\sum_{r \in R_d} x_{wr} \leq |R| * sig_{wd} \quad \forall w \in W, \ \ d \in D \tag{10}$$

$$\begin{aligned} Ua_{wd} \leq dc_{fr} + TTP * (1 - x_{wr}) \\ \forall w \in W, \ \ d \in D \ \ , r \in R_d \end{aligned} \tag{11}$$

$$\begin{aligned} Ca_{wd} + TTP * (1 - x_{wr}) \geq \\ dc_{lr} + \sum_{i \in T} y_{ir} * TD_{l,dep}^r / TAS_i \\ \forall w \in W, \forall \ \ d \in D \ \ , r \in R_d \end{aligned} \tag{12}$$

$$\begin{aligned} Ca_{wd} - Ua_{wd} \leq DMH_w + pun_{wd} \\ \forall w \in W, \ \ d \in D \end{aligned} \tag{13}$$

$$\sum_{d \in D} (Ca_{wd} - Ua_{wd}) \leq WMH_w \quad \forall w \in W \tag{14}$$

$$\begin{aligned} x_{wr} \leq (1 - FWC_{wc}) + \\ (1 - FWC_{wc}) * wcp_{wc} - SOFT\_FWC_{wc} \\ \forall w \in W,\ c \in C_r \end{aligned} \tag{15}$$

$$\begin{aligned} y_{ir} \leq (1 - FTC_{ic}) + \\ (1 - FTC_{ic}) * tcp_{ic} - SOFT\_FTC_{ic} \\ \forall i \in T \ \ , c \in C_r \end{aligned} \tag{16}$$

$$\begin{aligned} x_{wr} + y_{ir} \leq (2 - FTW_{iw}) + \\ (2 - FTW_{iw}) * twp_{iw} - 2 * SOFT\_FTW_{iw} \\ \forall w \in W, i \in T, r \in R \end{aligned} \tag{17}$$

$$\begin{aligned} penComp = M * (\sum_{i \in T} \sum_{c \in C} tcp_{ic} + \\ \sum_{w \in W} \sum_{c \in C} wcp_{wc} + \sum_{w \in W} \sum_{i \in T} twp_{iw}) \end{aligned} \tag{18}$$

$$\begin{aligned} dc_{dep,r} + \sum_{i \in T} y_{ir} * (TD_r - TD^r_{l,dep}) / TAS_i \\ + TST_r \leq ac_{lr} + CST^r_l + TTP * ncr_r \ \ \forall r \in R \end{aligned} \tag{19}$$

$$\begin{aligned} dc_{dep,r} + \sum_{i \in T} y_{ir} * (TD^r_{dep,f}) / TAS_r \\ \geq TWB_f - TTP * ncr_r \quad \forall r \in R \end{aligned} \tag{20}$$

$$x_{wr} \in 0,1 \ \ \forall w \in W, r \in R \tag{21}$$

$$y_{tr} \in 0,1 \ \ \forall t \in T, r \in R \tag{22}$$

$$ncr_r \in 0,1 \ \ \forall r \in R \tag{23}$$

$$0 \leq bc_{cr} \leq TWE_c - CST^r_c \quad \forall r \in R, c \in C_r \tag{24}$$

$$0 \leq ac_{cr} \leq TWE_c - CST^r_c \quad \forall r \in R, c \in C_r \tag{25}$$

$$0 \leq dc_{cr} \leq TWE_c \quad \forall r \in R, c \in C_r \tag{26}$$

$$Ua_{wd}, Ca_{wd} \geq 0 \quad \forall w \in W, d \in D \tag{27}$$

$$0 \leq pun_{wd} \leq 0.2 * DMH_w \quad \forall w \in W, d \in D \tag{28}$$

$$wcp_{wc} \geq 0 \quad \forall w \in W, c \in C \tag{29}$$

$$tcp_{tc} \geq 0 \quad \forall t \in T, c \in C \tag{30}$$

$$twp_{tw} \geq 0 \quad \forall t \in T, w \in W \tag{31}$$

The binary variable x_{wr} defines if the driver $w \in W$ is assigned to the route $r \in R$. Similarly, y_{ir} is a binary variable that defines if the truck $i \in T$ is assigned to the route $r \in R$. In the model implementation, no y variable is created when the truck capacity is lower than the total demand in the route. The variables ac_{cr}, bc_{cr} and dc_{cr} define, respectively, the arrival time, service begin time and departure time for client $c \in C$ and route $r \in R$. The binary variable ro_{rs} defines if the routes $r, s \in R$ can be assigned to the same driver and truck. Routes in different days have no restrictions of this kind. The variables Ua_{wd} and Ca_{wd} represent the first departure and last arrival time for driver $w \in W$ in day $d \in D$.

Some variables are defined to describe situations where penalties must be applied. The variable ncr_r represents a non-executed route, pun_{wd} represents instead the overtime of driver $w \in W$ in day $d \in D$. The variables, wcp_{wc}, tcp_{ic} and twp_{tw} are, respectively, binary variables that represent non-desirable driver/client, truck/client and truck/driver assignments. Finally, *penComp* simply sums up all the non-desirable assignment penalties.

Constraints (2) and (3) define that, in order to execute a route, we should assign a driver and a truck, otherwise the ncr_r variable corresponding to that route would be activated. Constraints (4) to (6) define the minimum begin service time, client departure time and client arrival time, respectively. In (7), we verify if it is possible to assign the same route to the same pair driver/client. The two following constraints avoid or permit it, according to the value of variables ro_{rs} and ro_{sr}. Constraints (10) check if a driver is used in the day. In constraints (11) to (14), the driver working hours (including eventual pauses) are calculated and limited. Constraints (15) to (18) check the assignment incompatibilities. Finally, (19) defines a lower bound to arrive in the last client and (20) defines an upper bound for the departure of a route. The remaining constraints ensure variable domains.

6 CASE STUDY

In this section, we present the computational results that we obtained on a case study. The aim of the tests we performed was to evaluate the DSS performance in finding good solutions. We used a PC equipped with a processor Intel Core i5-7200 with 2.5 GHz, Windows 10 and 8GB of RAM. The heuristic was implemented using C++ and the model using JuMP library of Julia language. To solve the model, we used the IBM MILP solver CPLEX 12.8.

The instances we used are taken from a realistic scenario originating in the Italian region of Basilicata. All the instances have some common characteristics, such as 187 clients, 2 types of vehicles, truck average speeds (40 km/h for the larger vehicle type and 60 km/h for the smaller vehicle type), the daily demands, the assignment resctrictions, 8 hours of maximum shift duration, and 60 minutes between two consecutive routes assigned to the same driver/truck. In Table 1 we report some details on number of customers and daily demands.

Table 1: Number of Clients and Total Demands per Day.

Day	Mon	Tue	Wed	Thu	Fri	Sat
N. of clients	37	37	44	38	51	2
Total demand	530	490	560	540	630	60

We created instances by attempting variations in the number of vehicles of each type, capacity of vehicles, time window size and maximum number of clients per route. All those variations generated a total of 40 instances, divided in 5 blocks of 8 instances each. All tests were executed like in a standard use of the software, as described in Section 4. To give a user perspective of the results we limited the maximum run time of the MILP solver to 30 minutes.

Table 2 summarizes the main results we obtained. The table reports the identification number of each instance (ID), the truck capacity (TC), the time window size (TW), the maximum number of clients per route (MCR) and the number of trucks per type (N. of trucks).

For what the concerns the results obtained by the optimization method, we highlight in column NR the number of routes generated by the MS-ILS heuristic adopted in the first phase of our algorithm, and in column Km the total distance of such routes. The first phase required between one and two minutes to solve any of the instances in the table. We could not find a clear correlation between instance configuration and run time of the algorithm. Regarding the solution quality, we observe that the number of routes generated by the first phase algorithm does not have a significant correlation with number of vehicles used and total distance (correlations -0.008 and 0.03, respectively). Even in instances with a total of 10 trucks the number of routes does not change significantly. On the other hand, fleet total capacity creates a larger variation on number of routes as well as the total distance run (correlations -0.73 and -0.71, respectively). As we can see, the number of routes and total distance grow almost equally as the fleet capacity reduce.

Average distance by route is 156 km, with a small standard deviation of only 4.7 km. This means all routes can be traveled in less than four hours, even with the slowest vehicle. This is an advantage in small-sized time window scenarios and makes the problem regarding breaks along the day less relevant.

In the second phase, the model was able to find assignments to all routes for about 30% of the instances. In another 42% of instances, only one or two routes were not assigned. In a real life operation this kind of solutions is not a major concern if visualized in advance. The decision maker can improve these solutions by contacting an external truck and driver, or delaying some deliveries.

However, cases where a higher number of non-assigned routes (as for instances 1 and 2 in block E) are more critical. Those cases could be caused by problems like deliveries imbalance, non appropriated fleet size or worse, a bad warehouse location. On the other hand, it could represent a lack o efficiency of the algorithm in building routes and schedules with the available resources, which can be verified with a deeper solution analysis.

Table 2 also highlights a low number of instances solved to optimality (14 out of 40) and some large gaps. The gap increases when the fleet capacity is reduced and the time windows get tighter. As the gaps were not directly connected with the quality of the solutions, we looked for another factors that could be interfering in the convergence of model solving. To investigate changes that could provide a better performance in the proposed method, we tested a modified version of the model in which variable sig_{wd} as well constraints 10 were removed.

The results that we obtained with this simplified model are shown in Table 3. The changes we made on the model were useful in improving the solution convergence, aa all the instances were solved to the proven optimality. All but one of instances were solved in less than one minute, and in many cases the gap between the objective functions found on regular and modified versions were lower than 10%. The numbers of non-assigned routes in this model version were the same as those found with the original model. This makes us conclude that the simplified model is a good compromise between the representation of the real-world problem and the need fo a quick and effective solution convergence.

Table 2: Instance Variable Parameters and Main Data about Obtained Solutions. Abbreviations : TC - Truck Capacity, TW - Time Window, MCR - Maximum Clients per Route, NR - Number of Routes, Time - Model Solving Run Time, UB - Model Objective Function, LB - Lower Bound, NAR - Non-Assigned Routes.

	Parameters				Results						
ID	TC	TW	MCR	N. of trucks	NR	Km	Time	UB	LB	Gap	NAR
1A				3 - 5	48	7635	1845	15435	15084	2.32	3
2A				2 - 6	47	7610	1857	15469	15133	2.2	3
3A				4 - 4	48	7587	1837	15435	15084	2.3	3
4A	70/60	6-18	6	5 - 3	49	7699	1813	5522	5176	6.7	1
5A				6 - 2	49	7551	1857	10522	10084	4.3	2
6A				7 - 3	49	7615	1839	15380	15107	1.8	3
7A				3 - 7	49	7632	1901	10380	10084	2.9	2
8A				5 - 5	49	7632	1833	10380	10084	2.9	2
1B				3 - 5	58	9197	1834	1198	751	5.9	0
2B				2 - 6	60	9388	1907	11251	5684	97.9	2
3B				4 - 4	58	9282	1833	5618	5212	7.8	1
4B	70/40	6-18	6	5 - 3	59	9163	1831	5671	5196	9.1	1
5B				6 - 2	56	9109	1814	671	212	216.5	0
6B				7 - 3	58	9512	1849	476	210	126.6	0
7B				3 - 7	58	9167	1839	1676	1374	21.2	0
8B				5 - 5	58	9237	1832	5444	5105	6.6	1
1C				3 - 5	69	10179	1814	1337	814	64.2	0
2C				2 - 6	69	10254	1812	6337	684	826.4	1
3C				4 - 4	68	10131	1812	1241	127	87.7	0
4C	50/40	6-18	6	5 - 3	70	10365	1813	1337	120	101.4	0
5C				6 - 2	70	10365	1813	1390	117	108.8	0
6C				7 - 3	69	10239	1812	440	84	423.8	0
7C				3 - 7	69	10402	1813	551	110	441.0	0
8C				5 - 5	69	10383	1838	508	84	504.7	0
1D				5 - 3	46	7409	1848	11284	10748	5.0	2
2D				2 - 6	47	7314	1833	11284	10748	5.0	2
3D				4 - 4	47	7359	115	11284	11284	0	2
4D	70/60	7-17	8	5 - 3	46	7402	106	16251	16251	0	3
5D				6 - 2	47	7338	1840	16347	15148	7.9	3
6D				7 - 3	46	7442	355	10476	10476	0	2
7D				3 - 7	47	7466	133	11076	11076	0	2
8D				5 - 5	47	7454	194	5476	5476	0	1
1E				5 - 3	46	7307	48	31205	31205	0	6
2E				2 - 6	47	7456	54	36453	34653	0	7
3E				4 - 4	46	7336	54	16559	16559	0	3
4E	70/60	7 - 13 and 11 - 18	8	5 - 3	47	7520	56	16400	16400	0	2
5E				6 - 2	46	7476	67	11443	11443	0	2
6E				7 - 3	47	7465	67	15637	15637	0	3
7E				3 - 7	47	7444	70	21301	21301	0	4
8E				5 - 5	46	7635	69	733	733	0	0

7 CONCLUSIONS

In this paper, we presented a decision support system to help decision makers in the solution of real cases of a Multi-Trip Rich Vehicle Routing Problem with Truck and Driver Scheduling, a problem where good delivery routes need to be created and then matched with the available trucks and drivers. We proposed a two-phase heuristic procedure, in which the first phase is an adaptation of a metaheuristic from the literature, and the second phase consists of a mathematical model.

Table 3: Results Obtained Solving the Model without Drivers Fixed Costs. Abbreviations: Time - Model Solving Run Time in Seconds, UB - Model Objective Function, LB - Lower Bound, NAR - Non-Assigned Routes.

	Results				
ID	Time	UB	LB	Gap	NAR
A1	16	15000	15000	0	3
A2	15	15000	15000	0	3
A3	13	15000	15000	0	3
A4	15	5000	5000	0	1
A5	14	10000	10000	0	2
A6	15	15000	15000	0	3
A7	14	10000	10000	0	2
A8	15	10000	10000	0	2
B1	20	600	600	0	0
B2	20	10600	10600	0	2
B3	16	5000	5000	0	1
B4	23	5000	5000	0	1
B5	16	600	600	0	0
B6	22	0	0	0	0
B7	19	1200	1200	0	0
B8	14	5000	5000	0	1
C1	21	600	600	0	0
C2	35	5600	5600	0	1
C3	32	0	0	0	0
C4	32	0	0	0	0
C5	211	0	0	0	0
C6	18	0	0	0	0
C7	17	0	0	0	0
C8	22	0	0	0	0
D1	20	10600	10600	0	2
D2	17	10600	10600	0	2
D3	28	10600	10600	0	2
D4	37	15600	15600	0	3
D5	17	15600	15600	0	3
D6	16	10000	10000	0	2
D7	22	10600	10600	0	2
D8	25	5000	5000	0	1
E1	15	30600	30600	0	6
E2	25	35600	35000	0	7
E3	26	15600	15600	0	3
E4	27	15600	15600	0	2
E5	72	10600	10600	0	2
E6	18	15000	15000	0	3
E7	14	20600	20600	0	4
E8	15	0	0	0	0

Extensive computational experiments were performed on realistic instances. We could observe that the system had troubles in identifying good solutions in very restricted scenarios, but it could consistently produce good quality solutions in other reasonable scenarios. For such scenarios, we could also note that the algorithm had a regular performance behavior, and this is an important feature to make the user trust the software. The run time was most of mostly low, satisfying the requirements of the system without compromising the solution qualities.

Future research directions will be concentrated on adapting and testing the current approach in more flexible and general scenarios. For example, when deliveries can be done in the next day, vehicles or drivers are not available in some days of the week, or when different truck average speeds must be used depending on the fact that the vehicle is in an urban area or not. Considering the user experience, we plan to make visible to the decision maker data about road blocks, tolls, and information on client satisfaction, to help her in the evaluation of eventual route changes. A synchronization with the warehouse operation software is also being evaluated to improve the allocation of workers to recover products and load trucks. We also plan to replace the mathematical model with a quick and effective metaheuristic, so as to be able to provide in quick time good-quality problem solutions.

ACKNOWLEDGEMENTS

We thank the University of Modena and Reggio Emilia (Italy) for the financial support to this work provided with the grant FAR 2018 "Analysis and optimization of healthcare and pharmaceutical logistic processes".

REFERENCES

Algethami, H., Landa-Silva, D., and Martinez-Gavara, A. (2017). Selecting genetic operators to maximise preference satisfaction in a workforce scheduling and routing problem. In *Proceedings of the 6th International Conference on Operations Research and Enterprise Systems*, pages 416–423.

Algethami, H., Martínez-Gavara, A., and Landa-Silva, D. (2019). Adaptive multiple crossover genetic algorithm to solve workforce scheduling and routing problem. *Journal of Heuristics*, 25(4-5):753–792.

Azi, N., Gendreau, M., and Potvin, J.-Y. (2010). An exact algorithm for a vehicle routing problem with time windows and multiple use of vehicles. *European Journal of Operational Research*, 202(3):756–763.

Azi, N., Gendreau, M., and Potvin, J.-Y. (2014). An adaptive large neighborhood search for a vehicle routing problem with multiple routes. *Computers & Operations Research*, 41:167–173.

Babaee Tirkolaee, E., Abbasian, P., Soltani, M., and Ghaffarian, S. A. (2019). Developing an applied algorithm for multi-trip vehicle routing problem with time windows in urban waste collection: A case study. *Waste Management & Research*, 37(1_suppl):4–13.

Battarra, M., Cordeau, J.-F., and Iori, M. (2014). Pickup-and-delivery problems for goods transportation. In Toth, P. and Vigo, D., editors, *The Vehicle Routing Problem*, Monographs on Discrete Mathematics and Applications, pages 161–191. SIAM, Philadelphia.

Battarra, M., Monaci, M., and Vigo, D. (2009). An adaptive guidance approach for the heuristic solution of a minimum multiple trip vehicle routing problem. *Computers & Operations Research*, 36(11):3041 – 3050.

Benkebir, N., Pouliquen, M. L., Trévien, J.-F., Bounceur, A., Euler, R., Pardiac, E., and Sevaux, M. (2019). On a multi-trip vehicle routing problem with time windows integrating european and french driver regulations. *Journal on Vehicle Routing Algorithms*, 2(1):55–74.

Brandao, J. and Mercer, A. (1997). A tabu search algorithm for the multi-trip vehicle routing and scheduling problem. *European journal of operational research*, 100(1):180–191.

Brandao, J. and Mercer, A. (1998). The multi-trip vehicle routing problem. *Journal of the Operational research society*, 49(8):799–805.

Campbell, A. M. and Savelsbergh, M. (2004). Efficient insertion heuristics for vehicle routing and scheduling problems. *Transportation science*, 38(3):369–378.

Cattaruzza, D., Absi, N., and Feillet, D. (2016). Vehicle routing problems with multiple trips. *4OR*, 14(3):223–259.

Cattaruzza, D., Absi, N., Feillet, D., and Vidal, T. (2014a). A memetic algorithm for the multi trip vehicle routing problem. *European Journal of Operational Research*, 236(3):833–848.

Cattaruzza, D., Absi, N., Feillet, D., and Vigo, D. (2014b). An iterated local search for the multi-commodity multi-trip vehicle routing problem with time windows. *Computers & Operations Research*, 51:257 – 267.

Dantzig, G. B. and Ramser, J. H. (1959). The truck dispatching problem. *Management science*, 6(1):80–91.

Doerner, K. and Salazar-González, J. (2014). Pickup-and-delivery problems for people transportation. In Toth, P. and Vigo, D., editors, *The Vehicle Routing Problem*, Monographs on Discrete Mathematics and Applications, pages 161–191. SIAM, Philadelphia.

Drexl, M., Rieck, J., Sigl, T., and Press, B. (2013). Simultaneous vehicle and crew routing and scheduling for partial- and full-load long-distance road transport. *Business Research*, 6(2):242–264.

Duhamel, C., Lacomme, P., and Prodhon, C. (2011). Efficient frameworks for greedy split and new depth first search split procedures for routing problems. *Computers & Operations Research*, 38(4):723–739.

Fleischmann, B. (1990). The vehicle routing problem with multiple use of vehicles. *Fachbereich Wirtschaftswissenschaften, Universität Hamburg*.

Goel, A. (2009). Vehicle scheduling and routing with drivers' working hours. *Transportation Science*, 43(1):17–26.

Golden, B., Raghavan, S., and Wasil, E., editors (2008). *The Vehicle Routing Problem: Latest Advances and New Challenges*. Operations Research/Computer Science Interfaces Series. Springer.

He, P. and Li, J. (2019). The two-echelon multi-trip vehicle routing problem with dynamic satellites for crop harvesting and transportation. *Applied Soft Computing*, 77:387–398.

Hernandez, F., Feillet, D., Giroudeau, R., and Naud, O. (2016). Branch-and-price algorithms for the solution of the multi-trip vehicle routing problem with time windows. *European Journal of Operational Research*, 249(2):551–559.

Kok, A. L., Meyer, C. M., Kopfer, H., and Schutten, J. M. J. (2010). A dynamic programming heuristic for the vehicle routing problem with time windows and european community social legislation. *Transportation Science*, 44(4):442–454.

Kramer, R., Cordeau, J.-F., and Iori, M. (2019). Rich vehicle routing with auxiliary depots and anticipated deliveries: An application to pharmaceutical distribution. *Transportation Research Part E: Logistics and Transportation Review*, 129:162–174.

Lahyani, R., Khemakhem, M., and Semet, F. (2015). Rich vehicle routing problems: From a taxonomy to a definition. *European Journal of Operational Research*, 241(1):1–14.

Macedo, R., Alves, C., de Carvalho, J. V., Clautiaux, F., and Hanafi, S. (2011). Solving the vehicle routing problem with time windows and multiple routes exactly using a pseudo-polynomial model. *European Journal of Operational Research*, 214(3):536 – 545.

Masmoudi, M. A., Hosny, M., Braekers, K., and Dammak, A. (2016). Three effective metaheuristics to solve the multi-depot multi-trip heterogeneous dial-a-ride problem. *Transportation Research Part E: Logistics and Transportation Review*, 96:60 – 80.

Mingozzi, A., Roberti, R., and Toth, P. (2013). An exact algorithm for the multitrip vehicle routing problem. *INFORMS Journal on Computing*, 25(2):193–207.

Solomon, M. M. (1984). Vehicle routing and scheduling with time window constraints: Models and algorithms (heuristics).

Taillard, É. D., Laporte, G., and Gendreau, M. (1996). Vehicle routeing with multiple use of vehicles. *Journal of the Operational research society*, 47(8):1065–1070.

Tang, J., Yu, Y., and Li, J. (2015). An exact algorithm for the multi-trip vehicle routing and scheduling problem of pickup and delivery of customers to the airport. *Transportation Research Part E: Logistics and Transportation Review*, 73:114–132.

Toth, P. and Vigo, D., editors (2014). *Vehicle Routing: Problems, methods, and applications*. SIAM, Philadelphia, 2nd edition.

Zäpfel, G. and Bögl, M. (2008). Multi-period vehicle routing and crew scheduling with outsourcing options. *International Journal of Production Economics*, 113(2):980–996.