

## A hybrid expert system, clustering and ant colony optimization approach for scheduling and routing problem in courier services

Eduyn López-Santana<sup>a\*</sup>, William Camilo Rodríguez-Vásquez<sup>a</sup> and Germán Méndez-Giraldo<sup>a</sup>

<sup>a</sup>*Sistemas Expertos y Simulación (SES), Facultad de Ingeniería, Universidad Distrital Francisco José de Caldas, Bogotá, Colombia*

### CHRONICLE

#### Article history:

Received June 29 2017  
Received in Revised Format  
July 1 2017  
Accepted August 10 2017  
Available online  
August 10 2017

#### Keywords:

*Courier services*  
*Clustering*  
*Expert system*  
*Routing*  
*Scheduling*

### ABSTRACT

This paper focuses on the problem of scheduling and routing workers in a courier service to deliver packages for a set of geographically distributed customers and, on a specific date and time window. The crew of workers has a limited capacity and a time window that represents their labor length. The problem deals with a combination of multiples variants of the vehicle routing problem as capacity, multiple periods, time windows, due dates and distance as constraints. Since in the courier services the demands could be of hundreds or thousands of packages to be delivered, the problem is computationally unmanageable. We present a three-phase solution approach. In the first phase, a scheduling model determines the visit date for each customer in the planning horizon by considering the release date, due date to visit and travel times. We use an expert system based on the know-how of the courier service, which uses an inference engine that works as a rule interpreter. In the second phase, a clustering model assigns, for each period, customers to workers according to the travel times, maximum load capacity and customer's time windows. We use a centroid based and sweep algorithms to solve the resulted problem. Finally, in the third phase, a routing model finds the order in which each worker will visit all customers taking into account their time windows and worker's available time. To solve the routing problem we use an Ant Colony Optimization metaheuristic. We present some numerical results using a case study, in which the proposed method of this paper finds better results in comparison with the current method used in the case study.

© 2018 Growing Science Ltd. All rights reserved

## 1. Introduction

The courier services are dedicated to pick up/delivery packages and/or documents that are sent from people to other people with the target of a faster and secure pick up/delivery. To manage these service systems there are different decision-making levels. In strategic level (or long-range term), the decisions state the objectives, resources, and policies of the organization in a long planning horizon, e.g., a typical problem consists in forecasting the kind of services to provide and the required capacity in the next three years. In tactical level (or medium-range term), the decisions are to determine in a gross mode the required resources to perform the service in a medium planning horizon, e.g., a typical problem consists of determining the required workforce in six months. In operational level (or short-range term), the decisions determine how to carry out the specific tasks in the service operation in short planning horizon, e.g., the scheduling problem of an operator to deliver a package to the customer's site. The operation

\* Corresponding author Tel.: +57-3124697088  
E-mail: [erlopezs@udistrital.edu.co](mailto:erlopezs@udistrital.edu.co) (E. López-Santana)

could be on all scales, from within specific towns or cities, to regional, national and global services. The complexity of the problems increases from the strategic level to operational level because the required information also increases (and more detailed) and the response time is less. The scheduling problems belong to the operational level of the decision-making process and thus is difficult to solve.

The courier services have grown in coverage and in operative complexity, although, in many of them, the traditional expertise of their workers is still used as the main input to execute operation (Rodríguez-Vásquez et al., 2016). This feature allows the chance of human error to develop inconsistencies in the process, triggered, among other reasons, by employee turnover and the constant loss of the compounded knowledge and experience about the specific tasks involved in the process. In addition, when they are faced with scheduling services manually, even the most experienced planners can only consider a limited number of possibilities and need to invest a significant amount of time to obtain a feasible schedule that it is generally far from the optimum.

The courier service generally consists in distributing thousands of packages that are managed daily to a set of customers geographically distributed by a crew of workers (Rodríguez-Vásquez et al., 2016). There is a combinatorial optimization problem for finding the best set of routes for a workforce crew known as the *vehicle routing problem* (Toth & Vigo, 2002). In a broad sense, this problem finds the best set of routes to be performed by a set of vehicles (crews) in order to serve a set of geographically-spread customers subject to some operational constraints. Thus, the courier services could be modeled as vehicle routing problem (VRP).

In the courier services, there are real-life characteristics that are related to variants of the VRP models. The VRP with time windows (VRPTW) is the most common variant to include in the courier services. This constraint arises when the appointments can be arranged before delivery and have a specific date and time to serve (Toth & Vigo, 2002). The capacitated VRP (CVRP) involves the capacity constraint as the maximum load capacity of the vehicles, (Toth & Vigo, 2002). The distance-constrained VRP (DC-VRP) involves the time or distance constraint, in addition to the capacity constraint, each vehicle has a maximum traveling distance that can be reached, usually given in terms of distance or time (Farahani et al., 2011). The multi-period VRP (MP-VRP) considers a planning horizon composed of a set of periods in which customers have to be reached at least once. Another extension of the VRP is the due dates (VRPD) and this constraint takes into account the customer's due dates (Archetti et al., 2015).

This paper presents a scheduling and routing problem in courier services that involves the variants of VRP described above: time windows, multiples periods, capacity, due date and distance-constrained. To solve this problem we propose a three-phase approach. The first phase consists of a scheduling model to find the visit date for each customer over the planning horizon by considering the release date, the due date and travel times between the customers and depot. To solve this problem, we use an expert system. The second phase is a clustering model for each period in order to assign customers to the crew according to the travel times, maximum load capacity and the customer's time windows. We use two algorithms to solve the problem: a centroid-based heuristic and sweep heuristics. Finally, the third phase consists of a routing model in order to find the sequence to visit all customers taking into account the customer's time windows and the available time of the vehicles. An Ant Colony Optimization (ACO) metaheuristic was developed to solve this problem.

The remainder of this paper is organized as follows. Section 2 reviews the background and literature related to scheduling and routing problems in courier services. Section 3 states the problem and its notation. Sections 4, 5 and 6 describe the scheduling, clustering and routing phases, correspondingly, to solve the scheduling and routing problems in courier services. Section 7 provides some numerical experiments of our proposed approach in an example inspired by a real-world case. Finally, Section 10 concludes this work and provides possible research directions.

## **2. Background and literature review**

Scheduling and routing are very complex problems in service industries like courier companies. These problems involve decisions to allocate deliveries to a set of limited resources in a specific time under

several operational constraints as time windows, capacity, due dates, among others. The complexity increases when the decision makers are faced with hundreds or thousands of packets to be scheduled in a short amount time by considering several performance measures as an operational cost, customer's service level, profitability, etc. The vehicle routing problem (VRP) is closely related with scheduling and routing problems in courier services because it involves several variants and extensions of VRP, thus these problems are NP-hard (Toth & Vigo, 2002).

Different applications of courier services tightly coupled with VRP can also be found in the literature. Malmborg (2000) states a preliminary application of scheduling in courier services. The author studies the bank messenger problems as scheduling application and it determines the starting time and the sequence of stop locations. The problem is solved using a heuristic based on a simplified criterion check processing delays. Ghiani et al. (2009) describe an approach to solve the dynamic vehicle dispatching problem with pickups and deliveries. They propose a set of anticipatory algorithms to solve the problem. Their objective function consists of minimizing the expected inconveniences of the customers and their results show better solutions compared with other approaches. Chang and Yen (2012) propose routing and scheduling strategies for city couriers in Taiwan. They seek to reduce the operational costs and improve the service level using a multi-objective multiple traveling salesman problems formulation. Their objective function minimizes the total traveled length and the unbalanced workload, simultaneously. They considered hard time windows and proposed a multi-objective scatter search framework in order to find the Pareto-optimal solutions. Yan et al. (2013) study the problem of planning and adjustment of courier routes and schedules applied to urban regions. They use VRP models that include demand and stochastic traveling times. Their approach adjusts planned routes according to actual operations in real time. Lin et al. (2014) show a prototype of a decision support system to solve the offline and online routing problems arising in courier service. They formulated the problem as a dynamic vehicle routing problem (DVRP) using fuzzy time windows in order to represent the service level. To solve the problem, a hybrid neighborhood search algorithm was developed. Their results find an improvement of the courier service level without the further expense of a longer traveling distance or a larger number of couriers. Janssens et al. (2015) present a two-phase approach to solve a courier scheduling problem. The first phase consists of partitioning a distribution region into smaller zones that are assigned to a preferred vehicle. Then, in the second phase, they modeled the problem for each zone based on VRP as a multi-objective optimization problem and develop a heuristic to solve it.

The works described above have in common the application of VRP in the courier services. Table 1 summarizes the constraints that usually are taken into consideration as capacity and time windows, also one or two simultaneous constraints are used. In this paper, we consider other constraints such as distance, and multiples periods, therefore, our solution considers four real-life characteristics at the same time, implying a high level of complexity.

**Table 1**

Literature of application of VRP in courier services

	Capacity	Time windows	Distance	Pickup and delivery	Multi-period	Dynamic demand	Stochastic demand
(Malmborg, 2000)	√					√	
(Ghiani et al., 2009)	√			√		√	
(Chang & Yen, 2012)	√	√					
(Yan et al., 2013)	√						√
(Lin et al., 2014)	√	√				√	
(Janssens et al., 2015)	√						
This paper	√	√	√		√		

The VRP is one of the most researched combinatorial optimization problems in the literature and the trends towards VRP includes more real-life characteristics (Braekers et al., 2016). Several variants of the VRP includes features related to scheduling and routing courier services. Braekers et al. (2016) present a survey of the trends in the VRP literature. The authors found that CVRP (capacitated VRP) and VRPTW (VRP with Time Windows) are the features that have the most publications considered and

these are the most important variants used in the modeling of courier services. Another important variation of the VRP is the Distance-Constrained (DC-VRP) that involves a real-life feature as the maximum distance that a vehicle is available to drive or the labor period, however some studies are dedicated to this constraint (Almoustafa et al., 2013; Kek et al., 2008; Nagarajan & Ravi, 2012; Tili et al., 2014). On the other hand, the pickup and delivery constraint is generally combined with VRPTW (Fikar & Hirsch, 2015; Küçüköglu & Öztürk, 2015; Lin, 2011; Liu et al., 2013; López-Santana & Romero Carvajal, 2015). The MP-VRP (multi-period VRP) considers a planning horizon where the customers need to be visited several times with periodic or non-periodic frequency according to the service operation. Francis et al. (2006) study both cases, periodic and non-periodic frequency, and implement a service choice variable and present several heuristics for its solution. Rodriguez et al. (2015) present an iterative method to solve the PVRP through the construction of a unique visit schedule. Their approach allows the load deconsolidation, using non-regular days of visits and variable frequencies to make it a more robust and flexible tool for a real-world environment. Also, the due date is a constraint added to customers arising a variant known as VRPD (VRP with due date). Archetti et al. (2015) study this variant with multiples periods and solve it with a branch-and-cut algorithm. According to Braekers et al. (2016), it is observed that several applications have real-life characteristics individually or in some case with a limited number of other characteristics, however, the literature lacks many combinations of realistic features. In the case of courier services, multiple features must be considered simultaneously as time windows, multiples periods, capacity, due date and distance-constrained. In addition, it is possible to develop efficient solution methods to solve these problems.

On the other hand, there are a wide variety of solution methods to solve the VRP in the literature, being able to be classified in exact methods, heuristics, and metaheuristics (Eksioglu et al., 2009). Among the main exact methods used to solve the VRP are the Branch and Bound (B&B), Branch and Cut and Set-covering-based (Farahani et al., 2011). An example of a B&B can be found in Almoustafa et al. (2013), where B&B is used to solve a VRP with distance constraints. For the Branch and cut, an example of the application for the VRP with multiple periods can be found in Archetti et al. (2015). An example of the Set-covering-based is the article (Cacchiani et al., 2014).

About the heuristics, the most used are the two-phase heuristics as: the cluster-first route-second, the route-first cluster-second and the PEDAL algorithms (Toth & Vigo, 2002). In the first, the classic example is the Sweep Algorithm (Toth & Vigo, 2002), the Adaptive Large Neighborhood Search framework (ALNS) in which several types of VRP can be solved (Pillac et al., 2012). For the second family, among the route-first cluster-second methods, application of Greedy Algorithms can be found (Chu et al., 2006; Sprenger & Mönch, 2012). Other heuristics are the constructive heuristic, the Clarke and Wright's Saving Algorithm (Clarke & Wright, 1964) is the most famous heuristic. Another more recent example is presented in (Lin, 2011) using constructive heuristic as a step of the broader heuristic proposal.

The third group is the metaheuristics, which are frequently used to solve large combinatorial optimization problems. Some of the most commonly used are: Tabu search (TS), developed from heuristic from local search while additionally including a solution evaluation, local searching tactics, termination criteria and elements such as tabus in a list and the tabu length (Jia et al., 2013); Ant Colony Optimization (ACO), inspired in the feeding process of the ants and is a collective intelligence algorithm used to solve complex combinatorial optimization problems as shortest path problems (Ding et al., 2012); and Genetic Algorithm (GA), defined as adaptive search heuristic that operates over a population of solutions, based on the evolution principle that improves the solution using crossover and mutation process (Pereira & Tavares, 2009). Another metaheuristic used to solve VRPs using neighborhoods is the work of Rincon-Garcia et al. (2017) that uses Large Neighborhood Search approaches and Variable Neighborhood Search techniques to guide the search. Likewise, Galindres-Guanca et al. (2018) present a methodology of three stages to solve the multi-depot VRP. In the first one, the starting solutions are built up using constructive heuristics. The second stage consists of improving each starting solution with an iterated local search

multi-objective metaheuristic (ILSMO). In the last stage, a single front is found using concepts of dominance and taking as a base for the previous results.

In the field of Expert Systems (ES), which is a knowledge-based system and uses the reasoning methods that emulate the behavior and performance of human expert to solve problems. The ES is a specific field of artificial intelligence (Díez et al., 2001; Méndez-Giraldo et al., 2013). An ES is a computer program that reasons and uses a knowledge base to solve complex problems in a particular domain (Krishnamoorthy & Rajeev, 1996; Kusiak, 1990). Sahin et al. (2012) show a statistical analysis of hybrid ES approaches and their applications. The publications have an increasing trending as an indicator of the popularity of hybrid ES. Dios and Framinan (2016) present a review of case studies in manufacturing scheduling tools, while Wagner (2017) shows a case study using ES in different areas and state future trends. Both studies affirm that the ESs play a strategic role in the scheduling methods and techniques since they allow to involve the human expertise in the algorithms and to hybrid with other techniques. However, in service systems the application of ES is scarce, López-Santana and Méndez-Giraldo (2016) present a proposal of an ES for scheduling in service systems. They state that the application of knowledge-based systems and ES for scheduling in services systems are scarce and propose a structure to determine the service system and identify the tools to solve a specific scheduling problem.

When examining the literature, one should notice the lack of modeling approaches for scheduling and routing problems in courier services because of its complexity in the multiple real-life characteristics and the large-scale of the problems. Indeed, most of the papers deal with applications of VRP and its variants individually. The novelty of this work consists of exploring and combining multiple real-life characteristics as time windows, due dates, multiples periods, distance-constrained and capacity constraints in the VRP model applied to courier services. According to the literature trends, we propose a hybrid solution approach that integrates an expert system to scheduling deliveries, two heuristics to clustering customers and a metaheuristic based in ACO to solve a routing problem.

### 3. Problem statement

We consider a set of customers  $V_c = \{1, 2, \dots, N\}$  geographically distributed, in which each customer expects to receive a package. The set of workers (vehicles) who visit the customers are identically denoted by  $K = \{1, 2, \dots, m\}$  and belong to a central hub  $\{0\}$  that is considered to be the starting point and the end point of all workers. We assume that a worker is assigned to a vehicle, thus we use worker and vehicle, interchangeably. All customers must be visited once during the planning horizon. Then, we define the problem in a directed complete graph  $G = \{V, A\}$  where  $V = V_c \cup \{0\}$  is the set of vertices and  $A = \{(i, j) : i, j \in V, i \neq j\}$  is the set of arcs. Each arc  $(i, j)$  has a non-negative associated value  $t_{ij}$  that represents the traveling time from  $i$  to  $j$ . The objective is to establish the set of routes to visit all customers seeking to minimize the total routing distance for the vehicles. The assumptions and conditions of the problem are summarized as follow:

- The service time is the same for all customers and all vehicles.
- Each customer imposes a hard time-window in which the delivery must start. This means that the vehicle must arrive to start the service before the end of that time window, and it can arrive before the beginning of the time window, but the customer will not be serviced before this earliest time.
- The vehicles are homogeneous, i.e., they have the same capacity but they have different time windows for their labor period. It means that each vehicle has an availability limitation for the traveling time and service time.
- The vehicles start and finish in a central depot.
- The travel times are deterministic and fulfill the triangle inequality.

Table 2 introduces the notation used in the mathematical models of our approach. Fig. 1 shows our approach that for solving three models:

**Table 2**

Notation used for scheduling, clustering and routing models

---

<b>Sets:</b>	
$V_C$ :	set of customers
$V$ :	set of vertices
$A$ :	set of arcs
$P$ :	set of periods in the planning horizon
$K$ :	set of vehicles
$V_p$ :	set of vertices to visit in period $p$
$A_p$ :	set of arcs that connect the set of vertices to visit in a period $p$
$V_{pk}$ :	set of vertices to visit in period $p$ for vehicle $k$
$A_{pk}$ :	set of arcs that connect the set of vertices to visit in a period $p$ for vehicle $k$
<b>Parameters:</b>	
$N$ :	number of customers
$m$ :	number of vehicles
$Tv$ :	service time in a customer
$Q_k$ :	capacity of vehicle $k$
$d_j$ :	demand of customer $j$
$e_j$ :	lower bound of time window for customer $j$
$l_j$ :	upper bound of time window for customer $j$
$n_k$ :	lower bound for the available time of vehicle $k$
$p_k$ :	upper limit for the available time of vehicle $k$
$R_k$ :	maximum work time for vehicle $k$
$t_{ij}$ :	travel time for arc $(i, j) \in A$
$h_j$ :	release date of the visit with customer $j$
$G_j$ :	due date to visit customer $j$
$D$ :	average visit capacity of the vehicles per hour
$H$ :	Total available hours of all vehicles
<b>Binary Variables:</b>	
$z_{jp}$ :	is 1 if the customer $j$ is visited on the period $p$ and 0 in otherwise
$y_{jk}$ :	is 1 if the customer $j$ is visited by the vehicle $k$ and 0 in otherwise
$x_{ij}$ :	is 1 if the arc $(i, j)$ is used for the optimal solution and 0 in otherwise
$y_k$ :	is 1 if the vehicle $k$ is used and 0 in otherwise
<b>Integer Variables:</b>	
$N_p$ :	amount of customers to visit on period $p$
<b>Continuous Variables:</b>	
$S_j$ :	time in which the visit of customer $j$ starts
$u_j$ :	accumulated load delivered up to reaching customer $j$

---

1. Scheduling model: The objective function, input, output and solution method for this phase is the set of parameters shown in Fig. 1. This model defines the visit date of each vertex from the set  $V_C$  in the planning horizon of  $p$  periods, by considering the release date, due date and the travel times  $t_{ij}$  for all  $(i, j) \in A$ . To solve this model, we use an expert system based on the know-how of the courier service as the knowledge base. The inference engine works as a rule interpreter. The output consists of two sets: a set of customers allocated for each period and its respective sub set of arcs for each period.
2. Clustering model: This model takes as input the outputs of the scheduling model, the demand and times windows of the customers, and the capacity and labor schedule for the workers (see Fig. 1). The clustering model consists of grouping all vertices of the set  $V_p$  according to the traveling time  $t_{ij}$ , that the arc  $(i, j) \in A_p$ , and allocates it to a vehicle  $k$  by considering its labor schedule  $[n_k, p_k]$ , its maximum load capacity  $Q_k$  and the time windows of the customers  $[e_j, l_j]$ . The outputs are a set of customers allocated for each vehicle in each period and its associated sub set of arcs.

3. Routing model: The inputs consists of the outputs of clustering model, the travel times, customer's time windows and vehicle's time windows. The routing model is an individual optimization process to determine the order in which each vehicle will visit all customers assigned during the clustering process, taking into account the time windows of the customers and the available time of the vehicles. The output is a set of routes for each vehicle, with the scheduled detailed in the start time of service at each customer.

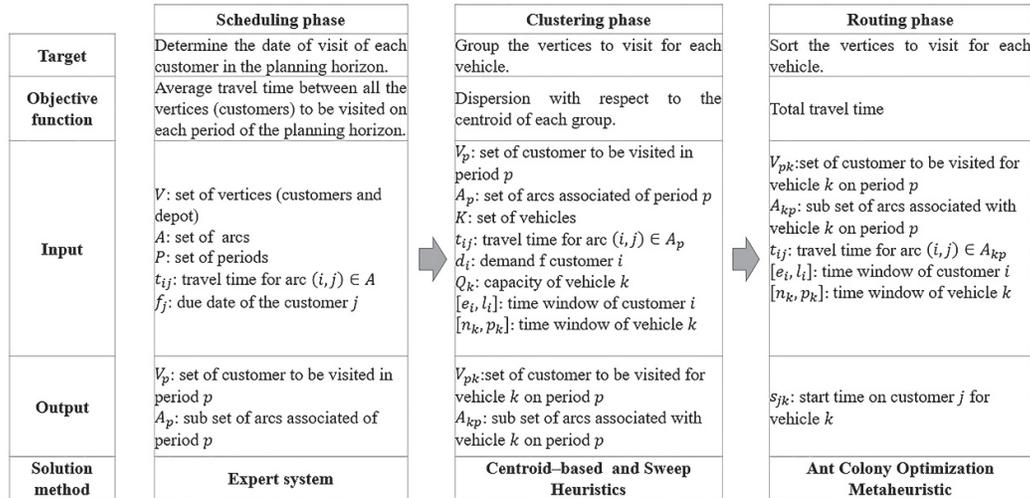


Fig. 1. Proposed solution approach to scheduling and routing in courier services

In the next three sections, we will present the components of our solution approach. First, we present the scheduling model to allocate the customers for each period; second, we develop the clustering model to determine the set of customers to each worker and finally, we present the routing model to find the sequence to visit the customers for each worker.

#### 4. Scheduling model

We have a set of customers to be scheduled in a period within the planning horizon. Since the number of customer is too large, then we need to solve a scheduling model by taking into account the appointments arranged with them, the due dates of the customers, the maximum load capacity of the vehicles, the number of vehicles available, the average visiting capacity of the vehicles by hour and the total available hours of the vehicle. The first phase consists of allocating customers to periods. The next sections explain the mathematical formulation of the problem and its solution procedure that consists of an expert system.

##### 4.1 Problem formulation

Given a set  $V_C$  as the set of customers that requires to be visited during the planning horizon  $P$ , we need to allocate each customer for a period to be visited. This problem uses a binary variable  $z_{jp}$  that take the value of 1 only if the customer  $j \in V_C$  is visited on period  $p \in P$ ; and 0, otherwise.

The scheduling model can be formulated mathematically as follows:

$$\min \frac{\sum_{p \in H} \sum_{(i,j) \in A} t_{ij} z_{jp}}{N} \tag{1}$$

subject to:

$$z_{jp} = f(h_j, G_j, t_{ij}, Q, m, H, D) \quad j \in V, p \in P \tag{2}$$

$$\sum_{p \in P} z_{jp} = 1 \quad p \in P \tag{3}$$

$$\sum_{j \in V} z_{jp} \leq HD \cdot D \quad p \in P \quad (4)$$

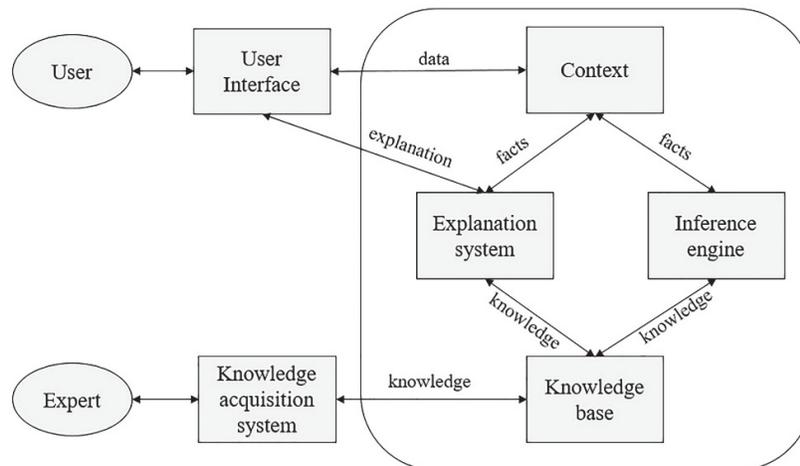
$$z_{jp} \in \{0, 1\} \quad j \in V, p \in P \quad (5)$$

The objective function (1) seeks to minimize the average distance between all customers to visit each period in the planning horizon, allowing to maximize the concentration in the visiting groups. Constraint (2) represents the relation between assigning a specific customer to a period depending on the appointments arranged with them, the due dates of the customers, the maximum load capacity of the vehicles, the amount of total vehicles available, the average visiting capacity of the vehicles by hour and the total available hours of the vehicles. In the scheduling model, this relation is represented by a set of rules that belongs to an expert system which also includes basic rules to comply with the constraints, e.g., the date of the appointments arranged with customers, and priority rules as ranking customers that are closer to the due date, customers who do not have an appointment and are closer than one who does so they can be visited in the same period. Constraint (3) assures that each customer will have a determined visiting date in the planning horizon. Constraint (4) controls the maximum amount of deliveries that can be programmed to be visited according to the total load capacity. And finally, Constraint (5) defines the binary variable used.

#### 4.2 Solution procedure

Regarding the courier service, and referring to the definition in this paper, scheduling is performed by the workers using mainly the know-how and expertise of the business to define the period in the planning horizon in which the visit of each customer will be executed. Our plan consists of providing a model which allows complying with the efficacy of the scheduling process and to prevent the chance of human error. An expert system (ES) is a well-designed system which replicates the cognitive process that experts use to solve particular problems (Turban, 1989). In addition, from the designed rules that came from workers experience, the ES will use as a performance function, the objective function established in Eq. (1), that consists in the minimization of the average distance between all customers to visit each period of the planning horizon.

The ES is specialized in a specific field and aims to solve problems through reasoning methods that emulates the performance of a human expert (Díez et al., 2001; Méndez-Giraldo et al., 2013). Fig. 2 shows the architecture of a traditional ES.



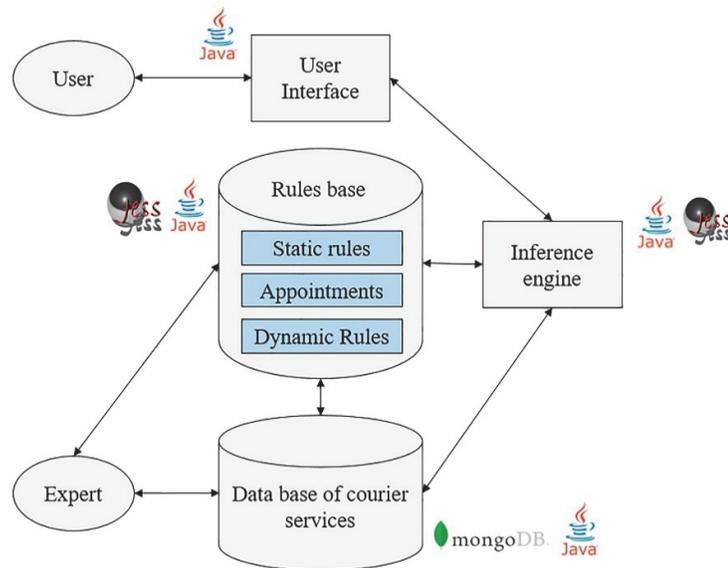
**Fig. 2.** Architecture of a traditional ES (source: modified of Krishnamoorthy and Rajeev (1996))

The ES operates according to López-Santana and Méndez-Giraldo (2016). When the ES starts the process of inference, it needs a context or working memory, which represents the set of established facts. The explanation system simulates the process of the answer of an expert to the questions: *How* is a decision arrived at? and *Why* do we need a data? Since the knowledge base has to be continuously updated and/or appended depending on the growth of knowledge in the domain, an interface between expert and ES is

necessary. The knowledge acquisition system executes this interface, and is not an on-line component, it can be implemented in many ways. In addition, another interface is necessary between the user and the ES. The user interface allows the user interacts with the ES giving data, defining facts and monitoring the status of the problem-solving.

The knowledge base is the component where all knowledge provided by experts is stored in an orderly and structured way under a set of relationships, such as rules or probability distributions, facts and heuristics that represent the thinking of the expert. The major task of ES rules the basis of its operation, but we can also use representation schemes such as semantic networks, frames, among others. The knowledge base is independent of the mechanisms of inference and search methods.

The inference engine performs two main tasks (López-Santana & Méndez-Giraldo, 2016): in the first phase, it examines the facts and rules, and if possible, add new facts; in the second one, it chooses the order in which inferences are made. Notably, the findings obtained by the inference engine, are made based on deterministic or stochastic information data and can be simple or compound.



**Fig. 3.** Architecture of knowledge base and inference engine

The main components that make up our proposed ES are the base of knowledge and the inference engine. Fig. 3 presents the structure of our ES. The *knowledge base* is represented by the data base of courier service through a process of documentary review of internal procedures, service agreement and also through interviews with employees who carry out scheduling activities, which may be considered as pseudo-expert. They will be formulated the rules of the type if (condition) then (action), which together create rules base. Both, data base and rules base, are the knowledge base of our ES and were built in mongoDB® and Java®.

The *inference engine*: this is a rule interpreter who decides when to apply the rules. This was built in Jess (Java expert system shell) and Java®. From this definition, the rules of the knowledge base will be clustered into three modules, to be applied in an orderly and sequential way as follows:

*Module of Static rules*: The first module contains static rules since they are applied to every customer's package to visit of the available stock. These rules are based on the internal procedures and service agreements with the customers. The result is a feature modification named route status, this feature can assume three values: *pending*, *dismissed* and *scheduled*. If some rule is fulfilled, the feature is changed to "*dismissed*". If none of the module rules is fulfilled, the feature is remained with the value "*pending*".

*Module of Appointments*: The second module contains only one rule, however, it is the major and application basis of the next module. The rule consists of evaluating if the customer has a booked

appointment within the planning horizon if so, it changes the route to be “*scheduled*” and the same route date as the appointment date.

*Module of Dynamic rules:* The third module contains a set of rules that are activated only if they are required. These rules were designed to take advantage of ES, which consists of intensive searches, in other words, instead of re-executing computational efforts by repeatedly applying different rules to the same ones, efforts are directed towards searching on a large basis of facts, those that activate the rules. These rules are built based on the experience and know-how of the pseudo-experts who program the routes.

The set of rules that make up the base of knowledge and the inference engine that was developed for the case study are listed in the Appendix A. The user interface and all components are integrated into Java®. The outputs of scheduling model are the sets of customers that are scheduled to visit for each period. The next section describes the clustering model in order to build a determined number of clusters for each period associate each one for a worker.

## 5. Clustering model

With the results of scheduling model, we have for each period in the planning horizon, a set of customers is scheduled to be visited, however, the scale of these sets are too large, thus we use a strategy for grouping in sets for each vehicle using a clustering model. To solve this model we propose two algorithms: a centroid-based and sweep algorithms. The next sections describe the mathematical formulation of the clustering model and its solution procedures.

### 5.1 Problem formulation

Given the set of customers  $V_p$  scheduled to be visited in each period  $p$ , we use a binary variable  $y_{jk}$  that takes the value of 1 only if the vehicle  $k \in K$  visits the customer  $j \in V_p$ ; and 0, otherwise. For each period  $p$ , the clustering model can then be stated as follows,

$$\min \frac{\sum_{k \in K} \sum_{i \in V_p} t_{ij} y_{jk}}{N_p} \quad (6)$$

subject to:

$$\sum_{j \in V_p} d_j y_{jk} \leq Q_k \quad k \in K \quad (7)$$

$$\sum_{k \in K} y_{jk} = 1 \quad j \in V_p \quad (8)$$

$$e_j - n_k y_{jk} \geq 0 \quad j \in V_p, k \in K \quad (9)$$

$$y_{jk} \in \{0, 1\} \quad j \in V_p, k \in K \quad (10)$$

The objective function (6) seeks to minimize the average distance among all the customers to be visited by the vehicle  $k$ . Constraints (7) limit the maximum load capacity of each vehicle. Constraints (8) state that a customer can be allocated only for one vehicle. Constraints (9) ensure that the customers with time windows are allocated to vehicles that start their shift before. Constraints (10) define the binary variable used.

### 5.2 Solution procedure

There are several methods that allow clustering customers in groups according to distances, costs, etc. (Patiño Chirva et al., 2016). We selected two procedures: Centroid-Based and Sweep algorithms. Given  $d(v_j)$  as the demand of customer  $j \in V_p$ ,  $v_j$  denotes the vertex of the customer  $j$ ,  $[e, l]_{v_j}$  is the time window of customer  $j$  and  $[e, l]_{l_i}$  is the schedule labor period of cluster  $l_i$ , i.e., the time window when a

vehicle is available to deliver the packages. We assume that a cluster corresponds to a vehicle.

The Centroid-Based algorithm starts from the geometry of the geometric centers, around which the cluster environment is generated. Algorithm 1 shows the pseudocode of the centroid-based algorithm for clustering the set of customers to be visited for each period. This method is divided into two steps according to Shin and Han (2012) and we modify to adapt the time windows for customers and vehicles.

---

**Algorithm 1. Centroid-Based algorithm**

---

```

1: First step // Clustering construction
2:  $i = 0$  and  $Q =$ vehicle capacity
3: while ( $V_p \neq \emptyset$ ) // all customers are allocated
4:    $v_j$ : the farthest-to-origin customer among all un-clustered customers
5:   if ( $[e, l]_{v_j} \in [e, l]_{l_i}$ )
6:     Generate cluster  $l_i$  with customer  $v_j$ 
7:      $Q_{l_i} = Q$  // State cluster capacity
8:     while ( $d(v_j) \leq Q_{l_i}$ )
9:       Add  $v_j$  to  $l_i$ 
10:       $Q_{l_i} = Q_{l_i} - d(v_j)$ 
11:      Compute  $CG(l_i)$ 
12:       $V_p = V_p - j$  and  $V_p' = V_p$ 
13:      for all  $k \in V_p'$ 
14:         $v_k$ : node closest to  $CG(l_i)$ 
15:        if ( $[e, l]_{v_k} \in [e, l]_{l_i}$ )
16:           $v_j = v_k$ 
17:          Break
18:        Else
19:           $V_p' = V_p' - k$ 
20:        end if
21:      end for
22:    end while
23:     $i = i + 1$ 
24:  end if
25: end while
26: End first step
27: Second step // Cluster adjustment
28: State  $L = \{l_0, l_1, \dots, l_m\}$ , which is the set of cluster generated in the first step
29: for  $i = 0$  to  $m$ 
30:   for all  $v_k \in l_i$ 
31:     for all  $l_j \in L$ .
32:       if ( $[e, l]_{v_k} \in [e, l]_{l_j}$  and  $i \neq j$  and  $v_k$  is closer to  $CG(l_j)$  than to  $CG(l_i)$  and  $d(v_k) \leq Q_{l_j}$ )
33:         Move customer  $v_k$  from  $l_i$  to  $l_j$ 
34:         Compute  $CG(l_j)$  and  $CG(l_i)$ 
35:          $Q_{l_i} = Q_{l_i} + d(v_k)$ 
36:          $Q_{l_j} = Q_{l_j} - d(v_k)$ 
37:       end if
38:     end for
39:   end for
40:    $i = i + 1$ 
41: end for
42: End Second step

```

---

In the first step is the clustering construction, the algorithm selects the node farthest from the source, within nodes that have not been previously allocated and a cluster is generated; then the geometric center calculated with Eq. (11) must be computed between these nodes, where  $w_i^x$  and  $w_i^y$  are coordinates in  $x$  and  $y$ , respectively, and the nodes belonging to the cluster.

$$CG(l_i) = \left( \sum_{i=0}^m w_i^x / m, \sum_{i=0}^m w_i^y / m \right) \quad (11)$$

To add nodes to the first cluster ( $l_0$ ), the cluster construction algorithm finds  $v_j$  among un-clustered nodes, which is located closest from  $CG(l_0)$ , and includes  $v_j$  to  $l_0$  only if the demand of  $v_j$  does not exceed the available capacity of  $l_0$  and its time window belongs to the labor schedule of  $l_0$ . If  $v_j$  is added to cluster  $l_0$ , the capacity of the cluster is reduced by the demand of  $v_j$  and  $CG(l_0)$  is recalculated. The same processes above are conducted until the available capacity of  $l_0$  becomes smaller than the demand of the closest node from  $GC(l_0)$ . When the demand of  $v_j$  exceeds the available capacity of  $l_0$ , the algorithm stops to expand  $l_0$ , and finds the farthest node among un-clustered nodes again in order to generate another cluster,  $l_1$ . These processes are repeated until no unvisited node exists, i.e., when  $V_p \neq \emptyset$ . In summary, the nodes closest to the geometric center are added taking into account the defined capability of each cluster, which corresponds to the capacity of a vehicle, the time windows associated with the customers and the available schedules of the vehicles. The time complexity of this step is  $O(n^2)$ .

In the second step, the clusters generated in step one are adjusted. Cluster adjustment means that if customer  $v_k$ , which belongs to cluster  $l_i$ , is closer to  $CG(l_j)$  than  $CG(l_i)$ , the demand of  $v_k$  does not exceed the available capacity of  $l_j$ , and the time window of  $v_k$  belongs of the available labor time of  $l_j$  then move  $v_k$  from  $l_i$  to cluster  $l_j$ . If a customer moves from  $l_i$  to  $l_j$ ,  $CG(l_i)$  and  $CG(l_j)$  are also recalculated. The time complexity of this step is also  $O(n^2)$ .

---

**Algorithm 2. Sweep algorithm**


---

```

1: First step // Initialization
2: Set  $i = 0$  and  $Q =$  vehicle capacity
3: Compute polar coordinates of customers  $(\rho_i, \theta_i)$ 
4: Sort cluster  $L_k$  by schedule
5: Order customers according value  $\theta_i$ 
6: if  $(\theta_i = \theta_{i+1})$ 
7:     Sort customers in ascending order according to the value  $\rho_i$ 
8: end if
9: End First Step
10: Second step //Cluster generation
11: for all  $k \in l_k$  //from first to last cluster in  $l_k$ 
12:     Select an initial customer  $v_j$ 
13:      $k = 0$  and  $l_k = \{v_j\}$ 
14:      $Q_{l_k} = Q_{l_k} - d(v_j)$ 
15:      $N_p = N_p - j$ 
16:     while  $(N_p \neq \emptyset)$ 
17:         Select the next customer  $v_{j+1}$ 
18:         if  $([e, l]_{v_{j+1}} \in [e, l]_{l_k} \text{ and } d(v_{j+1}) \leq Q_{l_k})$ 
19:              $l_k = l_k \cup \{v_{j+1}\}$ 
20:              $Q_{l_k} = Q_{l_k} - d(v_j)$ 
21:              $N_p = N_p - j$ 
22:         end if
23:     end while
24:      $k = k + 1$ 
25: end for
26: End Second Step

```

---

The Sweep algorithm is based on polar coordinates (Gillett & Miller, 1974; Toth & Vigo, 2002). Algorithm 2 shows pseudocode for clustering the customers based in sweep algorithm. The first step is the initialization process, which consists of computing the polar coordinates and sort them in ascending order. Moreover, the clusters  $L = \{l_0, l_1, \dots, l_K\}$  are sorted by schedule time. The second step is the cluster construction. It starts from an origin point a straight line which is rotating and creating a zone in which the customers must be assigned. The area addressed in the sweeping process constitutes the cluster, as long as it complies with the capacity constraint indicated for each one of them. The algorithm selects  $v_j$  among un-clustered nodes includes  $v_j$  to  $l_0$  only if the demand of  $v_j$  does not exceed the available capacity of  $l_0$  and its time window belongs to the labor schedule of  $l_0$ . If  $v_j$  is added to cluster  $l_0$ , the capacity of it is reduced by the demand of  $v_j$ . The same processes above are conducted until the available capacity of  $l_0$  becomes smaller than the demand of the closest node from  $GC(l_0)$ . Then we select the next cluster,  $l_1$  and the process is repeated until no unvisited node exists, i.e.  $V_p \neq \emptyset$ . Like centroid-based algorithm, the sweeping heuristics were modified to take into account to the time windows associated with the customers and the available schedules of the vehicles. The time complexity of this algorithm is  $O(n^2)$ .

We have two algorithms to compute the clusters, then we will compare the average distances to select the clusters with the less average of distance between all customers of set  $V_p$ . The outputs of this model are the sets of clusters that represent the workers or vehicles for each period that need to be visited. The next section describes the routing model to determine the sequence in which the customers are visited.

## 6. Routing model

With the results of the clustering phase, we have a set of customers to be allocated in paths in order to be served for the crew of vehicles. This situation arises with a special routing model that consists in a traveling salesman problem with time windows (TSPTW) because each vehicle is solved separately. The next sections explain the mathematical formulation of the routing model and the solution procedure that is based on the Ant Colony Optimization (ACO) metaheuristic.

### 6.1 Problem formulation

Given a set of customer  $V_{kp}$  allocated to a period  $p$  and a vehicle  $k$  for the clustering model, the vehicle must serve the customers, thus we are faced to solve a routing model. The problem is defined on the auxiliary directed graph  $G' = (V'_{kp}, A'_{kp})$ , with  $V'_{kp} = V_{kp} \cup \{0, n+1\}$  and  $A'_{kp} = \{(0, i): i \in V_{kp}\} \cup \{(i, n+1): i \in V_{kp}\} \cup \{(i, j): i, j \in V_{kp}, i \neq j\}$ , where vertices 0 and  $n+1$  denote the depot. The node  $n+1$  is a copy of the depot 0 and is introduced for the sake of clarity in the mathematical formulation presented hereafter. We assume a zero service time at the depot, i.e.  $T_{v_0} = T_{v_{n+1}} = 0$ .

This problem consists of designing a set of routes for each vehicle such that:

- Each customer is visited exactly once,
- The time window  $[e_j, l_j]$  is met, i.e., the vehicle must arrive to start service before time  $l_j$  and no later than  $e_j$ , but the customer will not be serviced before the beginning of the time window, and
- The time window  $[n_k, p_k]$  is met, i.e., the vehicle has a time window that represents its labor period.

The problem is formulated as a TSPTW. This problem uses binary variables  $x_{ij}$  that takes the value of 1 only if the vehicle traverses the arc  $(i, j) \in A_{kp}$ , where  $i \neq j$ ; and 0, otherwise. The variable  $s_j$  defines the start time of a service  $j \in V_{kp}$ . Finally, the variables  $u_j$  represents the accumulated load in customer  $j \in V_{kp}$ . For each vehicle  $k$  and each period  $p$  in the planning horizon, the mathematical model is formulated as follows:

$$\min \sum_{(i,j) \in A_{kp}} t_{ij} x_{ij} \quad (12)$$

subject to:

$$\sum_{j \in V_{kp} \setminus j \neq i} x_{ij} = 1 \quad i \in V_{kp} \quad (13)$$

$$\sum_{i \in V_{kp} \setminus i \neq j} x_{ij} - \sum_{i \in V_{kp} \setminus i \neq j} x_{ji} = 0 \quad j \in V_{kp} \quad (14)$$

$$(u_i + d_j - u_j) \leq M(1 - x_{ij}) \quad (i, j) \in A_{kp} \quad (15)$$

$$(u_i + d_j - u_j) \geq -M(1 - x_{ij}) \quad (i, j) \in A_{kp} \quad (16)$$

$$u_i \leq Q \quad i \in V_{kp} \quad (17)$$

$$s_i + t_{ij} + Tv - M(1 - x_{ij}) \leq s_j \quad (i, j) \in A_{kp} \quad (18)$$

$$e_j \sum_{i \in V_{kp}} x_{ji} \leq s_j \leq l_j \sum_{i \in V_{kp}} x_{ji} \quad j \in V_{kp} \quad (19)$$

$$n_k \leq s_j \leq p_k \quad j \in \{0, n + 1\} \quad (20)$$

$$x_{ij} \in \{0, 1\} \quad (i, j) \in A_{kp} \quad (21)$$

$$u_j \geq 0 \quad j \in V_{kp} \quad (22)$$

The objective function (11) aims to minimize the total travel time related to the visits performed for each vehicle. Constraints (12) ensure a visit each customer only once. Constraints (13) allow keeping fluency between trips in a route, i.e., after visiting one customer, a vehicle must immediately start another trip. Constraints (14) and (15) allocate values to the accumulated load variables and prevent sub-tours. Constraints (16) state the limited capacity of the vehicle. Constraints (17) establish the relationship between the vehicle's departure time from a customer and its immediate successor. Constraints (18) and (19) enforce the time windows at the customers and the vehicles. Constraints (20) impose binary conditions on the flow variables. Finally, constraints (21) defines the accumulated load variables as non-negative.

## 6.2 Solution procedure

With the previous procedure we allocate a set of customers for each period in the planning horizon, as a result we can simplify the problem as a TSPTW, capacity constraints, and schedule capacity. To solve this problem an ACO metaheuristic is proposed since TSPTW is considered an NP-complete, i.e., belong to the combinatorial optimization problems that are considered difficult to solve. However, according to Glover and Kochenberger (2003), ACO has had several successful implementations in a wide variety of combinatorial optimization problems. In these problems, generally, ACO algorithms are linked with additional features, such as local optimizers against specific problems, that take ant solutions for local optimum (Glover & Kochenberger, 2003). Some recent examples of successful applications of the ACO metaheuristic for the TSPTW can be found in the literature (e.g. Kara & Derya, 2015; López-Ibáñez & Blum, 2010). In addition, if we consider that ACO metaheuristic has also been used for the VRPTW, more applications can be found in the literature (e.g. Cheng & Mao, 2007; Ding et al., 2012; Pureza et al., 2012; Yu et al., 2011).

Cheng and Mao (2007) propose an ACO to solve the TSPTW. The authors developed a modified ant algorithm called ACS-TSPTW (Ant Colony System-TSPTW) based on the ACO technique to solve the TSPTW. Two local heuristics are integrated in ACS-TSPTW algorithm to manage the time windows limitations. Table 3 summarizes the main notation used for the proposed ACO.

**Table 3**

Notation used for proposed ACO to solve the TSPTW

$\tau_{ij}(t)$ : pheromone from node $i$ to $j$ during the iteration $t$
$\Delta\tau_{ij}(t)$ : pheromone update from node $i$ to $j$ during iteration $t$
$A$ : set of arcs $(i, j)$
$\mathcal{N}_i$ : paths that are missing from the current ant
$g_{ij}$ : heuristic that increases the importance of nodes near their closing time
$h_{ij}$ : heuristic that increases the importance of nodes that do not require timeout
$\mathcal{T}^*$ : best current path
$\mathcal{L}^*$ : travel time of the best current road

According to Cheng and Mao (2007), the ACO algorithm defines the amount of pheromone deposited for each path  $(i, j), \forall (i, j) \in A$  with global and local update. The local update is computed as follows:

$$\tau_{ij}(t) = (1 - \varpi)\tau_{ij}(t-1) + \varpi\Delta\tau_{ij}(t), \quad (23)$$

where  $\tau_{ij}(t)$  is the pheromone quantity on the way  $i, j$  for the time  $t$ ,  $0 < \varpi < 1$  is the local evaporation rate and  $\Delta\tau_{ij}(t)$  is the pheromone increase on the way  $i, j$  given by:

$$\Delta\tau_{ij}(t) = \begin{cases} \mathcal{L}^*, & \text{if } (i, j) \in \mathcal{T}^* \\ 0, & \text{otherwise} \end{cases}, \quad (24)$$

where  $\mathcal{L}^*$  is the distance traveled by the best path  $\mathcal{T}^*$  and, the global update is given by:

$$\tau_{ij}(t) = (1 - \theta)\tau_{ij}(t-1) + \theta \Delta\tau_{ij}(t), \quad (25)$$

where  $0 < \theta < 1$  is the global evaporation rate. The transition node rule consists in choice to move to the next node  $j$  is being taken with the Eq. (26).

$$j = \begin{cases} \arg \max_{z \in \mathcal{N}_i} \tau_{ij}(t)(g_{ij})^\beta (h_{ij})^\gamma, & \text{if } q \leq q_0 \\ S, & \text{otherwise} \end{cases}, \quad (26)$$

Two local heuristics are defined by Eq. (28) and Eq. (29). The parameters  $(\beta, \gamma)$  are user defined that determine the importance of  $g_{ij}$  and  $h_{ij}$  and  $S$  is the transition probability rule defined in Eq. (27).

$$P_{ij} = \begin{cases} \frac{\tau_{ij}(t)(g_{ij})^\beta (h_{ij})^\gamma}{\sum_{z \in \mathcal{N}_i} \tau_{iz}(t)(g_{iz})^\beta (h_{iz})^\gamma}, & \text{if } j \in \mathcal{N}_i \\ 0, & \text{otherwise} \end{cases}, \quad (27)$$

where  $\mathcal{N}_i$  is the set of nodes that the ant has not visited in this tour.

The local heuristic  $g_{ij}$  increases the importance of the nodes near their closing time and is defined by:

$$g_{ij} = \begin{cases} \frac{1}{1 + \exp(\delta(G_{ij} - \mu))}, & \text{if } G_{ij} \geq 0 \\ 0, & \text{otherwise} \end{cases}, \quad (28)$$

where  $G_{ij}$  is the remaining time to the node  $j$  to close and is defined as  $G_{ij} = b_j - t_j$  where  $b_j$  is the time to the node  $j$  close and  $t_j$  is the current time,  $\delta$  controls the probability curve and  $\mu$  is defined as the average of  $G_{ij} \geq 0$ . The intuition behind this heuristic is that the ant should visit those nodes whose arrival times are closer to their upper time-window constraints before those nodes with later upper time-window constraints in order to avoid the risk of lateness (Cheng & Mao, 2007). The local heuristic  $h_{ij}$  increases the importance of nodes near to its input time and is defined by:

$$h_{ij} = \begin{cases} \frac{1}{1 + \exp(\lambda(H_{ij} - v))}, & \text{if } H_{ij} \geq 0 \\ 1, & \text{otherwise} \end{cases}, \quad (29)$$

where  $H_{ij}$  is the remaining time to the node  $j$  to open and is defined as  $H_{ij} = a_j - t_j$  where  $a_j$  is the time to the node  $j$  to open and  $t_j$  is the current time,  $\lambda$  controls the probability curve and  $v$  is defined as the average of  $H_{ij} \geq 0$ . Note that  $h_{ij} = 1$ , when  $H_{ij}$  is negative, i.e., the open nodes are given the highest priority.

Algorithm 3 introduces the basic processing steps for the proposed ACO. The first step consists of starting the variables with initial values. The second step builds a path according to local heuristics  $g_{ij}$  and  $h_{ij}$ . The third step saves the best path found so far and his travel time, and updates the pheromone. The fourth one evaluates the stopping criterion, if it is met the algorithm stops and reports the solution, else the second step is run again and the iterative process continues until the stopping criterion is reached.

---

**Algorithm 3. Pseudocode ACO-TSPTW**

---

```

1: First step // Initialization
2: Set  $t = 0$  or  $t =$  route start time.
3: for all  $(i, j) \in A \rightarrow$  Set initial values  $\tau_{ij}(t)$ 
4: Put  $m$  ants in the tank or at a random node.
5: for all  $(i, j) \in A \rightarrow$  Set  $\Delta\tau_{ij}(t) = 0$ 
6: End first step
7: Second Step //Path construction
8: for  $i = 0$  to  $n$ : //  $n$  is the number of nodes
9:     for  $k = 1$  to  $m$ :
10:        if  $\mathcal{N}_i \neq \emptyset$ 
11:           //Local heuristics
12:           for all  $j \in \mathcal{N}_i \rightarrow$  Compute  $g_{ij}$  and  $h_{ij}$  // equations (28) and (29)
13:           if  $g_{ij} = 0 \forall j \in \mathcal{N}_i$ 
14:               $k = k + 1$  // no more feasible paths
15:           end if
16:           //Move:
17:            $j \rightarrow$ equations (26) and (27)
18:           Delete  $j$  from  $\mathcal{N}_i$ 
19:        end if
20:     end for
21: end-for
22: End second step
23: Third step
24: Save  $\mathcal{T}^*$  and  $\mathcal{L}^*$  // the best path found so far and his travel time, respectively
25: for all  $i \in \{1..m\} \mathcal{N}_i = n$  //for each ant
26: for all  $(i, j) \in A$ ://Local update
27:      $\tau_{ij}(t) \rightarrow$ equation (23) //update the pheromone
28: end for
29: for all  $(i, j) \in \mathcal{T}^*$ :
30:      $\tau_{ij}(t) \rightarrow$ equation (25) //update the pheromone
31: end for
32: End third step
33: Fourth step // Stopping criterion
34: if the stopping criterion is met
35:     Exit
36: Else
37:     Go to Second step
38: end if
39: End fourth step

```

---

For the initial amount of pheromone on each arc, Cheng and Mao (2007) suggest to determine it based on an approximate estimate of the tour length and, the stopping criterion is set to a maximum acceptable number of iterations to run the algorithm.

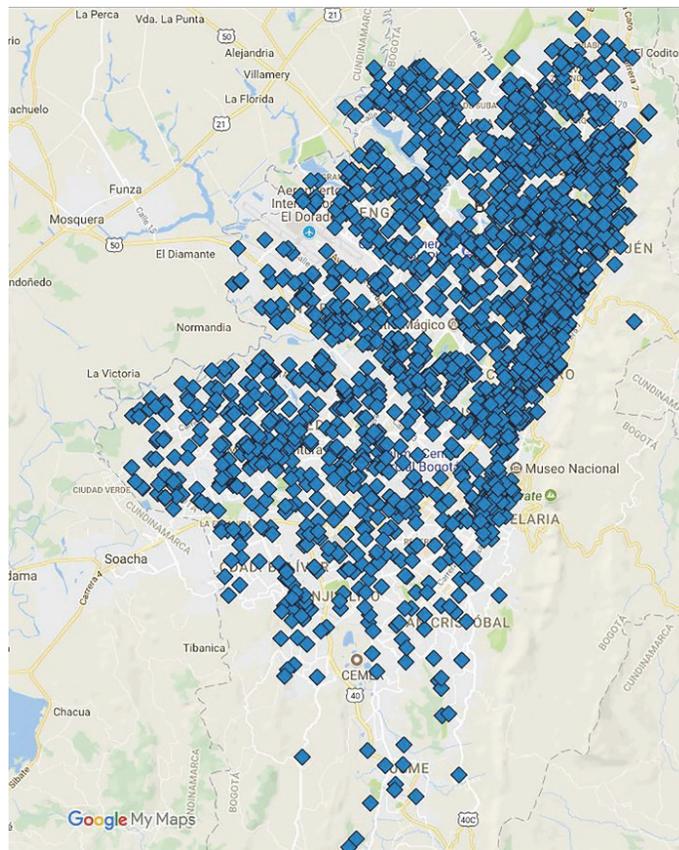
## 7. Results

In this section, we illustrate the results of our proposed method for scheduling and routing in courier services obtained in a real world case. All tests in this work were run using Java 8 on a Windows 8 64-bit machine, with an Intel i5 3337 processor ( $2 \times 1.8$  GHz) and 6 GB of RAM.

### 7.1 Case study

We illustrate the performance of the proposed method with an example based on a real-world scenario, where a courier company in Bogotá has a crew of more than two hundred motorcycle operators with delivery package in the city. The company has three operating centers, from which approximately 2.100 packages per day are delivered. That is 700 packages on average for each center, with 20 to 32 motorcycle operators, these can carry from 10 to 60 packages per day according to a schedule. We assume a planning horizon of three days.

The packages that must be delivered to the customer have a set of attributes, which determine the selection according to priorities in order to be scheduled. In the real case study, the packages have more than 60 attributes, however, many of them are informative and do not affect the rules of our ES. Table 4 lists the set of attributes considered in the case study. For the sample, a database with the packages that were in its system was requested by the courier company on September 20, 2016. In the sample received there were 17.492 packages. Fig. 4 shows the geographical dispersion of the packets on the map of Bogota. We assume a capacity of 2.700 visits per day and 75 vehicles.



**Fig. 4.** Geographical dispersion of sample of case study

**Table 4**

List of rules attributes of the case study



Table 5 shows the capacity  $Q_k$  as the number of visits per day and the time windows  $[n_k, p_k]$  for each vehicle  $k$ .

**Table 5**

Capacity and time window of vehicles in case study

$k$	$Q_k$	$n_k$	$p_k$	$k$	$Q_k$	$n_k$	$p_k$	$k$	$Q_k$	$n_k$	$p_k$
1	60	8:00:00	18:00:00	26	30	10:30:00	16:00:00	51	38	8:00:00	13:30:00
2	40	8:00:00	15:00:00	27	25	8:00:00	12:00:00	52	38	8:00:00	14:00:00
3	25	9:00:00	14:00:00	28	35	10:00:00	17:00:00	53	38	8:00:00	14:00:00
4	20	9:30:00	14:00:00	29	35	10:30:00	17:00:00	54	38	8:00:00	14:00:00
5	30	10:00:00	16:00:00	30	25	8:00:00	12:00:00	55	41	8:00:00	14:30:00
6	30	10:00:00	16:00:00	31	25	8:00:00	12:00:00	56	41	8:00:00	14:00:00
7	30	8:00:00	14:00:00	32	35	10:00:00	17:00:00	57	41	8:00:00	14:30:00
8	30	8:00:00	14:00:00	33	30	10:00:00	16:00:00	58	41	8:00:00	14:30:00
9	30	10:00:00	16:00:00	34	35	10:00:00	17:00:00	59	44	8:00:00	14:30:00
10	30	10:00:00	16:00:00	35	25	8:00:00	12:00:00	60	44	8:00:00	15:30:00
11	30	11:30:00	16:00:00	36	25	8:00:00	12:00:00	61	50	8:00:00	16:00:00
12	40	8:00:00	14:30:00	37	35	10:30:00	17:00:00	62	60	8:00:00	18:00:00
13	30	10:30:00	16:00:00	38	35	10:00:00	17:00:00	63	60	8:00:00	18:00:00
14	40	8:00:00	15:00:00	39	35	10:30:00	17:00:00	64	60	8:00:00	18:00:00
15	30	10:00:00	16:00:00	40	25	9:30:00	13:30:00	65	60	8:00:00	17:00:00
16	30	10:00:00	16:00:00	41	25	8:30:00	12:30:00	66	60	8:00:00	17:00:00
17	30	10:00:00	16:00:00	42	25	9:30:00	13:30:00	67	60	8:00:00	17:00:00
18	30	10:00:00	16:00:00	43	29	10:00:00	14:30:00	68	60	8:00:00	18:00:00
19	30	10:30:00	16:00:00	44	32	8:00:00	12:00:00	69	60	8:00:00	18:00:00
20	30	10:30:00	16:00:00	45	32	8:00:00	13:00:00	70	60	8:00:00	18:00:00
21	30	10:30:00	16:00:00	46	32	8:00:00	13:00:00	71	60	8:00:00	18:00:00
22	30	10:30:00	16:00:00	47	32	9:30:00	14:30:00	72	60	8:00:00	18:00:00
23	30	10:00:00	16:00:00	48	32	8:00:00	13:00:00	73	60	8:00:00	17:30:00
24	30	10:30:00	16:00:00	49	35	8:00:00	13:30:00	74	60	8:00:00	17:30:00
25	30	10:30:00	16:00:00	50	35	9:30:00	15:00:00	75	60	8:00:00	17:30:00

## 7.2 Results of case study

In this section, we report some of the results for scheduling, clustering and routing models of the proposed method. Finally, we compare the results of proposed method with the results of the current method in the case study.

### i. Results of scheduling model

Section 4 described the structure of our ES for scheduling phase. For the ES we use the Jess rule engine library (available in <http://www.jessrules.com/jess/index.shtml>). We apply the scheduling model over a

planning horizon of three days. The performance measure is the average distance between all customers for each day. Table 6 presents the results of scheduling phase for three days for the case study. Fig. 5 shows the geographical distribution of the customers for each day.

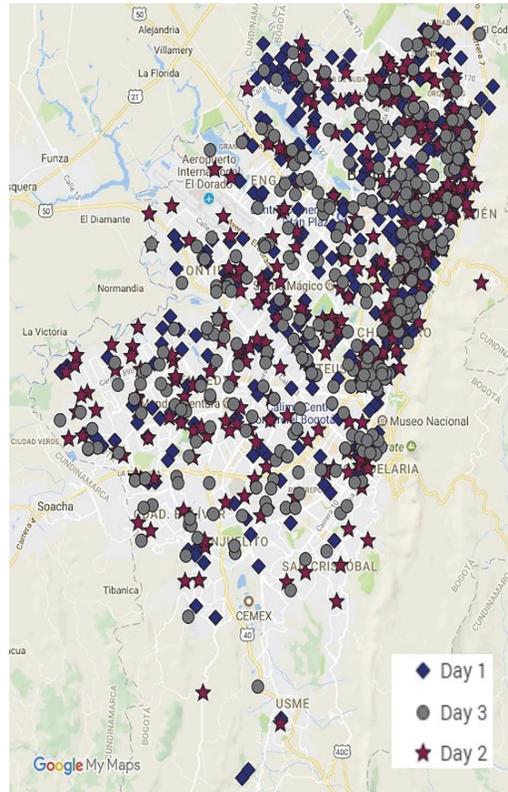


Fig. 5. Geographical distribution of customers allocated in scheduling phase

Table 6

Results of scheduling phase for three days in the case study

$p$	Number of customers	Average distance (km)
Day 1	2700	10.020
Day 2	2700	9.710
Day 3	2700	9.740

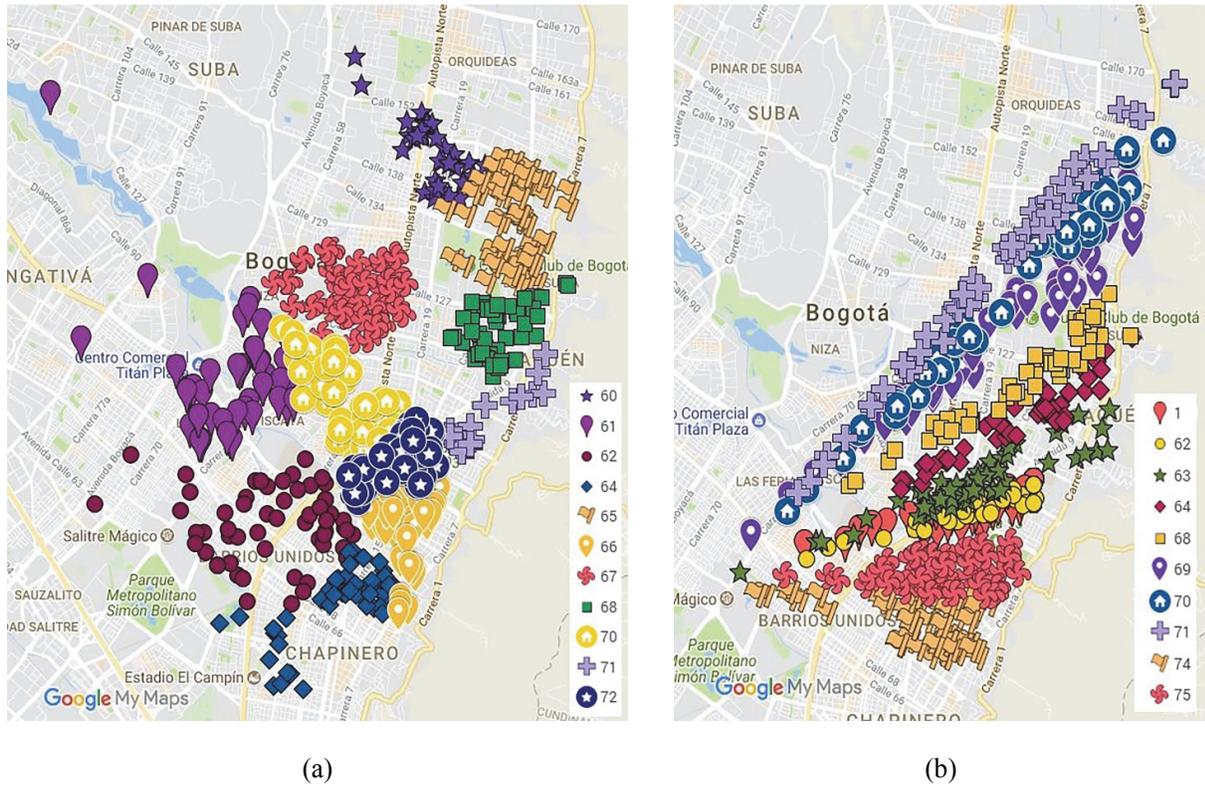
## ii. Results of clustering model

For the clustering phase, we apply both algorithms, centroid-based and sweep, for each day in the planning horizon. Table 7 shows the results of the average of distance between all customers. The average distance obtained for the centroid-based algorithm is in average 62% less than sweep algorithm, which indicates the centroid-based algorithm found better results in cooperation with the sweep algorithm for all days. To illustrate our results, Fig. 6 shows ten clusters generated with both algorithms, in a) centroid-based and b) sweep. We can observe that the centroid-based algorithm's cluster are more concentrated while the sweep algorithm's clusters are scattered. In addition, Table 8 shows the average distance of each cluster generated with the centroid-based algorithm for Day 1.

Table 7

Results of clustering phase for three days in the case study

$p$	Number of customers	Average distance (km)		Difference Centroid-based vs Sweep	% difference Centroid-based vs Sweep
		Sweep	Centroid-based		
Day 1	2700	3.160	1.200	-1.960	-62%
Day 2	2700	3.020	1.140	-1.870	-62%
Day 3	2700	2.960	1.150	-1.810	-61%
<b>Average</b>	<b>2700</b>	<b>3.040</b>	<b>1.160</b>	<b>-1.880</b>	<b>-62%</b>



(a) (b)  
**Fig. 6.** Example of geographical distribution of ten clusters generated with *a)* centroid-based algorithm, and *b)* sweep algorithm

**Table 8**  
 Results of centroid-based algorithm for Day 1

Cluster <i>k</i>	Average distance (km)	Cluster <i>k</i>	Average distance (km)	Cluster <i>k</i>	Average distance (km)
1	3.95	26	0.87	51	1.39
2	1.60	27	1.40	52	1.24
3	1.21	28	1.22	53	1.05
4	1.34	29	1.12	54	1.39
5	1.36	30	1.47	55	0.74
6	1.63	31	0.68	56	0.88
7	1.34	32	1.29	57	1.36
8	1.50	33	1.18	58	1.33
9	1.12	34	1.20	59	1.24
10	1.62	35	0.76	60	0.91
11	1.41	36	1.23	61	1.51
12	1.80	37	1.71	62	1.57
13	1.50	38	1.06	63	0.78
14	1.42	39	1.42	64	1.16
15	1.48	40	0.63	65	1.05
16	1.43	41	0.71	66	0.95
17	1.23	42	1.00	67	1.01
18	1.32	43	1.51	68	0.80
19	1.25	44	1.15	69	1.44
20	1.05	45	0.88	70	1.10
21	1.11	46	1.12	71	0.46
22	0.94	47	0.46	72	0.73
23	1.13	48	0.85	73	0.73
24	1.21	49	1.04		
25	0.69	50	1.30		

### iii. Results of routing model

For the routing phase we apply the ACO algorithm described in Section 6 for each cluster. For setting the parameters of ACO algorithm, we use the numbers suggest by Cheng and Mao (2007) as follows,

$$m = 3, q_0 = 0.99, \theta = \omega = 0.1, \beta = 0.5, \gamma = 3 \text{ and } \delta = \lambda = 0.05.$$

Table 9 shows the total routing distance for each day, the number of customers scheduled and the average distance between customers in the route. A total of 2700 customers did not schedule because the 5% of the paths were infeasible since the time windows.

**Table 9**

Results of ACO algorithm in routing phase for three days in the case study

$p$	Total routing distance (km)	Scheduled customers	Average distance between customers (km)
Day 1	2512	2613	0.96
Day 2	2237	2273	0.98
Day 3	2567	2553	1.01
<b>Average</b>	<b>2438</b>	<b>2480</b>	<b>0.98</b>

Table 10 shows the detailed results for the 73 vehicle's paths in Day 1. To illustrate our results, Fig. 7 shows the path 68 in Day 1. The visiting order is represented by colors and numbers.

**Table 10**

Results of ACO algorithm in routing phase for three days in the case study

$k$	Routing distance (km)	Scheduled customers	Average distance between customers (km)	$k$	Routing distance (km)	Scheduled customers	Average distance between customers (km)
1	0.00	0	Infeasible	38	20.19	35	0.58
2	75.18	40	1.88	39	23.59	35	0.67
3	10.67	25	0.43	40	5.34	25	0.21
4	13.30	20	0.67	41	7.97	25	0.32
5	45.21	30	1.51	42	8.75	25	0.35
6	18.50	30	0.62	43	50.92	29	1.76
7	18.29	30	0.61	44	19.62	32	0.61
8	16.54	30	0.55	45	10.89	32	0.34
9	35.05	30	1.17	46	13.60	32	0.43
10	25.14	30	0.84	47	16.92	32	0.53
11	35.67	30	1.19	48	11.76	32	0.37
12	36.06	40	0.90	49	17.66	35	0.50
13	48.75	30	1.63	50	52.07	35	1.49
14	27.35	40	0.68	51	18.90	38	0.50
15	52.78	30	1.76	52	24.23	38	0.64
16	22.01	30	0.73	53	16.25	38	0.43
17	45.82	30	1.53	54	25.17	38	0.66
18	46.45	30	1.55	55	33.33	41	0.81
19	40.38	30	1.35	56	19.61	41	0.48
20	18.32	30	0.61	57	67.15	41	1.64
21	15.93	30	0.53	58	18.44	41	0.45
22	34.39	30	1.15	59	61.89	44	1.41
23	27.25	30	0.91	60	43.23	44	0.98
24	42.52	30	1.42	61	82.77	50	1.66
25	7.83	30	0.26	62	111.54	60	1.86
26	13.76	30	0.46	63	57.84	60	0.96
27	30.77	25	1.23	64	82.65	60	1.38
28	53.58	35	1.53	65	82.97	60	1.38
29	18.63	35	0.53	66	68.23	60	1.14
30	13.15	25	0.53	67	72.80	60	1.21
31	6.68	25	0.27	68	60.51	60	1.01
32	49.14	35	1.40	69	41.89	60	0.70
33	43.17	30	1.44	70	52.51	60	0.88
34	52.70	35	1.51	71	27.25	60	0.45
35	17.26	25	0.69	72	53.70	60	0.90
36	12.83	25	0.51	73	23.92	27	0.89
37	58.32	35	1.67				

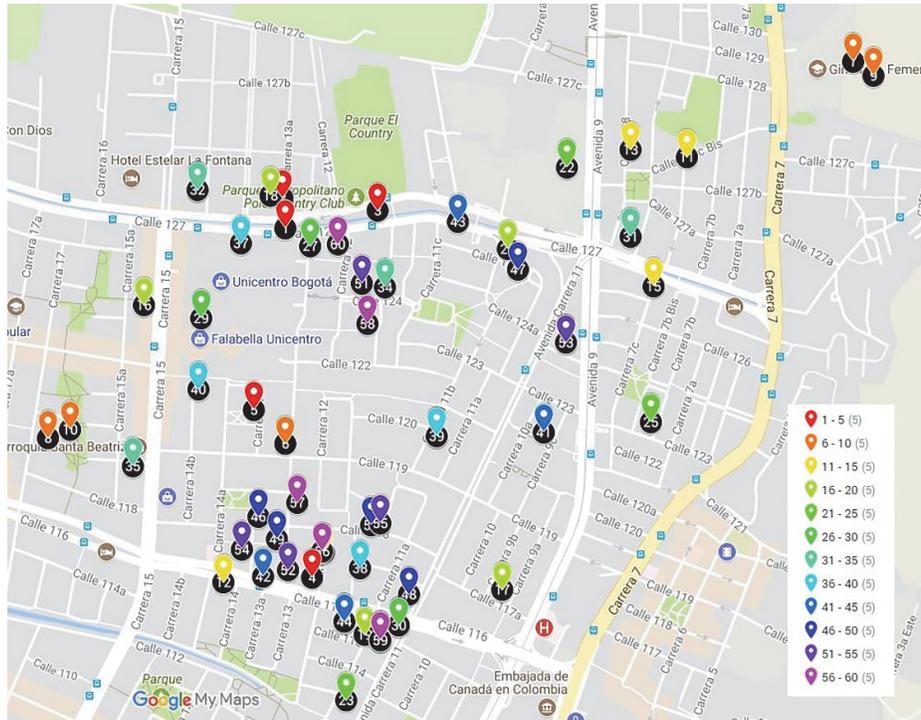
iv. Summary and comparison

We compare the average distance between the customers for each phase. Table 11 shows the results of each phase. The average distance between the customers decreases in each phase for all days. This result indicates that the customers are concentrated, thus the routing distance is reduced.

**Table 11**

Summary of average distance between customers for each phase in the case study

<i>p</i>	Scheduling phase	Clustering phase	Routing phase
Day 1	10.02	1.20	0.96
Day 2	9.71	1.14	0.98
Day 3	9.74	1.15	1.01
<b>Average</b>	<b>9.82</b>	<b>1.16</b>	<b>0.98</b>



**Fig. 7** Example of geographical distribution of path 68 in Day 1

Likewise, we compare the service level as the fulfillment of the customers with the delivery date. In the company, the actual service level is between 80% and 85%. Our approach obtains an average of 93%. The improvement is 11%. Table 12 summarizes these results for each day.

**Table 12**

Summary of service level of solution approach and actual results in the case study

<i>p</i>	Customers with delivery date	Customers with delivery date scheduled	% service level of our approach	% service level of company's method	% improvement our approach vs company's method
Day 1	546	524	96%	81%	15%
Day 2	433	367	85%	82%	3%
Day 3	333	324	97%	82%	15%
<b>Average</b>	<b>437</b>	<b>405</b>	<b>93%</b>	<b>82%</b>	<b>11%</b>

**8. Conclusions and future work**

This work has presented a hybrid expert system, clustering, and ant colony optimization approach to solve a scheduling and routing problem in courier services where a set of customers geographically

distributed are scheduled to be served in a planning horizon and a crew of operators is routing in order to visit the customers' sites and deliver a package or documents. The contribution of this paper is twofold: for the scheduling and routing point view, the problem is closer to VRPs but the problem exhibits some distinctive aspects like the presence of time windows in the customers and the vehicles, the due dates for the customers, multiples periods and distance-constrained; for the solution method point view, the courier services are faced with a large-scale problems thus an exact method is not available to find a good solution in a short time, then we propose an integration of knowledge based model and heuristics/metaheuristics techniques in order to solve the problem using the knowledge in the courier service to identify and classify the customers and clustering and routing method to allocate the available crew to attend the customers.

We proposed a solution approach which consists of the integration of knowledge based model and heuristics/metaheuristics techniques in a three phase procedure. In the first phase, a scheduling model determined the visit date for each customer over the planning horizon considering the release date, the due date and travel times between the customers and depot. We have used an expert system to solve this problem seeking to minimize the average distance between the customers. In the second phase, a clustering model used two algorithms: a centroid-based and a sweep algorithms. For each period, the model built a set of groups the customers according to the travel times, maximum load capacity and the customer's time windows. Finally, in the third phase, a routing model assigned and scheduled the customers for each cluster to one operator and found the sequence to visit them taking into account the customer's time windows and the available time of the operators. We have solved a traveling salesman problem with time windows using an ACO algorithm.

Results over a real-world case inspired in a courier company have illustrated the application of our proposed procedure in order to allocate. Our proposed method looked for reducing the average of distance between the customers as a performance measure of the concentration of the customers. These results suggest that as the average distance is reduced, the construction is increased; thus the routing distances is less and the customers are served with a better service level. The results of the case study suggest that our proposed method could outperform a procedure-based in the intuition of the planners, which does not include the expert system, and clustering algorithms.

Future work should focus on the extension of the problem to a dynamic setting, in which unexpected customer's orders or operator's features occur and it is necessary to reschedule to attend the customers. Also, it is possible to incorporate additional constraints such as the technical skills for operators, uncertainty in the travel times and visits, etc. Another opportunity to improve the solution is to incorporate other features in the expert system likes a learning module in order to add more information in the knowledge base, other classification methods based on fuzzy logic in order to manage imprecision and uncertainty, for instance the priority could be managed as linguistic variable and be represented as a fuzzy number and use a fuzzy inference system to determine the scheduling package, etc. In addition, to improve the execution time, it is possible to develop a parallelizable algorithm, which could be distributed among several computers (or multiple cores) thus reducing the computation time for all phases in our approach.

### **Acknowledgements**

We would like to acknowledge to Centro de Investigaciones y Desarrollo Científico at Universidad Distrital Francisco José de Caldas (Colombia) by supporting partially under Grant No. 2-602-468-14.

### **Appendix A. Rules of the Proposed Expert System**

Section 4 introduced the knowledge base and the inferences system of our expert system. These were built with the internal procedures and the service agreement in a real world company of courier services. For the Module of Statics Rules, the attributes that allow discarding the packages that do not have to get on the route are:

Appointment: Date of the Next appointment  
 Real status: Process state in which the package is located  
 Manageable Stock: internal rating  
 Real reason: qualification of the last visit that was made  
 Telephone management rating: last call rating

Table A1 lists the set of rules that are used in Module of Static Rules for the expert system in scheduling phase.

**Table A1**

**List of rules of Module of Static Rules**

If has an appointment date after the planning horizon then change from pending to dismissed
If has real status Route Assigned then change from pending to dismissed
If has real status Charged and verified then change from pending to dismissed
If has real status Packed then change from pending to dismissed
If has real status Delivered then change from pending to dismissed
If has real status City Forwarded then change from pending to dismissed
If has Not manageable stock then change from pending to dismissed
If has Real reason Non covered city then change from pending to dismissed
If has Real reason Customer has not ready documents then change from pending to dismissed
If has Real reason Incomplete address 2 then change from pending to dismissed
If has Real reason Incomplete delivery address then change from pending to dismissed
If has Real reason Address 2 does not exist then change from pending to dismissed
If has Real reason Delivery address does not exist then change from pending to dismissed
If has Real reason Complete management then change from pending to dismissed
If has Real reason Deficient identification of the final user then change from pending to dismissed
If has Real reason Non localized delivery then change from pending to dismissed
If has Real reason Moved then change from pending to dismissed
If has Real reason Entity request then change from pending to dismissed
If has Real reason Final user request then change from pending to dismissed
If has Real reason Rejected then change from pending to dismissed
If has Real reason No request rejection then change from pending to dismissed
If has Real reason Stock rejection then change from pending to dismissed
If has Real reason Destroyed rejection then change from pending to dismissed
If has Real reason Phone rejection then change from pending to dismissed
If has Real reason Declined then change from pending to dismissed
If has Real reason Unspecified time leaving then change from pending to dismissed
If has Real reason Holder without details then change from pending to dismissed
If has Real reason Uncovered zones then change from pending to dismissed
If has telephonic management rating Customer calls entity then change from pending to dismissed
If has telephonic management rating Customer has not ready documents then change from pending to dismissed
If has telephonic management rating Wrong customer names then change from pending to dismissed
If has telephonic management rating Returned to entity then change from pending to dismissed
If has telephonic management rating No ID then change from pending to dismissed
If has telephonic management rating Rejected then change from pending to dismissed
If has telephonic management rating Rejection because no needs then change from pending to dismissed
If has telephonic management rating Unspecified time leaving then change from pending to dismissed
If has telephonic management rating Return request then change from pending to dismissed
If has telephonic management rating Minor child owner then change from pending to dismissed

For Module of Appointments, only the following rule applies: “If the appointment date is contained in the planning horizon then change route status to scheduled and make route date equal to appointment date”.

The rules of Module 3 that were built from the experience of the pseudo-experts use the concept of onward chaining (Forward Chaining), which is why they were ordered from the rule with conditions to comply from the most strict to the more flexible, this in order to prioritize the most important and urgent packages to manage. Table A2 describes the rules of Module of Dynamic Rules.

**Table A2**

## List of rules of Module of Dynamic Rules

If there is available capability and the customer is closer to 5 customers with appointment or more and the package is expired and has 0 visits or less and has more than 5 calls then change from pending to scheduled.
If there is available capability and the customer is closer to 4 customers with appointment or more and the package is expired and has 0 visits or less and has more than 4 calls then change from pending to scheduled.
If there is available capability and the customer is closer to 3 customers with appointment or more and the package is expired and has 1 visit or less and has more than 3 calls then change from pending to scheduled.
If there is available capability and the customer is closer to 2 customers with appointment or more and the package is expired and has 1 visit or less and has more than 2 calls then change from pending to scheduled.
If there is available capability and the customer is closer to 1 customer with appointment or more and the package is expired and has 2 visits or less and has more than 1 call then change from pending to scheduled.
If there is available capability and the customer is closer to 0 customers with appointment or more and the package is expired and has 2 visits or less and has more than 0 calls then change from pending to scheduled.

**References**

- Almoustafa, S., Hanafi, S., & Mladenović, N. (2013). New exact method for large asymmetric distance-constrained vehicle routing problem. *European Journal of Operational Research*, 226(3), 386–394. Journal Article. <https://doi.org/10.1016/j.ejor.2012.11.040>
- Archetti, C., Jabali, O., & Speranza, M. G. (2015). Multi-period Vehicle Routing Problem with Due dates. *Computers & Operations Research*, 61, 122–134. <https://doi.org/10.1016/j.cor.2015.03.014>
- Braekers, K., Ramaekers, K., & Van Nieuwenhuyse, I. (2016). The vehicle routing problem: State of the art classification and review. *Computers & Industrial Engineering*, 99, 300–313. <https://doi.org/10.1016/j.cie.2015.12.007>
- Cacchiani, V., Hemmelmayr, V. C., & Tricoire, F. (2014). A set-covering based heuristic algorithm for the periodic vehicle routing problem. *Discrete Applied Mathematics (Amsterdam, Netherlands : 1988)*, 163(Pt 1), 53–64. <https://doi.org/10.1016/j.dam.2012.08.032>
- Chang, T.-S., & Yen, H.-M. (2012). City-courier routing and scheduling problems. *European Journal of Operational Research*, 223(2), 489–498. <https://doi.org/10.1016/j.ejor.2012.06.007>
- Cheng, C.-B., & Mao, C.-P. (2007). A modified ant colony system for solving the travelling salesman problem with time windows. *Mathematical and Computer Modelling*, 46(9–10), 1225–1235. Journal Article. <https://doi.org/10.1016/j.mcm.2006.11.035>
- Chu, F., Labadi, N., & Prins, C. (2006). A Scatter Search for the periodic capacitated arc routing problem. *European Journal of Operational Research*, 169(2), 586–605. <https://doi.org/10.1016/j.ejor.2004.08.017>
- Clarke, G. u., & Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12(4), 568–581. Journal Article. <https://doi.org/10.1287/opre.12.4.568>
- Díez, R. P., Gómez, A. G., & Martínez, N. de A. (2001). *Introduction to Artificial Intelligence: Expert Systems, Artificial Neural Networks, and Evolutionary Computation*. Universidad de Oviedo.
- Ding, Q., Hu, X., Sun, L., & Wang, Y. (2012). An improved ant colony optimization and its application to vehicle routing problem with time windows. *Neurocomputing*, 98, 101–107. Journal Article. <https://doi.org/10.1016/j.neucom.2011.09.040>
- Dios, M., & Framinan, J. M. (2016). A review and classification of computer-based manufacturing scheduling tools. *Computers and Industrial Engineering*, 99, 229–249. <https://doi.org/10.1016/j.cie.2016.07.020>
- Eksioglu, B., Vural, A. V., & Reisman, A. (2009). The vehicle routing problem: A taxonomic review. *Computers & Industrial Engineering*, 57(4), 1472–1483. Journal Article. <https://doi.org/10.1016/j.cie.2009.05.009>
- Farahani, R. Z., Rezapour, S., & Kardar, L. (2011). *Logistics Operations and Management*. Book, Elsevier. <https://doi.org/10.1016/C2010-0-67008-8>
- Fikar, C., & Hirsch, P. (2015). A matheuristic for routing real-world home service transport systems facilitating walking. *Journal of Cleaner Production*, 105, 300–310. <https://doi.org/10.1016/j.jclepro.2014.07.013>

- Francis, P., Smilowitz, K., & Tzur, M. (2006). The Period Vehicle Routing Problem with Service Choice. *Transportation Science*, 40(4), 439–454. <https://doi.org/10.1287/trsc.1050.0140>
- Galindres-Guanacha, L. F., Toro-Ocampo, E. M., & Gallego-Rendón, R. A. (2018). Multi-objective MDVRP solution considering route balance and cost using the ILS metaheuristic. *International Journal of Industrial Engineering Computations*, 9(1), 33–46. <https://doi.org/10.5267/j.ijiec.2017.5.002>
- Ghiani, G., Manni, E., Quaranta, A., & Triki, C. (2009). Anticipatory algorithms for same-day courier dispatching. *Transportation Research Part E: Logistics and Transportation Review*, 45(1), 96–106. <https://doi.org/10.1016/j.tre.2008.08.003>
- Gillett, B. E., & Miller, L. R. (1974). A Heuristic Algorithm for the Vehicle-Dispatch Problem. *Operations Research*, 22(2), 340–349. <https://doi.org/10.1287/opre.22.2.340>
- Glover, F. W., & Kochenberger, G. A. (2003). *Handbook of metaheuristics*. (G. Fred & Gary A. Kochenberger, Eds.) (Vol. 57). Book, Springer US. <https://doi.org/10.1007/b101874>
- Janssens, J., Van den Bergh, J., Sörensen, K., & Cattrysse, D. (2015). Multi-objective microzone-based vehicle routing for courier companies: From tactical to operational planning. *European Journal of Operational Research*, 242(1), 222–231. <https://doi.org/10.1016/j.ejor.2014.09.026>
- Jia, H., Li, Y., Dong, B., & Ya, H. (2013). An Improved Tabu Search Approach to Vehicle Routing Problem. *Procedia - Social and Behavioral Sciences*, 96, 1208–1217. <https://doi.org/10.1016/j.sbspro.2013.08.138>
- Kara, I., & Derya, T. (2015). Formulations for Minimizing Tour Duration of the Traveling Salesman Problem with Time Windows. *Procedia Economics and Finance*, 26, 1026–1034. [https://doi.org/10.1016/S2212-5671\(15\)00926-0](https://doi.org/10.1016/S2212-5671(15)00926-0)
- Kek, A. G. H., Cheu, R. L., & Meng, Q. (2008). Distance-constrained capacitated vehicle routing problems with flexible assignment of start and end depots. *Mathematical and Computer Modelling*, 47(1–2), 140–152. <https://doi.org/10.1016/j.mcm.2007.02.007>
- Krishnamoorthy, C. S., & Rajeev, S. (1996). *Artificial intelligence and expert systems for engineers*. Boca Raton: CRC Press.
- Küçükoğlu, İ., & Öztürk, N. (2015). An advanced hybrid meta-heuristic algorithm for the vehicle routing problem with backhauls and time windows. *Computers & Industrial Engineering*, 86, 60–68. <https://doi.org/10.1016/j.cie.2014.10.014>
- Kusiak, A. (1990). *Intelligent manufacturing systems*. London: Prentice Hall International.
- Lin, C., Choy, K. L., Ho, G. T. S., Lam, H. Y., Pang, G. K. H., & Chin, K. S. (2014). A decision support system for optimizing dynamic courier routing operations. *Expert Systems with Applications*, 41(15), 6917–6933. <https://doi.org/10.1016/j.eswa.2014.04.036>
- Lin, C. K. Y. (2011). A vehicle routing problem with pickup and delivery time windows, and coordination of transportable resources. *Computers & Operations Research*, 38(11), 1596–1609. <https://doi.org/10.1016/j.cor.2011.01.021>
- Liu, R., Xie, X., Augusto, V., & Rodriguez, C. (2013). Heuristic algorithms for a vehicle routing problem with simultaneous delivery and pickup and time windows in home health care. *European Journal of Operational Research*, 230(3), 475–486. <https://doi.org/10.1016/j.ejor.2013.04.044>
- López-Ibáñez, M., & Blum, C. (2010). Beam-ACO for the travelling salesman problem with time windows. *Computers & Operations Research*, 37(9), 1570–1583. Journal Article. <https://doi.org/10.1016/j.cor.2009.11.015>
- López-Santana, E. R., & Méndez-Giraldo, G. A. (2016). A Knowledge-Based Expert System for Scheduling in Services Systems. In J. C. Figueroa-García, E. R. López-Santana, & R. Ferro-Escobar (Eds.), *Applied Computer Sciences in Engineering WEA 2016* (pp. 212–224). Springer International Publishing AG. [https://doi.org/10.1007/978-3-319-50880-1\\_19](https://doi.org/10.1007/978-3-319-50880-1_19)
- López-Santana, E. R., & Romero Carvajal, J. de J. (2015). A hybrid column generation and clustering approach to the school bus routing problem with time windows. *Ingeniería*, 20(1), 111–127. <https://doi.org/http://dx.doi.org/10.14483/udistrital.jour.reving.2015.1.a07>
- Malmberg, C. J. (2000). Current modeling practices in bank courier scheduling. *Applied Mathematical Modelling*, 24(4), 315–325. [https://doi.org/10.1016/S0307-904X\(99\)00044-X](https://doi.org/10.1016/S0307-904X(99)00044-X)

- Méndez-Giraldo, G., Álvarez, L., Caicedo, C., & Malaver, M. (2013). *Expert system for scheduling production-research and development of a prototype* (1st ed.). Colombia: Universidad Distrital Francisco José de Caldas.
- Nagarajan, V., & Ravi, R. (2012). Approximation algorithms for distance constrained vehicle routing problems. *Networks*, 59(2), 209–214. <https://doi.org/10.1002/net.20435>
- Patiño Chirva, J. A., Daza Cruz, Y. X., & López-Santana, E. R. (2016). A Hybrid Mixed-Integer Optimization and Clustering Approach to Selective Collection Services Problem of Domestic Solid Waste. *Ingeniería*, 21(2), 235–247. <https://doi.org/http://dx.doi.org/10.14483/udistrital.jour.reving.2016.2.a09>
- Pereira, F. B., & Tavares, J. (2009). *Bio-inspired Algorithms for the Vehicle Routing Problem*. (F. B. Pereira & J. Tavares, Eds.). Springer-Verlag Berlin Heidelberg. <https://doi.org/10.1007/978-3-540-85152-3>
- Pillac, V., Guéret, C., & Medaglia, A. L. (2012). A parallel matheuristic for the technician routing and scheduling problem. *Optimization Letters*, 1–11. <https://doi.org/10.1007/s11590-012-0567-4>
- Pureza, V., Morabito, R., & Reimann, M. (2012). Vehicle routing with multiple deliverymen: Modeling and heuristic approaches for the VRPTW. *European Journal of Operational Research*, 218(3), 636–647. <https://doi.org/10.1016/j.ejor.2011.12.005>
- Rincon-Garcia, N., Waterson, B. J., & Cherrett, T. J. (2017). A hybrid metaheuristic for the time-dependent vehicle routing problem with hard time windows. *International Journal of Industrial Engineering Computations*, 8(1), 141–160. <https://doi.org/10.5267/j.ijiec.2016.6.002>
- Rodríguez-Vásquez, W. C., López-Santana, E. R., & Méndez-Giraldo, G. A. (2016). Proposal for a Hybrid Expert System and an Optimization Model for the Routing Problem in the Courier Services. In J. C. Figueroa-García, E. R. López-Santana, & R. Ferro-Escobar (Eds.), *Applied Computer Sciences in Engineering WEA 2016* (pp. 141–152). Springer International Publishing AG. [https://doi.org/10.1007/978-3-319-50880-1\\_13](https://doi.org/10.1007/978-3-319-50880-1_13)
- Rodríguez, S., Correa, D., & López-Santana, E. (2015). An Alternative Iterative Method to Periodic Vehicle Routing Problem. In S. Cetinkaya and J. K. Ryan (Ed.), *IIE Annual Conference and Expo 2015* (pp. 2001–2010).
- Sahin, S., Tolun, M. R., & Hassanpour, R. (2012). Hybrid expert systems: A survey of current approaches and applications. *Expert Systems with Applications*, 39(4), 4609–4617. <https://doi.org/10.1016/j.eswa.2011.08.130>
- Shin, K., & Han, S. (2012). A Centroid-based Heuristic Algorithm for the Capacitated Vehicle Routing Problem. *Computing and Informatics*, 30(4), 721–732. Retrieved from <http://www.cai.sk/ojs/index.php/cai/article/view/192>
- Sprenger, R., & Mönch, L. (2012). A methodology to solve large-scale cooperative transportation planning problems. *European Journal of Operational Research*, 223(3), 626–636. <https://doi.org/10.1016/j.ejor.2012.07.021>
- Tlili, T., Faiz, S., & Krichen, S. (2014). A Hybrid Metaheuristic for the Distance-constrained Capacitated Vehicle Routing Problem. *Procedia - Social and Behavioral Sciences*, 109, 779–783. <https://doi.org/10.1016/j.sbspro.2013.12.543>
- Toth, P., & Vigo, D. (2002). *The vehicle routing problem*. Optimization (Vol. 9). Philadelphia: SIAM. <https://doi.org/10.1137/1.9780898718515>
- Turban, E. (1989). *Decision support and expert systems: management support systems* (2nd ed.). Book, Prentice Hall PTR.
- Wagner, W. P. (2017). Trends in expert system development: A longitudinal content analysis of over thirty years of expert system case studies. *Expert Systems with Applications*, 76, 85–96. <https://doi.org/10.1016/j.eswa.2017.01.028>
- Yan, S., Lin, J.-R., & Lai, C.-W. (2013). The planning and real-time adjustment of courier routing and scheduling under stochastic travel times and demands. *Transportation Research Part E: Logistics and Transportation Review*, 53, 34–48. <https://doi.org/10.1016/j.tre.2013.01.011>
- Yu, B., Yang, Z. Z., & Yao, B. Z. (2011). A hybrid algorithm for vehicle routing problem with time windows. *Expert Systems with Applications*, 38(1), 435–441.



© 2018 by the authors; licensee Growing Science, Canada. This is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).