A PROBLEM OF ECOLOGICAL DYNAMIC LOGISTICS

José M. Cadenas, jcadenas@um.es
Mª Carmen Garrido, carmengarrido@um.es
Raquel Martínez, raquel.m.e@um.es
Dpto. Ingeniería de la Información y las Comunicaciones. Facultad de Informática.
Universidad de Murcia, 30071 Espinardo (Murcia). Spain.
Tel: +34 868 884847 / 4629 / 7866 -- Fax: +34 868 884151

Enrique Muñoz, enrique.munoz@softcomputing.es
European Center for Soft Computing
33600 Mieres (Asturias). Spain.
Tel: +34 985 456545 -- Fax: +34 985 456699

ABSTRACT

Nowadays companies have to be competitive to survive, offering their customers a more customized and complete service. Moreover, society is increasingly more aware of the need to solve problems in an ecological way trying to reduce pollution/environmental impact. Furthermore, when we try to solve management problems in crowded cities, companies expect their solutions not to depend excessively on traffic conditions and not to have an effect on the worsening of these conditions.

In this paper we study how to handle the aforementioned requirements, defining the problem of ecological dynamic logistics in urban environments. In this problem, a fleet of agile environmentally friendly vehicles have to pick-up and deliver items. However, these vehicles have a limited storage capacity, and they need the support of larger environmentally friendly vehicles, mobile warehouses, to increment their range. The problem is solved by planning the routes for both delivery/pick-up vehicles and mobile warehouses in a coordinated manner, allowing the incorporation of new items during execution of the plan. To obtain a solution for this problem we propose to use the Ant Colony Optimization metaheuristic.

INTRODUCTION

In a competitive economic environment, where the number of alternatives for companies and consumers is greater than ever, it is crucial for the companies to effectively organize their logistics activities in order to face the changing needs of the XXI century. Nowadays, companies work in an environment where customer's requirements are very demanding, and in particular, they cannot allow themselves to have problems with the delivery of their goods. Delivery requirements may include, in many occasions, to perform them on the same day in which they are sent. For this reason, courier companies must make sure to provide really efficient services and to offer to the customers a more detailed and specific support.

Additionally, today's concerns about the environment, are encouraging the search of solutions that try to reduce emissions, and consequently, environmental impact to a minimum. In this sense, courier companies can find new business opportunities by adopting more environmentally friendly policies. A good way to adapt to these requirements is to use low or no-emission vehicles to deliver their items, as for example, bicycles or electric vehicles. Along with these ecological requirements, courier companies are interested in finding solutions that can overcome cities' traffic problems, which may include, changing traffic conditions, blocked streets… For this reason it is advisable to use agile vehicles like bicycles or small motorcycles that can move faster on a city, and that are less affected by traffic conditions. However, these vehicles have a limited capacity to carry items, and to improve their range of action it is useful to rely on bigger vehicles that can carry more items. These other vehicles may also be environmentally friendly, as electric vans or cars.

Taking all these requirements into account, a new niche market is appearing, which encourages the appearance of courier companies that focus on the delivery and pick up of items on cities, using a flexible and ecological approach. However this is a complex problem that includes many challenges: 1) it has to permit the traceability of goods, effectively and on time; 2) it needs to ensure a maximum utilization of resources; 3) it has to allow to make decisions on real time, like updating orders while the process has started; 4) it has to avoid as much as possible problems due to cities' traffic conditions, taking into account the availability of online information; and 5) it needs to use ecological vehicles that can reduce environmental impact to a minimum.

Given this idea, this paper focuses on the problem of organizing the fleet of vehicles that carry out the delivery/pick-up activities. We have named this problem "ecological dynamic logistics in urban environments". In order to solve this problem we have to define the routes that must be followed by the vehicles. Additionally, these routes have to reduce the cost for the company and provide an optimum service in terms of time. We propose to perform the delivery/pick-up operations using agile ecological vehicles (called terminal vehicles), which as was previously stated, have a limited storage capacity. For this reason, they must be supported by higher capacity vehicles (called mobile warehouses) that allow them to gather new items which must be delivered and to deposit the items that have already been picked up. This approach allows to reduce both the costs for the company and the environmental impact of their actions. Hence, the problem consists in planning the routes of both kinds of vehicles in a coordinated manner, given a set of delivery and pick-up points, and a set of points were mobile warehouses can wait for terminal vehicles. Additionally, the solution has to allow the adding of new elements of delivery/pick-up during the execution of a plan.

In this paper we analyze this problem and propose to use Ant Colony Optimization to solve it in approximate way. In particular, this paper shows a feasibility study for a company that wants to develop this new type of business in cities. The remainder of the paper is organized as follows: in Section "Background" we study related works, in Section "Understanding the Problem" we detail the problem and explain its different parameters and constraints, in Section "Methodology of Resolution" we describe the proposed algorithm to provide a feasible solution to this problem, in addition we show how to construct the solution of the problem and possible actions to take in case of not finding a feasible solution. In Section "Application Examples" we show examples of application, and finally, in Section "Conclusions" we present the conclusions of this study.

BACKGROUND

Logistics problems have been and continue being an important research topic that has had an important impact in everyday life. In this paper we are interested on a particular kind of logistics problems known as vehicle routing problems. Some examples of these kind of problems are the popular Travelling Salesman Problem (Puris, Bello, Martinez & Nowe, 2007) or the problem of finding optimal paths for the delivery and pick up of items (Montemanni, Lucibello & Gambardella, 2007).

Vehicle routing problems have been applied to a wide variety of real world problems, such as garbage collection in a city, (Nuortioa, Kytöjokib, Niskaa & Bräysyb, 2006), or the distribution of aid after a natural disaster, such as an earthquake, (Tzeng, Cheng & Huang, 2007), considering that aid must first reach the worst affected areas. Among the many application areas we have to emphasize those that paid attention to environmentally friendly solutions, as the works of (Crainic, Ricciardi & Storchi, 2009; Kenworthy, 2006) that try to solve logistics problems in urban centers using ecological approaches.

The main problem when we try to solve this kind of problems is that most of them belong to the class of NP-complete problems, which means that there are no fast solutions to solve them. Consequently, when the size of the problem is relatively large, its resolution becomes intractable using exact methods. For this reason, many researchers have proposed to solve them using approximation methods. Among these methods we can emphasize metaheuristics. Several studies have shown that metaheuristics are successful tools for providing reasonably good solutions to optimization problems (in some cases excellent) using a moderate number of resources (Donati, Montemanni, Casagrande, Rizzoli & Gambardella, 2008; Silva, Sousa & Runkler, 2008; Cadenas, Garrido & Muñoz, 2009).

In particular, there is a metaheuristic that is especially well suited for the resolution of vehicle routing problems, Ant Colony Optimization (Dorigo & Stützle, 2003; Silva, Sousa & Runkler, 2008). This metaheuristic has been used to solve problems similar to the one proposed in this study. For example, in (Montemanni, Gambardella, Rizzoli & Donati, 2005) it was used to solve the problem of Dynamic Routing of Vehicles. In (Bella & McMullen, 2004), authors employ Ant Colony Optimization to address a Vehicle Routing problem. In (Fuellerera, Doernera, Hartla & Iorib, 2009) Ant Colony Optimization was used to solve the Vehicle Routing problem with Charge. In (Gajpal & Abad, 2009) an Ant Colony Multisystem solving the Vehicle Routing problem with Return Network is proposed. As in the problem that we propose, they have end users with certain items to pick up and/or deliver. To solve this problem, two multipaths of local search using MACS (Ant Colony Multisystem) are used.

For this reason, in this paper we propose to use Ant Colony Optimization to solve the problem of ecological dynamic logistics in urban environments.

UNDERSTANDING THE PROBLEM

Introduction

In this article we face the problem of studying a viability of the ecological dynamic logistics in urban environments that serves as a reference for companies to establish a new standard in the field.

In the problem, we have a set of items that have to be delivered and another set of items that have to be picked up, both types of items are represented by points on a map and are called delivery and pick-up points, respectively. As previously indicated, the problem considers the use of agile ecological terminal vehicles with limited storage capacity. These terminal vehicles start working from the company head office, visiting delivery and pick-up points until they run out of storage capacity. When that occurs they have to regenerate it by visiting the head office or a mobile warehouse, where they can leave picked up items and gather new items for delivery. This process continues until the terminal vehicles have reached their maximum working hours, taking into account that they have to finish their working day in the head office.

On the other hand, mobile warehouses start from the company head office and have a series of appointments with terminal vehicles, in which they gather picked up items and provide new delivery items. These appointments can take place only at certain locations, where vehicles are allowed to stop for a certain period of time to load/unload their cargo. We call these points, service points. Hence, mobile warehouses' routes only include service points and the head office, which can be considered a special type of service point. As in the case of terminal vehicles, mobile warehouses also have a maximum number of working hours and have to finish their working day in the head office.

One aspect to consider is that there may exist many possible service points in a city, however, we assume that before starting the resolution of the problem we know those which are more feasible.

In addition, it is important to note that the distances between all points (service, delivery, pick-up and head office points) are assumed to be known. In a real environment we will use positioning systems and updated maps, capable of indicating the distance between any two points.

Apart from all previous requirements, companies need flexible solutions that allow them to modify the routes obtained in case there appear very valuable new operations. In order to face such challenge we propose to take a snapshot of the current status of the problem and restart the resolution of the problem, considering only the new tasks and those delivery and pick-up points that have not been handled yet, and taking into account current positions of terminal vehicles and mobile warehouses.

Characterization of the problem

This section defines the problem of ecological dynamic logistics in urban environments and introduces the concepts used in this paper. The formalization is similar to that used for pickup and delivery routing problems with time windows (Bent & Van Hentenryck, 2006).

Pickup, delivery and service points: the problem is defined in terms of n customer points (including deliveries and pick-ups), a set of s service points, and a head office ($depot$). We use $Customers$ to define the set of customer points, $Services$ to define the set of service points, and $Sites$ to denote the whole set of points. We use $Customer^p$ and $Customer^d$ to designate pick-up and delivery points, respectively. The cost of traveling from any two points is denoted by c_{ij}. Every customer point has a service time z_i. On the other hand, terminal vehicles require a time g to regenerate their capacity at a service point.

Vehicles: the problem is defined in terms of two sets of vehicles. T represents the set of terminal vehicles, each vehicle having a capacity Q_t. On the other hand M represents the set of mobile warehouses, each one having a capacity Q_m.

Terminal vehicle Routes: a terminal vehicle route starts at the $depot$, visits a set of customers and, possibly, service points and returns to the $depot$. In other words, the route of a terminal vehicle i is a sequence

$$r_t^i = < 0, a_1^i, \dots, a_h^i,\ b_1^i, a_{h+1}^i, \dots, a_k^i, b_l^i, a_{k+1}^i, \dots, a_o^i, 0 >$$

where $a_j^i \in Customers,\ j = 1, \dots, o,$ and all a_j^i are different, $b_j^i \in Services,\ m = 1, \dots, l,$ which are not required to be different, and 0 represents the $depot$. We define a subroute as a subset of a route that includes all customer points between two consecutive service points. The customer points of a route r_t^i, denoted by $cust(r_t^i)$, is the set $\{a_1, \dots, a_o\}$. The service points of a route, denoted by $serv(r_t^i)$, is the set $\{b_1, \dots, b_l\}$. The size of a route, denoted by $|r_t^i|$, is the number of customers $|cust(r_t^i)|$. The travel cost of a route r_t^i, denoted by $c_t(r_t^i)$, is the cost of visiting all its customers, i.e.,

$$c_t(r_t^i) = c_{oa_1^i} + c_{a_1^i a_2^i} + \dots + c_{a_h^i b_1^i} + c_{b_1^i a_{h+1}^i} + \dots + c_{a_k^i b_1^i} + c_{b_1^i a_{k+1}^i} + \dots + c_{a_o^i o}$$

if the route is not empty, and 0 otherwise.

Terminal vehicle routing plan: A terminal vehicle routing plan is a set of routes $\{r_t^1, \dots, r_t^u\}$ with $u \leq |T|$ visiting every customer exactly once, i.e.,

$$\bigcup_{i=1}^{u} cust(r_t^i) = Customers$$

$$cust(r_t^i) \cap cust(r_t^j) = \emptyset \qquad \forall i, j, \qquad 1 \leq i, j \leq |T|$$

We define a visited-service-point as a service point visited by terminal vehicle i for the j-th time, and denote it b_j^i. We define the set s_σ, which includes all visited-service-points used in the routing plan σ. In other words, s_σ contains $\{b_1^1, \dots, b_l^1, b_1^2, \dots, b_x^2, \dots, b_1^n, \dots, b_y^n\}$, where b_j^i represents the j-th service point visited by terminal vehicle i. To simplify the notation we will consider $s_\sigma = \{v_i, \dots, v_d\}$ where d is the total number of visited-service-points. In addition we define the set $x_\sigma = Customers \cup s_\sigma$, which includes all the points visited in the terminal vehicle routing plan, and call a point $x_i \in x_\sigma,\ i = 1, \dots, (n + d)$ a visited-point. Note that in a terminal vehicle routing plan, every visited-point is visited exactly once.

Observe that a terminal vehicle routing plan assigns a unique successor and predecessor to every visited-point. The successor and predecessor of customer i in routing plan σ are denoted by $succ(i, \sigma)$ and $pred(i, \sigma)$. For simplicity, our definitions assume an underlying routing plan σ and we use i_σ^+ and i_σ^- to denote the successor and predecessor of i in σ. We use the function $presubroute(x)$, which returns the set of customer points that precede a visited-point p in the subroute where it is included, and the function $postsubroute(x)$ which returns the set of customer points that succeed visited-point x in the subroute where it is included. Note that if p is a visited-service-point, it is included in two subroutes, in this case $presubroute(x)$ returns the set of customer points of the subroute that ends in x, and $postsubroute(x)$ returns the set of customer points of the subroute that starts in x.

Mobile warehouse Routes: a mobile warehouse route starts at the $depot$, visits a set of visited-service-points included in a terminal vehicle routing plan σ, and returns to the $depot$. In other words, a mobile warehouse route is a sequence $r_w^i =< 0, v_1, \dots, v_e, 0 >$, where all v_i represent visited-service-points, and are required to be different. The service points of a route $r_w^i =< 0, v_1^i, \dots, v_e^i, 0 >$ is the set $\{v_1^i, \dots, v_e^i\}$, denoted by $serv(r_w^i)$. The function $preroute(v)$ returns the set of visited-service-points that precede the visited-service-point v in the mobile warehouse route in which it is included, without including the $depot$. The function $postroute(v)$ returns the set of visited-service-points that succeed the visited-service-point v in the mobile warehouse route in which it is included, without including the $depot$. The travel cost of a route $r_w^i =< 0, v_1, \dots, v_e, 0 >$ denoted by $c_w(r_w^i)$ is the cost of visiting all its service points, i.e., $c_w(r_w^i) = c_{0v_1^i} + c_{v_1^i v_2^i} + \dots + c_{v_e^i 0}$, if the route is not empty, and zero otherwise.

Mobile warehouse routing plan: A mobile warehouse routing plan is a set of mobile warehouse routes $\{w_r^1, \dots, w_r^m\}$, with $m \leq |M|$, visiting every visited-service-point included in the set s_σ of a terminal vehicle routing plan σ exactly once, i.e.,

$$\bigcup_{i=1}^{m} serv(r_w^i) = s_\sigma$$

$$serv(r_w^i) \cap serv(r_w^j) = \emptyset \qquad \forall\, i, j, \qquad 1 \leq i, j \leq |M|$$

Observe that a mobile warehouse routing plan φ assigns a unique successor and predecessor to every service point. These successors and predecessors are sites. The successor and predecessor of customer i in routing plan φ are denoted by $succ(i, \varphi)$ and $pred(i, \varphi)$. For simplicity, our definition assume an underlying routing plan φ and we use i_φ^+ and i_φ^- to denote the successor and predecessor of i.

Full routing plan: A full routing plan is a tuple $< \sigma, \varphi >$, where σ is a terminal vehicle routing plan and φ is a mobile warehouse routing plan.

Time constraints: terminal vehicles that visit a service point impose some time constraints to mobile warehouses. A visited-service-point v_i of a terminal vehicle routing plan σ is visited at time δ_{v_i}. A mobile warehouse that includes v_i in its route must arrive at that point before δ_{v_i}. It may arrive earlier but it has to wait for the terminal vehicle until time δ_{v_i}. In addition, it has to spend a time g regenerating the terminal vehicle capacity. The time at which a visited-point $i \in x_\sigma$ is visited is calculated recursively as

$\delta_0 = 0$

$\delta_i = \delta_{i_\sigma^-} + c_{i_\sigma^- i} + z_i \qquad if\ x_i \in Customers$

$\delta_i = \delta_{i_\sigma^-} + c_{i_\sigma^- i} + g \qquad if\ x_i \in Service$

On the other hand the earliest time at which a mobile warehouse arrives to a visited-point i, which also is a visited-service-point, is calculated recursively as

$\alpha_0 = 0$

$\alpha_i = \delta_{i_\varphi^-} + g + c_{i_\varphi^- i}$

A full routing plan $< \sigma, \varphi >$ satisfies the time constraints if $\alpha_i \leq \delta_i \quad \forall i \in s_\sigma$.

Working day constraints: Each terminal vehicle i can deliver and pick up items for a given period of time $[s_i, l_i]$, where s_i and l_i represent its start and the leaving times, which correspond to its working hours. If we define β_i the time of the last visit of terminal vehicle i to the $depot$ (which indicates the time when its working day ends), a full routing plan $< \sigma, \varphi >$ satisfies the working day constraints if $\beta_i \leq l_i\ ,\ \forall i \in T$.

Analogously, each mobile warehouse i can work for a given period of time $[d_i, f_i]$, where d_i and f_i represent the departure and the finish time, which correspond to its working hours. If we define γ_i as the time of the last visit of mobile warehouse i to the $depot$ (which indicates the time when its working day ends), a full routing plan $< \sigma, \varphi >$ satisfies the working day constraints of mobile warehouses if $\gamma_i \leq d_i\ ,\ \forall i \in M$.

Capacity constraints: The used capacity of a terminal vehicle at visited-point x includes the number of pick-ups preceding x in the subroute that includes x, the number of deliveries succeeding x in the subroute that includes x, and one more item if x is a pick-up. The used capacity is calculated as $q_t(x) = \sum_{i \in presubroute(x)} pickup(i) + \sum_{i \in postsubroute(x)} delivery(i) + pickup(x)$ where $pickup(i)$ is a function that returns 1 if $i \in Customers^p$ and 0 otherwise. Analogously $delivery(i)$ is a function that returns 1 if $i \in Customers^d$ and 0 otherwise. The constraints on terminal vehicles capacity are satisfied if $q_t(x) \leq Q_t\ ,\ \forall x \in Customers$.

The used capacity of a mobile warehouse at a visited-service-point v includes the number of pick-ups of the terminal vehicles that regenerate their capacity using the mobile warehouse in the service points preceding v, the number of deliveries of the terminal vehicles that will regenerate their capacity using the mobile warehouse in the service points succeeding v, and the pick-ups of the terminal vehicle that regenerates its capacity in service point v. It is calculated as

$q_w(v) = \sum_{i \in preroute(v)} \sum_{j \in presubroute(i)} pickup(j) + \sum_{i \in postroute(x)} \sum_{j \in postsubroute(i)} delivery(j) + \sum_{i \in presubroute(v)} pickup(i)$.

The constraints on mobile warehouses capacity are satisfied if $q_w(v) \leq Q_w,\ \forall v \in s_\sigma$.

Final formulation: A solution to the proposed problem is a full routing plan $< \sigma, \varphi >$ that satisfies the time constraints, the working day constraints and the capacity constrains, i.e.,

$\alpha_i \leq \delta_i \qquad \forall i \in s_\sigma$

$\beta_i \leq l_i \qquad \forall i \in T$

$q_t(v) \leq Q_t \qquad \forall v \in Customers$

$q_w(v) \leq Q_w \qquad \forall w \in s_\sigma$

The size of a terminal vehicle routing plan σ, denoted by $|\sigma|$ is the number of non-empty terminal vehicle routes in σ, i.e., $\{r_t^i \in \sigma\ /\ cust(r_t^i) \neq \emptyset\}$. The size of a mobile warehouse routing plan φ, denoted by $|\varphi|$ is the number of non-empty mobile warehouses routes in φ, i.e., $\{r_w^i \in \varphi \: / \: serv(r_w^i) \neq \emptyset\}$. The cost of a terminal vehicle routing plan is calculated as $\sum_{r_t^i \in \sigma} c_t\,(r_t^i)$. On the other hand, the cost of a mobile warehouse routing plan is calculated as $\sum_{r_w^i \in \varphi} c_w\,(r_w^i)$. The objectives of the problem are:

$$\min |\sigma| \tag{1.1}$$

$$\min |\varphi| \tag{1.2}$$

$$\min \sum_{r_t^i \in \sigma} c_t\left(r_t^i\right) \tag{1.3}$$

$$\min \sum_{r_w^i \in \varphi} c_w\,(r_w^i) \tag{1.4}$$

METHODOLOGY OF RESOLUTION

To solve the problem, a specific exact method to find the optimal solution could be used. However, the business expectations indicate that the size of the instances of the problem can grow to intractable sizes (distribution in a medium city, as Murcia, or big as Madrid in Spain). This fact means the exact method needs a lot of time (of the order of days) to obtain a solution. In cases where it is not possible to obtain an optimal solution in a feasible time, it is necessary to use techniques that are able to obtain high-quality solutions in a reasonable time. Among these techniques, metaheuristics stand out. Metaheuristics are intelligent strategies to design or improve very general heuristic procedures with high performance. Such strategies have been widely used in the resolution of all kinds of optimization problems obtaining excellent results. Among metaheuristics various techniques stand out such as Genetic Algorithms, Tabu Search, the Particle Swarm Optimization, Ant Colony Optimization, ... For the problem to be solved, the metaheuristic that may be more appropriate is the Ant Colony Optimization (ACO) (Dorigo & Stützle, 2003; Silva, Sousa & Runkler, 2008), since it was specifically designed to solve problems of minimization of paths and has been applied with great success to several problems such as the Traveling Salesman problem, Packet Routing problems in Telecommunication Networks, problems of Pick-up and Delivery. For this reason, we propose to use it (Dorigo, Maniezzo & Colorni, 1996; Dorigo & Stützle, 2003).

Ant Colony

Ants are social insects that live in colonies and whose behavior is directed to the development of the colony as a whole rather than to individual development, (Bella & McMullen, 2004). An interesting feature of the behaviour of ant colonies is how they can find the shortest path between the nest and the food, because they are blind. Ants are able to do so because, in their path, they deposit a substance called pheromone that they can smell. This trail allows ants to find the best path from the food back to their nest. In particular, ants move in a random way and choose those paths where the pheromone is stronger with a higher probability. Thus, the most promising paths accumulate greater amounts of pheromone because they are used by more ants, while the less promising paths lose pheromone by evaporation because they are gradually abandoned by ants. Natural ants are able to solve shortest path problems with minimum effort. The algorithms of Ant Colony Optimization (ACO) try to reproduce the behaviour of real ants to solve such problems.

Principles of the Ant Colony

Each ant is a probabilistic mechanism able to construct solutions. To construct a solution each ant uses:

- Artificial pheromone trails which change with time to reflect the experience of the other ants in solving the problem.
- Some heuristic information about the particular instance of the problem.

The basic operating mode of an ACO algorithm is as follows: the ants of the colony move concurrently and asynchronously, through adjacent states of the problem. This movement is performed following a transition rule which is based on local information available to the components (nodes). This local information includes heuristic information and the pheromone trails to guide the search. By moving through the construction graph, ants incrementally build solutions. Once each ant has generated a solution, this is evaluated and the ant can deposit a quantity of pheromone which is a function of the quality of the solution. This information will guide the search for other ants in the colony in the future. In addition, the operating mode of a generic ACO algorithm includes two additional procedures, the evaporation of the pheromone trails and the daemon actions. The pheromone evaporation is performed by the environment and is used as a mechanism to avoid the local optima in the search and allows the ants to find and explore new regions of space. The daemon actions are optional actions, which have no natural counterpoint, able to implement tasks from a global perspective that the ants cannot perform because they offer a local perspective. Some of these actions used in ACO algorithms are: monitoring the quality of the solutions generated and deposit an additional amount of pheromone only in transitions/components associated with some solutions, or apply a local search procedure that improves the obtained solutions before updating the pheromone trail.

The generic structure of an ACO is presented in Algorithm 1.

Algorithm 1: ACO_metaheuristic

```
ACO(N:integer; F:initial pheromone) = x_best
var
   C     /*neighborhood structure of an ant*/
   M     /*collective memory structure of ants */
   P     /*probability structure for each movement of an ant */
   pos   /*ants positions*/
begin
   repeat
        for i:=1 to N do
             {M}:= UpdateMemory(M);
             while not Solution do
                  {C}:= Neighborhood(pos);
                  {P}:= EvaluateNeighbours(C,M);
                  {pos}:= AntMovement(P);
                  {F}:= DepositPheromone(pos);
                  {x}:= AddComponent(pos);
             end
             {x}:= UpdateSolutions(x);
        end
        {F}:= EvaporatePheromone(x,F);
        {F}:= DaemonActions(x,F);
   until termination;
   {x_best}:= ChooseBest(x);
end
```

There are various implementations of the concept of ACO, one of the most competitive being the MAX-MIN ant system (MM-AS). This MM-AS version will be used to solve the problem proposed in this paper. MM-AS provides a good balance between exploration and exploitation and reduces the possibility to reach local optima, obtaining very competitive results.

MM-AS has the following particular aspects:

- An update of the pheromone trails is applied offline. That is, after all ants have constructed their solutions, each pheromone trail suffers evaporation and the pheromone is deposited only on the best solution which can be either the best global solution or the best solution of the iteration.
- Possible values for the pheromone trails are limited to the range $[\tau_{min}, \tau_{max}]$, so as to reduce the likelihood of the algorithm stalling. Also, it should be noted that in order to increase the exploration of new solutions, a re-initialization of these trails is sometimes used.
- Pheromone trails are initialized to a high value (the maximum allowed τ_{max}), instead of a lower one as usual. This helps in diversification of the search since the initial differences in the trails will be small.

Applying the algorithm to the problem

As we have discussed above, the metaheuristic that will be used to solve the problem is the Ant Colony Optimization in its MAX-MIN version.

Construction of the solution using ants

The process to obtain solutions is constructive, and each ant has to build its own solution in each iteration. This process is supported by the use of a pheromone matrix that links all points of the problem ($Customers \cup Services$) between them, regardless of whether they represent points of pick-up, delivery or service. Thus the cell (i, j) of the matrix represents the suitability of moving from points x_i to point x_j.

This pheromone trail matrix associated to connections is initialized to the maximum value of range (as we have mentioned above): $\tau_{ij} = \tau_{max}, \quad \forall\ (i, j)$ links

The construction of the solution is divided into three subprocesses:

- First the route of a terminal vehicle is built until its capacity is exhausted (both delivery and pick-up). It should be noted that all terminal vehicles start from the head office.
- Second, after exhausting the capacity, we have to decide how to regenerate it (charging items to deliver and downloading the picked-up items) either the terminal vehicle could return to the head office or it could go to a point of service in which a mobile warehouse is waiting for it.
- Third, after regenerating its capacity, we can continue to construct routes in the same way until the maximum allowed hours for terminal vehicles are reached. After that, it is necessary to use a new terminal vehicle, following this process, until all tasks are completed.

In this process, the pheromone matrix is updated every iteration following equation (2).

$$\tau_{ij}(t+1) = \begin{cases} \rho\tau_{ij}(t) & \text{if the link } (i,j) \text{ not belongs to } BS \\ \rho\tau_{ij}(t) + \Delta\tau(t) & \text{if the link } (i,j) \text{ belongs to } BS \end{cases} \quad (2)$$

where $0 \leq \rho \leq 1$ is the pheromone trail evaporation rate, $\Delta\tau(t)$ is the amount of the added information in the iteration t and BS is the best route obtained.

$\Delta\tau(t)$ is defined as $\Delta\tau(t) = \sum_{nlinks} \frac{1}{A}$ where A is the value obtained when evaluating the best route by the objective function that we are using and $nlinks$ is the number of links included in the best solution.

The construction process of the routes uses the pheromone matrix and a simple heuristic, which is the inverse of the distance to be covered. Thus the ant decides which is the next node to visit based on the assessment of the nodes that can be reached considering its constraints of schedule and capacity (transition rule). It should be noted that one of these limitations is that the terminal vehicle has to return to the head office before ending its workday. This transition rule for the k-th ant is shown in equation (3).

$$\vartheta_{ij}(k) = \eta_{ij}^{\beta} \cdot \tau_{ij}^{\alpha} \qquad \forall\, i,j \in T \cup M \tag{3}$$

where i represents the location of the mobile vehicle, j is the point of its neighborhood where mobile vehicle would go, α and β are parameters of the algorithm, and heuristic value is $\eta_{ij} = \frac{1}{c_{ij}}$ and represents heuristic information not provided by the ants.

Given these evaluations of the transition rule, every possible movement receives a selection probability, and finally the movement is selected from a stochastic mode.

These probabilities of selection are calculated from equation (3) and are shown in equation (4).

$$p_{ij}(an) = \begin{cases} \dfrac{\eta_{ij}^{\beta} \cdot \tau_{ij}^{\alpha}}{\sum_h \eta_{ij}^{\beta} \cdot \tau_{ij}^{\alpha}} \rho \tau_{ij}(t) & \text{for the } h \text{ possible movements of the } an-\text{th ant} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Having exhausted the capacity of the terminal vehicle, this must be regenerated going to either a service point or head office. This decision is made in a similar way to the decision of the next point of delivery/pick-up. Firstly, the service points which both terminal vehicle and mobile warehouse may reach are calculated. Additionally, the head office is added, which is always available. Finally, each of the points is assessed using the equation (3) and based on its assessment receives a probability using the equation (4). Using these probabilities the service point is chosen in a stochastic way. There is one exception to this process, which happens when a mobile warehouse is parked at a service point near the terminal vehicle and the latter is able to reach its position before it leaves. In this case the service point is automatically selected to reduce costs.

To calculate what service points are available for the terminal vehicle we perform the following process. First service points that are closer than the head office are checked. Of these points, we analyze if it is possible for some of the mobile warehouse already in circulation to reach the point when needed. Otherwise, if there are available mobile warehouses which have not yet begun their workday, they may be used. Once the service points have been decided (where terminal vehicles have to go), the task of supplying a terminal vehicle is assigned to the mobile warehouse which is closer and is able to get there in time. However, if no mobile warehouse is available at that time, then the terminal vehicle has to return to the head office.

As we can see, the construction of the routes of the mobile warehouses depends on the terminal vehicle routes. In order to construct them we use a time table, which stores the location of each mobile warehouse at each moment. At this point, we could include, instead of a time table, a window with time constraints as it is specified in works as (Figliozzi, 2009) and (Bent & Van Hentenryck, 2006), but the main reason why we have not chosen this solutions is because we need to find a feasible solution in the shortest possible time. Therefore, we have chosen to have a time grid so we can obtain the solution in the shortest time.

When a terminal vehicle finishes its workday, it has to return to the head office and, if there are terminal vehicles available, a new one will begin to perform tasks until it exhausts its workday. This process is repeated iteratively until all tasks have been completed or all terminal vehicles have been used.

A notable aspect regarding the terminal vehicle management is that although the solution is constructed sequentially, in the case of requiring more than one terminal vehicle, all start their workday at the same time, but it is possible that one of them finishes delivery/pick-up items before the end of its workday, because there are no more items for that day.

Evaluation of the solutions

The solution is provided as a sequence of points to which vehicles must go. Each terminal vehicle is assigned a route and each mobile warehouse is also assigned a schedule indicating where it must be found in certain moments.

The cost of the solution provided by the metaheuristic is given by the cost of all terminal vehicle routing plans and the cost of all mobile warehouses routing plans. To minimize the number of terminal vehicles and mobile warehouses, when the cost of each solution is calculated, it is penalized in function of the number of terminal vehicles and mobile warehouses used in the solution at a rate of $\frac{|\sigma|}{|T|}$ and $\frac{|\varphi|}{|M|}$. So, the smaller the number of terminal vehicles and mobile warehouses used in the solution, the lower the cost. Therefore, the four objective functions (1.1), (1.2), (1.3) and (1.4) are replaced by the objective function (5).

$$\min\left[\left(\sum_{r_t^i\in\sigma} c_t\left(r_t^i\right)\cdot\sum_{r_w^i\in\varphi} c_w\left(r_w^i\right)\right)\cdot\left(\frac{|\sigma|}{|T|}\cdot\frac{|\varphi|}{|M|}\right)\right] \qquad (5)$$

But, it is possible that for certain input parameters such as the number of terminal vehicles, the number of mobile warehouses and items to deliver and/or pick up, there will not be a feasible solution. Then the following solutions are adopted:

- Discard items remaining to deliver/pick up when no more terminal vehicles available. In this case we shall take into account the priority of the items. For this, the objective (5) must be modified to include these priorities.

  Each item contains a value indicating the priority of delivery or pick-up. This value will range from 1 to 10 where 1 indicates lowest priority and 10 the highest. Value pri_i denote the priority of item i, $\forall\, i \in Customers$. The objective function that reflects these priorities is as follows $\min\left(10\cdot|Customers| - \sum_{\substack{i\in Customers\\ i\in Sol}} pri_i\right)$ where $Sol = \bigcup_{i=1}^{u} cust\left(r_t^i\right)$.

  In this situation, the new objective function is added in (5) as a new factor weighted by K positive constant indicating the weight of this component.

- Increasing the workday of the last terminal vehicle in order to complete the work.

Decision making when there isn't a feasible solution, is carried out by the company, which must determine the most favorable decision based on various input parameters. Because the computation time of metaheuristic is not very high, the company will evaluate the different solutions running the metaheuristic as many times as it is necessary.

Resource requirements and scalability

Once we have selected the metaheuristic, which we are going to use, we can make an estimate of resources required for its application. One of the main limiting factors is the size of the distance matrix that links the different delivery, service and pick-up points, in particular the size of the distance matrix can be estimated. The memory used by today's personal computers, may be between 4 and 8 GB, assuming you could only use 2 GB, a distance matrix of 20000 points can be handled without problems, which could be considered a number of really significant deliveries and pick-ups for a single company of distribution.

Regarding the time necessary to get the solutions, as it is a metaheuristic that is always able to provide an approximate solution (although not the best), it will depend on how much time we want to spend to improve the initial solutions. We expect that the time necessary is in the order of minutes. Tests carried out using the prototype designed show a reasonable time, from a few seconds for small instances (100 points) to about 5 minutes for instances of larger size (1000 points).

If constraints of time were too tight, the ant colony could be parallelized easily, since each ant can build its solution in parallel during the same iteration, which will help to reduce the execution time in an important factor. Therefore, we believe that initially the needs for instances of a significant magnitude (20000 points of delivery, pick-up and service) could be fulfilled with a personal computer of the last generation. In the case that the magnitude of the instances is increased or time constraints could be overflowed, we could choose to configure a cluster composed of several personal computers.

APPLICATION EXAMPLES

The proposal is tested in a network of a simulated urban city. We identify each point, which may represent a delivery, pick-up, or service point, by a number. There is a limited number of vehicles and each vehicle has a limited capacity and speed. The proposed solution obtains the minimum route for deliver/pick up items. This solution uses the minimum number vehicles required to give sufficient coverage.

We show two test cases. The first shows how to find a solution for a problem of delivery and pick-up of items with different priorities, having a maximum number of vehicles. The second shows how the solution is adapted running the previous case when we introduce new delivery and pick-up items. Note that the procedure penalizes solutions which contain lower priority items. For these test cases, the algorithm MM-AS runs with the parameters shown in Table 1.

Table 1: Parameters to run MM-AS

ρ	τ_{min}	τ_{max}	α	β	**# ants**	**iterations**
0.9	1	5	1	2	10	50

Case 1

We are going to suppose we have a graph with the structure of the streets of a city and every possible points of delivery, pick-up and service are described in this structure: each point is denoted by a number. Table 2 shows the input data to configure the problem.

Table 2: Configuration file of Model

File of Distances distance.file=Murcia.map
File of Points(services,deliveries,pick-ups) point.file=pointsMurcia.txt
Messenger Features messenger.speed=8 messenger.capacity=5 messenger.hours=8 messenger.quantity=10
Truck features truck.speed=40 truck.capacity=1000 truck.hours=8 truck.quantity=5

Table 3: Configuration of points of Delivery, Pick-up, Service and Head Office

Pick-ups	21 1	51 1	30 2	36 1	27 1	58 1	80 1	81 2	33 8	13 5	26 4	73 2	76 2	45 2	100 1	17 1	3 1	16 2	54 7	71 1	25 1	94 1	90 3	39 3	78 1	19 1	14 9	15 4	86 1	11 1
Deliveries	41 1	4 1	50 3	75 1	32 4	98 1	52 1	22 2	35 3	60 1	87 2	67 6	89 9	66 2	68 1	34 7	47 9	12 7	99 4	69 3	61 2	93 1	48 1	43 1	7 6	2 1	20 3	49 2	92 1	24 2
Pick-ups/ Deliveries	57/55 9	96/28 1																												
Service points	88	97	72	79	59	40	77	64	82	53	84	95	31	38	6	85	63	18	46	23	29	44	37	9	65	42	62	10	56	5
Head office	1																													

Both the graph and the points of delivery and pick-up of the problem are defined in Table 2, where the speed of vehicles, their capacities, their working time and the maximum available amount of each them are described too. Table 3 shows the delivery and pick-up points, along with service points within the structure. Head office point is located in 1. As we can see in the table, besides the corresponding delivery and pick-up points, under each of these points, a value indicating the priority of the item appears. Table 4 shows the obtained solution. In this table we use the following nomenclature:

- TV_x: Terminal vehicle x
- MW_x: Mobile warehouse x
- *WTPS*: Waiting time of MW_x at the point of service.
- TM_x: Time of meeting with mobile warehouse or at the head office.
- e_x: Item to pick up at point x
- r_x: Item to deliver at the point x
- s_x: The terminal vehicle is at a service point x
- rr_x: Item to be picked up at the point x and delivered on the same route.
- ee_x: Item picked up over the route of the terminal vehicle to be delivered at the point x
- c_x: Head Office of the company

Table 4: Feasible solution for case 1

TV_1 **TM_x**	c_1	r_2	r_{12}	e_{11}	e_{13}	e_3	r_4	r_{20}	r_7	s_6 2:37	e_{16}	e_{17}	e_{27}	e_{26}	r_{24}	r_{22}	r_{32}	e_{33}	r_{43}	r_{34}	c_1 4:45	e_{14}	e_{15}	e_{25}	r_{35}	e_{36}	e_{39}	r_{48}	r_{47}	r_{41}	c_1 7:55
TV_2 **TM_x**	c_1	e_{21}	e_{19}	e_{30}	r_{50}	r_{60}	r_{49}	e_{58}	r_{68}	r_{69}	s_{59} 3:00	e_{45}	e_{51}	r_{61}	e_{71}	e_{73}	e_{54}	r_{52}	rr_{57}	r_{67}	r_{66}	ee_{55}	c_1 7:37								
TV_3 **TM_x**	c_1	r_{75}	e_{76}	e_{86}	e_{94}	r_{93}	r_{92}	e_{81}	r_{87}	r_{89}	s_{79} 3:45	e_{80}	e_{90}	e_{78}	r_{98}	r_{99}	e_{100}	rr_{96}	ee_{28}	c_1 7:37											

MW_1 **WTPS**	c_1 0:00-0:00	s_6 2:26-2:41	s_{59} 2:55-3:10	s_{79} 3:41-3:56	c_1 4:25-4:25

We briefly analyze the route of TV_1, which is divided in 3 subroutes. After leaving the head office, TV_1 can deliver three items and pick up five. Once it has exhausted its capacity it regenerates it at service point 16 where it meets MW_1. On the other hand, after finishing its second subroute, it regenerates its capacity at the head office. We can also outline that MW_1 can also support TV_2 and TV_3.

Case 2

We are going to suppose we introduce in the chain of delivery/pick-up an urgent item. It has to be picked up at point 91 and has to be delivered at the point 74. At this moment, the algorithm is executed once again, but we discard all the tasks that have already been treated. Consequently the algorithm only takes into account the remaining tasks and the new one. The proposed solution is shown in Table 5.

Table 5: Feasible solution for case 2

TV_1	r_{34}	s_{44}	r_{52}	r_{48}	r_{35}	r_{47}	r_{41}	c_1	
TM_x	4:15	4:23							
TV_2	e_{51}	r_{61}	e_{71}	rr_{91}	r_{98}	e_{73}	ee_{74}	e_{54}	c_1
TM_x	4:15								
TV_3	e_{78}	r_{67}	r_{66}	e_{99}	e_{100}	rr_{96}	ee_{28}	c_1	
TM_x	4:15								

MW_1	s_{44}	c_1
WTPS	4:15 - 4:30	4:40 - 4:40

In this situation, TV_1 is supported by MW_1 again in point 44, where it can pick up or deliver some items. As we can see in the readjustment of plans, tasks 14, 15, 25, 36 and 39 are discarded.

CONCLUSIONS

This paper presents the study, analysis and proposed solution for the problem of finding routes for delivery and/or pick-up of items using environmentally friendly vehicles. In the problem the cost of the solution is minimized, along with the number of resources used: the number of mobile warehouses and terminal vehicles.

This study has been performed to show the feasibility of this approach, so that a company, which tries to open a new market from the ecological viewpoint and wants to provide a more customized attention for the end user, can use it. We propose to address this problem using the ant colony metaheuristic, which provides a flexible and scalable system that can fulfill the needs of the company of the future. Tests show that the proposed solution is feasible since it is possible to find good solutions in a few minutes for a medium size problem.

ACKNOWLEDGMENTS

Supported by the project TIN2011-27696-C02-02 of the Ministry of Economy and Competitiveness of Spain. Thanks also to ‘‘Fundación Séneca – Agencia de Ciencia y Tecnología de la Región de Murcia” (Spain) for the Funding Program for Research Groups of Excellence (04552/GERM/06) and the support given to R. Martínez by FPI scholarship program.

REFERENCES

Bella, J.E. & McMullen, P.R. (2004). Ant colony optimization techniques for the vehicle routing problem. Advanced Engineering Informatics, 18, 41—48.

Bent, R. & Van Hentenryck, P. (2006). A two-stage hybrid algorithm for pickup and delivery vehicle routing problems with time windows. Computers and Operations Research 33, 875—893.

Cadenas, J.M. & Garrido, M.C. & Muñoz. E. (2008). Using machine learning in a cooperative hybrid parallel strategy of metaheuristics. Information sciences: an international journal 179, 3255—3267.

Crainic, T.G. & Ricciardi, N. & Storchi, G. (2009). Models for evaluating and planning City Logistics systems. Transportation Science 43(4), 432—454.

Donati, A.V. & Montemanni, R. & Casagrande, N. & Rizzoli, A.E. & Gambardella, L.M. (2008). Time dependent vehicle routing problem with a multi ant colony system. *European Journal of Operational Research 185 (3), 1174—1191*

Dorigo, M. & Maniezzo, V. & Colorni, A. (1996). Ant System: Optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics - Part B, 26(1), 29—41.

Dorigo, M., & Stützle, T. (2003). Ant Colony Optimization Metaheuristic. In F. Glover & G.A. Kochenberger (Eds.), *Handbook of Metaheuristics* (pp. 251–286). London: Kluwer Academic Publisher.

Figliozzi, M.A. (2009). Planning approximations to the average length of vehicle routing problems with time window constraints. *Transportation Research B 43, 438—447.*

Fuellerera, G. & Doernera, K.F. & Hartla, R.F. & Iorib, M. (2009). Ant colony optimization for the two-dimensional loading vehicle routing problem. *Computers and Operations Research, 36, 655—673.*

Gajpal, Y. & Abad, P.L. (2009). Multi-ant colony system (MACS) for a vehicle routing problem with backhauls. *European J. of Operational Research, 196, 102—117.*

Kenworthy, J.R. (2006). The eco-city: ten key transport and planning dimensions for sustainable city development. *Environment and Urbanization 18(1), 67—85.*

Montemanni, R. & Gambardella, L.M. & Rizzoli, A.E. & Donati, A.V. (2005). Ant Colony System for a Dynamic Vehicle Routing Problem. *Journal of Combinatorial Optimization, 10, 327—343.*

Nuortioa, T. & Kytöjokib, J. & Niskaa, H. & Bräysyb, O. (2006). Improved route planning and scheduling of waste collection and transport. *Expert Systems with Applications 30, 223—232.*

Puris, A. & Bello, R. & Martinez, Y. & Nowe, A. (2007). Twostage ant colony optimization for solving the traveling salesman problem. In J. Mira & J.R. Álvarez (Eds.), *Nature Inspired Problem-Solving Methods in Knowledge Engineering* (pp. 307—316). Berlin / Heidelberg: Springer.

Rizzoli, A.E. & Montemanni, R. & Lucibello, E. & Gambardella, L.M. (2007). Ant Colony Optimisation for real world vehicle routing problems: from theory to applications. *Swarm Intelligence 1(2), 135—151.*

Silva, C.A. & Sousa, J.M.C. & Runkler, T.A. (2008). Rescheduling and optimization of logistic processes using GA and ACO. *Engineering Applications of Artificial Intelligence 21 (3), 343—352*

Tzeng, G-H. & Cheng, H-J. & Huang, T.D. (2007). Multi-objective optimal planning for designing relief delivery systems. *Transportation Research Part E 43, 673—686.*

TERMS AND DEFINITIONS

Logistics problem: Supply chain management and of the flow of resources between points of origin and points of destination in order to meet some requirements for the improvement of company efficiency (of customers and corporations).

Dynamic problem: When a problem, defined in terms of the input data, allows them to change over time.

Ecological logistics problem: Supply chain management and of the flow of resources between points of origin/destination in order to meet some requirements for the improvement of company efficiency (of customers and corporations) that reduce the environmental and energy footprint.

Ecological vehicle: Vehicle (with or without motor) that produces less damage to the environment that the conventional combustion engine.

Metaheuristic: A high-level strategy for solving a very general class of computational problems by combining user given black-box procedures — usually heuristics — in a hopefully efficient way

Ant Colony: Approach to problem solving that takes inspiration from the foraging behaviour of some ant species. These ants deposit pheromone on the ground in order to mark the best routes that should be followed by other members of the colony.

Pheromone: Number indicating how suitable a route is in the search process of an algorithm. This amount is deposited by local agents and it is decreasing as time passes.