DEMAND AND ROUTING MODELS: AN APPROACH FOR SIMULATING THE URBAN GOODS MOVEMENTS

Antonio Polimeni, Antonino Vitetta
Mediterranea University of Reggio Calabria, Italy
Department of Computer Science, Mathematics, Electronics and Transportation
antonio.polimeni@unirc.it, vitetta@unirc.it

ABSTRACT

This paper presents a method to analyse goods movement simulation in urban/metropolitan areas. Urban goods movements involves two components: demand in terms of urban goods movements and vehicle routing with constraints (time windows, number of vehicles, …).
To analyse demand we consider a multi-step model, while to analyse goods movements a Vehicle Routing Problem with Time Windows (VRPTW) is formalized.
In the demand analysis, goods movements are evaluate considering a macro-levels approach, which this approach goods movements are analysed from upper macro-levels (commodity and vehicle level) to the path choice model. At the last macro-level of demand analysis is considered a path model, and the one-to-one and the one-to-many approaches are evaluate. In details, the one-to-one approach allows to evaluate the path cost and the path probability; the one-to-many approach allows to optimize the vehicle routes. In the one-to-many approach, the optimization is performed applying a genetic algorithm for which some procedures are proposed for the solution evolution. A real case application is designed to detect the paths and the stops of some vehicles (2-6 tons) to deliver dairy products at retailers in a city. The observed paths are optimized using the proposed procedure.

Keywords: City Logistics, goods movement, vehicle routing problem, genetic algorithm.

INTRODUCTION

This paper presents a method to simulate goods movements in urban/metropolitan areas. Two components are involved: demand in terms of urban goods movements and vehicle routing with constraints (time windows, fleet size, load factor …). For demand we propose a commodity-based model, which simulates the goods quantity to be delivered to a client (i.e. a retailer); for the Vehicle Routing Problem (VRP), a VRP with Time Windows (VRPTW) to simulate goods movements is developed.

The aim of this work is the urban goods demand analysis considering the goods demand, the goods distribution, the paths. The demand models are oriented to forecast the goods quantity purchased according to some attributes (i.e. shop characteristics, end-consumer behaviour and so on). The goods distribution refers to the goods movements in an urban area to define the best strategies to optimize the supplying operations. The strategies can regard supplying frequency, supplying quantity, the retailers (shops) supplied in function of their localization, the order in which the supplied retailers (shops) are visited. This work can be important for both policy and for business: in the first case the methodology can be used to regulate the goods vehicle access in the urban area; in the second case the methodology is a means for business users to manage their fleet. Another aspect of the problem are some commercial issue imposed by the clients that need to re-design the tour in real time (for example, goods to be delivered only at certain times of the day and communicate by client during the tour). The distribution strategy should take into account this aspect, in order to optimize the travel time (or distance). An opportune real time system for fleet management needs and reduce the travel time and the operative costs.

Various models and methods may be used to analyse urban goods demand: the main classifications concern the output considered and the structure. As regards the latter, some models have a structure similar to that used for passengers (multi-step models), while others are based on the macro-economic approach (spatial price equilibrium models) (Harker, 1985). In terms of output, while some models estimate the commodity quantities transported (Ogden, 1992; Holguìn-Veras, 2002), others estimate the vehicle number involved in goods transport in urban areas (Ogden, 1992; Holguìn-Veras and Thorson, 2003). With vehicle-based models, interactions with other traffic components can be evaluated and paths (and/or routes) followed can be analyzed; considering that the vehicle is given the optimal route has to be found. Hunt and Stefan (2007) proposed a microsimulation model to simulate the number of routes between zones in the study area split by time period and trip purpose. As regards route analysis (without taking the order of visits into account) of goods vehicles in urban areas, attempts have been made to correlate route length with the number of stops (Veras and Patil, 2005) or study the number and length of empty trips (Figliozzi e*t al.* 2006). An extended review concerning demand models is reported in Russo and Comi (2007).

The VRP consists in designing optimal routes from a depot to a set of customers, subject to various constraints (time windows, vehicle capacity, route length,…) (Laporte, 2007). It was introduced by Dantzig and Ramser (1959) to optimize the movements of a fleet of gasoline delivery trucks. Hence, various specifications are proposed for the VRP: the DVRP (Dynamic VRP) has been advocated (Montemanni *et al.*, 2005; Hanshar and Ombuki-Berman, 2007) given that the number of customers is not known a priori, but is an input that is variable in time. Others have proposed the VRP with Time Windows (VRPTW), in which deliveries can be made within a set time interval. Time window constraints can be rigid (Hu *et al.*, 2007) or soft (Ando and Taniguchi, 2006; Figliozzi, 2009). Further specifications have been suggested: i.e. the Time-Dependent VRP (TDVRP) in which travel time is a function of the travel day (Donati *et al.*, 2008) or a modification of VRP in the Inventory Routing Problem (IRP) (Campbell and Savelsbergh, 2004).

The vehicle routing problem can be solved using exact (Fisher, 1994; Fisher et al. 1997, Kohl and Madsen, 1999; Toth and Vigo, 2002; Chabrier, 2006; Baldacci et al., 2008; Qureshi et al., 2009; Azi et al., 2010) or heuristic algorithms (Badeau et al., 1997; Laporte et al., 2000;

Jones et al., 2002; Montemanni et al., 2005; Hanshar and Ombuki-Berman, 2007; Laporte, 2007; Vitetta et al., 2008; Bin et al., 2009; Repoussis and Tarantilis, 2009; Zachariadis and Kiranoudis, 2010). Exact approaches have limitations related to calculation times and the limited size of the problems that can be solved. An extended review concerning the VRP, several variants and solution approaches is reported in Laporte (2007) and Gendreau et al. (2008). Furthermore, in recent years many commercial tools have been developed with a view to solving routing problems, based on both exact algorithms and heuristic algorithms. In Vitetta et al. (2009) a list of the main tools and their characteristics is reported.

In this paper we formulate a problem that can be decomposed into two subproblems: a demand analysis in which goods movements are considered in terms of quantity and a VRPTW in which routes are optimized; the costs involved in the problem are defined by using network theory.

The problem is solved by means of a bio-inspired approach based on the use of a genetic algorithm. In the proposed algorithm, some new crossovers are proposed to recombine the genetic material: two random crossovers which can be used simultaneously and a crossover that allows us to insert an element in a position that minimizes cost. Performance of the algorithms is compared by using crossover operators and their combinations.

The paper is structured as follows. In second section we present the formulation of the models, while in third section the proposed genetic algorithm is reported. In fourth section two applications are reported: a test to evaluate the proposed procedure and a real case application. Finally, some conclusions are drawn and future developments are outlined.

THE PROBLEM

The general macro-architecture of reference is that stated in the literature (Russo *et al.*, 2007). In the proposed macro-architecture, goods movements are analysed from upper macro-levels (commodity and vehicle level) to the path choice model (figure 1).

At the first macro-level the goods quantity purchased in a zone by a retailer can be analysed on two levels:

commodity level, consisting of two macro-models:

- attraction macro-model; this refers to end-consumer quantities;
- acquisition macro-model; this concerns logistics trips from the retailer's outlet

vehicle level, which consists of two macro-models:

- service macro-model;
- path macro-model.

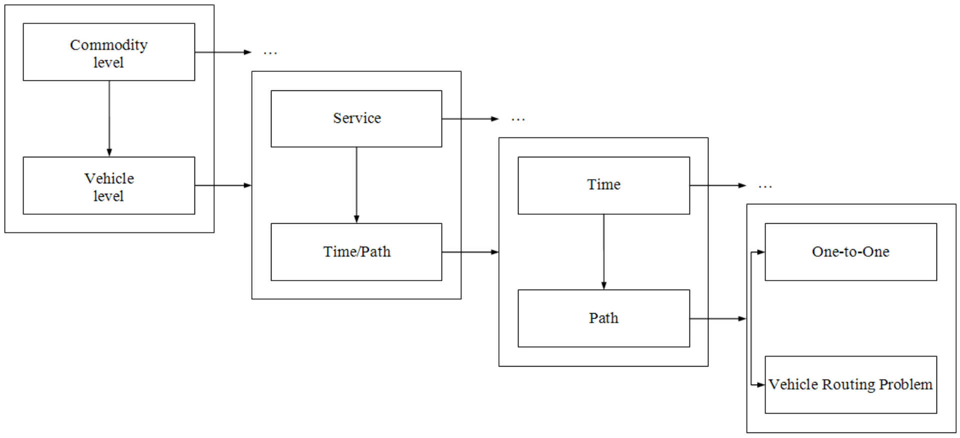


Figure 1- Macro-architecture of goods movements

Demand analysis

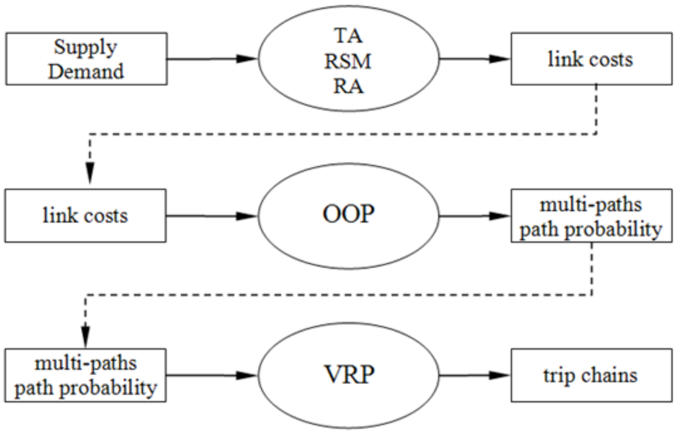
The paper analyzes the movements for which the decision-maker can be considered a carrier who uses his/her own vehicle fleet to deliver goods. In this case, a retailer who needs to be supplied depends on the decisions of the carrier as to how and when to undertake delivery. Demand is simulated through two models: the first simulates the goods attraction to the retailer, the second the number of trips (or their frequency) to deliver the required goods.
The first model simulates the goods quantity attracted by an outlet, in a fixed time period T, as a function of an attributes vector **X** that reflects the outlet characteristics (size, number of employees, and so on):
$q_i = q_i(\beta, \mathbf{X}_i, T)$
where *i* is the outlet, β a vector of parameters to calibrate and $\mathbf{X}_i$ the attribute of outlet *i*.
The second model simulates the supply trips to deliver the goods provided by the previous model. This model can be deterministic or probabilistic: the former case has a probability of 1 that the goods are delivered with a fixed number of trips; in the latter, the probability is less than 1. With this model, the total quantity to be delivered is spread across a fixed number of time subintervals, each with its own probability. This model derives from the input for vehicle routing.

Vehicle routing

To identify the optimal paths in the path macro-model, three steps can be followed:
- system performance estimation, which can be achieved through one, or a combination, of the following: traffic assignment (TA) (static or dynamic), real-time system monitoring (RSM) or reverse assignment (RA) (Russo and Vitetta, 2005);
- one-to-one problem (OOP) solution, which consists, given the costs on the network obtained by the previous step, in generating alternative paths for each origin-destination pair;
- many-to-one problem solution, which consists, given the optimal paths between every origin-destination pair obtained by solving the OOP, in the solution of a VRP formulated as

a classic optimization problem whose objective is to calculate the best combination of one-to-one paths in order to visit a certain number of network nodes in succession.

In figure 2 we report the steps to simulate goods movements.



TA: Traffic Assignment; RSM: Real-time System Monitoring; RA: Reverse Assignment; MOOP: Multi-path One-to-One Problem; MVRP: Multi-path Vehicle Routing Problem

Figure 2- Simulating goods movements

System performance estimation

In this section, simulation of the transport system is analysed to evaluate its performance. The input consists in demand and supply; the output comprises the link flows and costs. To calculate the link flows and costs, three methods could be used: TA, RSM and/or RA.

The TA method simulates the demand-supply interaction to determine system performance (flows and costs). Two approaches are possible: User Equilibrium (UE) or Dynamic Process (DP) (Wardrop, 1952; Beckman *et al.*, 1956; Sheffi, 1985; Ben Akiva *et al.*, 1998; Cascetta, 2001; Russo and Vitetta, 2003; Cantarella *et al.*, 2006; Russo and Vitetta, 2006). The TA input comprises:

- a supply model simulating network characteristics;
- a demand model simulating user behaviour;

which give as output:

- link flows;
- link costs.

RSM allows us to obtain the traffic flow data using monitoring techniques and can be obtained with:

- measurement at fixed points in the network with traditional measurement systems like loop detectors and image processing (Hoose, 1991);
- floating cars (Torday and Dumont, 2004) in the network (individual cars, taxis, transit system vehicles);

RSM costs and flows are usually made for a subset of the network links.

RA models (Russo and Vitetta, 2005) have the following input:

- link flows;
- link performance in terms of costs;

and give as output

- the link cost parameters of the cost-flow functions used in the supply model;
- the value (number of trips) and/or the model parameters of the demand model.

Starting from observed costs and flows (i.e. provided by RSM), RA models provide the demand value and/or parameter and/or the link cost parameters of the cost-flow functions used in the supply model.

One-to-one problem

As input, the one-to-one problem (OOP) has costs and flows and, as output, it supplies the probability of considering a path for vehicle routing, taking user behaviour into account.
In this paper, to solve the OOP, a probabilistic approach is adopted. In this approach, having established a criterion to define the cost (e.g. minimum travel time or a combination of costs), the link cost, and hence the path cost, is a random variable resulting from the user's perception of the possible paths. The probability of considering a path in vehicle routing $p_n(k)$ can be calculated as the sum, on all the sub-sets Γ_i which contain the alternative k, of the product between the probability $p_n(\Gamma_i^n)$ of perceiving the sub-set Γ_i^n and the conditional probability $p_n(k/\Gamma_i^n)$ of considering path k given the choice set Γ_i^n (Manski, 1977):

$$p^n(k) = \sum_i p^n(\Gamma_i^n)\, p_n\,(k/\Gamma_i^n)$$

For the link cost, we specify the following function:

$$t_{ij} = \left(L_{ij}/\beta_1\right) \cdot \left(1 + \left(\beta_2 + \beta_3 X_{ij}\right)^{\beta_4}\right)$$

where:
t_{ij} is the cost of link ij;
L_{ij} is the length of link ij;
X_{ij} is a binary variable that depends on link flow;
β_i are parameters to calibrate.
In this way, starting from observed link costs (times) it is possible to calibrate the cost function so as to evaluate the cost of all network links. For this purpose the links were divided into categories according to their characteristics and for each of them the cost function parameters can be calibrated.

Vehicle routing problem

In this paper the problem proposed is a classical VRPTW, applied to the case of a commercial vehicle fleet. The origin and destination is a central depot. The problem solution is a sequence of nodes (points where the clients are located) associated to each vehicle.

The following notation is used:

- d is the depot;
- **N** = {1, 2, …, n}, node set;
- **L** = {a : a = (i,j) $\forall$ i, j $\in$**N**}, link set;
- **Z** = {1, 2, …, m}, **Z** $\subset$ **N** nodes can be visited (clients);
- **V** ={1, 2, …, m}, set of goods vehicles;
- f_a generic vehicle flow on link $a \in L$;
- r_i demand on node *i*;
- **f** is the vector of link flow;
- b_j vehicle capacity, j=*1, 2, …, m*;
- c_a link *a* cost;
- $BT_{l,k}$ penalty before time at client *l* using path *k*;
- $AT_{l,k}$ penalty after time at client *l* using path *k*;
- $OT_{l,k}$ operation time at client *l* using path *k*;
- y_{iv} variable that is equal to *1* if node *i* has already been visited by vehicle *v*, *zero* otherwise;
- x_{kv} variable that is equal to *1* if path *k* is used by vehicle *v*, *zero* otherwise;

Objective function

Considering an oriented graph **G(N, L)**, the relation between links and paths is a binary matrix Δ, called the link-path incidence matrix. To each link *a* a vehicle flow, f_a, and cost, c_a, are associated. The relation between link flow f_a and path flow h_k is:

$$f_a = \sum_k \delta_{ak} \cdot h_k$$

Using vectorial notation:

$$\mathbf{f} = \Delta\,\mathbf{h}$$

The relation between path cost g_k and link cost c_a is:

$$g_k = \sum_a \delta_{ak} \bullet c_a + g_k^{NA}$$

in which g_k^{NA} are non-additive costs.

In some cases, the link cost is a function of the flow (for more details, see Cascetta, 2006):

$$\mathbf{c} = \boldsymbol{c}(\mathbf{f})$$

Combining graph definition and cost function, we define a *network* and if the cost is a function of the flow, the network is congested.

The case of a congested network is considered, and the VRPTW is expressed with an optimum problem:

$$\text{minimizing } \sum_k \left(g_k(\mathbf{f}) \cdot x_{kv}\right) \qquad (1)$$

Variables

The problem variable is x_{kv}: It is a binary variable that is equal to *one* if path *k* is used by vehicle *v*, *zero* otherwise. This formulation allows us to consider all the costs related to the route followed by a vehicle.

Constraints

The objective function (1) is subject to some typical vehicle routing constraints:

$$\sum_{v=1,\dots,m} y_{iv} = 1 \qquad \forall\, i \in \mathbf{Z},\ i \neq d \qquad (2)$$

$$\sum_{v=1,\dots,m} y_{dv} = m \qquad (3)$$

$$\sum_{i \in Z} r_i \cdot y_{iv} \leq b_v \qquad \forall\, v \in \mathbf{V} \qquad (4)$$

$$x_{kv} \in \{0,1\} \qquad \forall\, k \qquad (5)$$

$$g_k(\mathbf{f}) = \Sigma_a \delta_{ak} \cdot c_a(\mathbf{f}) + g_k^{NA} \qquad \forall\, k \qquad (6)$$

Constraint 2 indicates that only one vehicle can visit a node, constraint 3 that all vehicles go back to depot. Constraint 4 is a capacity constraint; 5 indicates that the problem variable can only take the value zero or one. In constraint 6, the term $g_k(\mathbf{f})$ is the path cost between an origin/destination pair (depot – client, client – client, client – depot). The path cost is the sum of two elements: additive costs, that depend on link and flow characteristics, and non-additive costs. The first element is obtained by solving an OOP, using a shortest path search procedure. Assuming that the travel time on the path is the path cost, a cost matrix may be defined, in which the generic element is the travel time between an origin/destination pair. The second element consists of three components: the before-time ($BT_{l,k}$), after-time ($AT_{l,k}$) and operation time ($OT_{l,k}$) for client l visited by path k. These components are calculated for each client reached by vehicle v that follows the path considered.

Before-time indicates the time penalty for advance arrival at the node. It is assumed that *before-time* is a linear function of arrival time. *After-time* indicates the time penalty for delayed arrival at the node.

Several formulations can be used to define the cost associated to *before-time* and *after-time*: the cost function can be linear, non-linear, constant, continuous or discontinuous. In figure 3 (in which the x axis indicates time as minutes after midnight) two possible cases are reported: in case a) the before-time and after-time costs are linear; in case b) the before-time cost is a linear function, and before-time is a constant value. Of course, various combinations are possible.

In this work, for the penalty, the case reported in figure 3b is assumed.

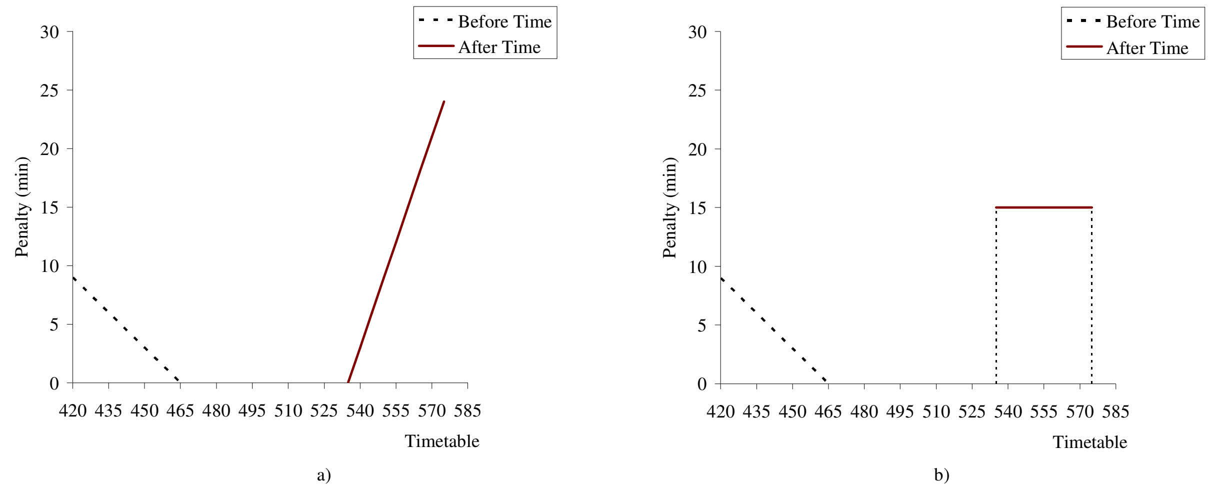


Figure 3- Non-additive path cost: penalty

Operation time indicates the time for unloading operations, it is a function of the quantity of goods delivered to client *l*:

$$OT_l = \mu \cdot q_l \tag{7}$$

in which
μ is the proportionality factor;
q_l is the quantity of goods delivered to node (client) *l*.
The non-additive path cost can then be formalized as:

$$g_k^{NA} = \sum_l (BT_{l,k} + AT_{l,k} + OT_{l,k}) \tag{8}$$

Finally, (6) is a constraint on the variables x_{kv} of the problem which can be set to 0 or 1.

ALGORITHMS FOR THE VEHICLE ROUTING PROBLEM

A special class of heuristic algorithms for the VRP is made up by bio-inspired algorithms, that is algorithms designed to simulate natural systems. Examples of bio-inspired algorithms include the Genetic Algorithm (GA) (Goldberg, 1989), Ant Colony Optimization (ACO) (Dorigo et al., 2006), and Particle Swarm Optimization (PSO) (Kennedy and Eberhart, 2001). An extended review concerning bio-inspired algorithms for VRP is reported in Potvin (2009).
The genetic algorithm is a heuristic approach that allows an optimization problem to be solved by imitating the natural evolution of a living species. Baker and Ayechew (2003) use a genetic algorithm in which selection and crossover operators are supported by a method for children placement in the new generation. In Haghani and Jung (2005) the genetic algorithm is used for optimizing the vehicle fleet routes on the basis of data collected in real time. Hanshar and Ombuki-Berman (2007) propose a crossover in which some nodes are deleted from a parent, and placed in the other in the position that minimizes the cost; while placement is not possible a new route is created. A parallel genetic algorithm (Berger and Barkaoui, 2004) can also be used, in which two populations of solutions evolve

simultaneously to optimize two different objectives; the aim of the parallel is to improve the algorithm convergence. In some cases (Prins, 2004; Alverenga et al., 2007; Marinakis and Marinaki, 2010) the genetic algorithm is used with another algorithm to improve its performance.

Specification

The solution procedure proposed is a genetic algorithm based on the concepts of natural selection and genetic theory. Indeed, the algorithm combines the survival of the fittest individuals (natural selection) with their mating (genetic theory) (Goldberg, 1989).

We define:

- trip chain as an *ordered* sequence of nodes associated to one vehicle: $\kappa_j = (d\ldots, i, \ldots d)\ \forall\, i \in \mathbf{Z}$; each trip chain has the depot *d* as the initial and final node;
- solution **S**: a set of trip chains $\mathbf{S}=\{(\kappa_1, \kappa_2,\ldots, \kappa_j,\ldots)\ \ j=1,2,\ldots, m\}$.

In general, each trip chain is associated with a vehicle: it is a case of parallel VRP. In the cases where the vehicle number is smaller than the trip chains, there are one or more vehicles that undertake more than one trip chain. The proposed formulation takes both cases into account.

Performance of the genetic algorithm depends on the operators (selection, crossover and mutation) implemented. In the literature various operators are proposed (Gwiazda, 2006); the aim is to accelerate the algorithm's convergence and find a good solution. The implemented operators are reported below.

The *selection* operator depends on the fitness value. Indeed, the selection probability is the ratio between the fitness FF_i of element *i* and the sum of fitness of all elements of the population:

$$pr_i=\left(FF_i/\Sigma_j FF_j\right) = \exp\left(-\alpha\cdot \Sigma_k\left(g_k(\mathbf{f}) \cdot x_{ki}\right)/\Sigma_j \exp\left(-\alpha \cdot \Sigma_k\left(g_k(\mathbf{f}) \cdot x_{kj}\right)\right)\right)$$

The proposed formulation of the fitness function allows the selection probability to be calculated with a Logit model. The selection operator is applied to the parent population to select the fittest parents. In the proposed algorithm, a random selection procedure is defined: the population is represented by a roulette plate; part of the roulette plate, proportional to the selection probability, is associated to each parent, and a number of random extractions (equal to population size) is made. The higher the selection probability, the higher the probability that the element is extracted. The *parents* set is the output of the selection operator and the input for the *crossover* operator.

The *crossover* operator allows us to cross two parents to obtain a new solution. In this paper, some crossovers are proposed:

- *endo-crossover*, the crossover is made *inside* the solution (between two trip chains) and between two solutions; the elements involved in the crossover are selected and crossed randomly;
- *eso-crossover*, the crossover is made between two solutions, the elements involved in the crossover are selected and crossed randomly;
- *eso-spread crossover*, the elements involved in the crossover are selected randomly and inserted in the solution optimizing the objective function.

Only one element of the population, selected randomly, is involved in the *endo-crossover*. In the selected element two trip chains to cross are selected (figure 4, a). For each trip chain a node interchange set is selected; the nodes in the interchange set are swapped for a random crossover, obtaining two new trip chains with a new sequence and a different number of nodes (figure 4, b). Moreover, given that a load is associated to each node, the total load associated to a trip chain also changes. This may violate the capacity constraint: a capacity test is thus required to ascertain whether the goods quantity is less than or equal to vehicle capacity. If the test is verified, the crossover is stopped, or else the cut points are shifted one position until the constraints are satisfied. If shifting the cut points does not allow an admissible solution to be obtained, two new trip chains are selected and the procedure is repeated. If the solution coincides with a trip chain (as is the case where we have a single vehicle) the endo-crossover degenerates into a mutation operator. The capacity test was described above for the random crossover. Finally, the solution cost is updated. The endo-crossover steps are schematized below:

STEP 1 **Select**

Select a parent from the parent set.
In the parent select two trip chains.
For each trip chain, select a node interchange set.

STEP 2 **Swap**

Swap the two node sets.
If the swap is not possible (because a constraint is broken) shrink the node interchange sets.
If shrinking is not possible, go to step 1.

STEP 3 **Update**

Update the solution cost

Depot	Depot	**Depot**	Depot	**Depot**
14	10	**7**	24	**17**
11	20	**5**	16	**12**
13	22	**23**	6	**2**
	8	**25**	9	**4**
		15	21	**19**
		18		
		3		
Depot	Depot	**Depot**	Depot	**Depot**

a)

Depot	Depot	Depot	Depot	Depot
14	10	7	24	17
11	20	5	16	12
13	22	23	6	2
	8	**4**	9	**25**
		19	21	**15**
				18
				3
Depot	Depot	Depot	Depot	Depot

b)

Figure 4 - Endo-crossover procedure

Two randomly selected solutions (parents) are involved in the *eso-crossover*, which consists in four steps. In the first step, for each parent a trip chain is selected, for each of which a node interchange set that will be crossed is selected (figure 5, a). In the second step, the interchange node sets are swapped and appended (figure 5, b). If the swap is not possible because the capacity constraint is broken, one or both node interchange sets are shrunk until the constraint is satisfied. At the third step, an admissibility test is made to search and eliminate repeated nodes (figure 5, b). Finally, the cost solution is updated. The eso-crossover steps are schematized below:

STEP 1 **Select**

Select a parent pair from the parent set.
For each parent in a pair, select a trip chain.
For each trip chain, select a node interchange set.

STEP 2 **Swap**

Swap the two node interchange sets. If the swap is not possible (because the capacity constraint is broken) shrink the node sets.

STEP 3 **Search and eliminate**

For each parent search and eliminate the repeated nodes.

STEP 4 **Update**

Update the solution cost.

Depot	Depot	Depot	Depot	Depot	Depot	Depot	Depot	Depot	Depot
20	17	2	22	18	14	10	17	24	7
16	5	7	25	**6**	**11**	20	12	16	5
23	24	14	9	**4**	**13**	22	2	6	23
12	21	3	11	**19**		8	4	9	25
8	10	13	15				19	21	15
									18
									3
Depot	Depot	Depot	Depot	Depot	Depot	Depot	Depot	Depot	Depot

a)

Depot	Depot	Depot	Depot	Depot	Depot	Depot	Depot	Depot	Depot
20	17	2	22	18	14	10	17	24	7
16	5	7	25	6	11	20	12	16	5
23	24	14	9	4	13	22	2	**6**	23
12	21	3	11	19	**6**	8	**4**	9	25
8	10	**13**	**15**	**11**	**4**		**19**	21	15
				13	**19**				18
									3
Depot	Depot	Depot	Depot	Depot	Depot	Depot	Depot	Depot	Depot

b)

repeated nodes

Figure 5 – Eso-crossover procedure

The *eso-spread crossover* involves two solution selected randomly. The procedure can be summarized in four steps.
In the first step (figure 6a), using a random procedure, a parent pair is selected in the parents set. In the same way, for each parent a trip chain is selected. Finally, the two trip chain are swapped; this procedure have, generally, two effects on the parents: a) some nodes are repeated; b) some nodes are missing. Those defects are make up in the second and in the third step.
The second step (figure 6a) consisting of a *search procedure* and a *build procedure*, applied at each parent. The search procedure is used to find the repeated nodes that are removed

from the parent. The built procedure allows to build (for each parent) a list of missing nodes that will be used in the third step.
In the third step (figure 6b) an *insertion procedure* is performed in order to insert the missing nodes in the solution. When a node is select for insertion, two phase are developed: I) find the trip chain in which the node will be inserted; II) insert the node in the best position. In the phase I) it is select a trip chain in which to insert a node respecting the problem constraints (in this case the vehicle capacity). If the insert is not possible (because the constraints are violated), an *adaptation procedure* is used: the adaptation procedure allows to modify the trip chain moving one or more nodes to another trip chain. In the phase II) all the possible positions to insert the node in the trip chain are considered: for each position is valued the trip chain cost and it is chosen the position for which the cost is the lowest (best node position). Note that a node insertion is not influenced by nodes in the list of missing nodes.
In the fourth step, finally, for each trip chain in the solutions the costs are updated.
The figure 6 depict the eso-spread crossover procedure in the case in which the adaptation procedure is not necessary. The eso-spread crossover steps are schematized below:

STEP 1 **Select and swap**

Select a parent pair from the parent set.
For each parent in a pair, select a trip chain.
Swap the trip chains.

STEP 2 **Search, eliminate and build**

For each parent, search and eliminate the repeated nodes.
For each parent, build the list of missing nodes.

STEP 3 **Swap**

For each parent and for each node in the missing nodes list, insert in a trip chain a missing node minimizing the trip chain cost. If insertion is not possible (because a constraint is broken) adapt the trip chain until the constraint is respected.

STEP 4 **Update**

Update the solution cost

Depot	Depot	**Depot**	Depot	Depot	Depot	Depot	Depot	Depot	**Depot**
20	17	**7**	22	**18**	**14**	10	17	24	**2**
16	**5**	**5**	**25**	6	11	20	12	16	**7**
23	24	**23**	9	4	**13**	22	**2**	6	**14**
12	21	**25**	11	19		8	4	9	**3**
8	10	**15**	**15**				19	21	**13**
		18							
		3							
Depot	Depot	**Depot**	Depot	Depot	Depot	Depot	Depot	Depot	**Depot**

a)

Depot	Depot	Depot	Depot	Depot	Depot	Depot	Depot	Depot	Depot
13	**2**	7	22	**14**	**18**	10	**5**	24	2
20	17	5	9	6	**15**	20	17	16	**7**
16	24	23	11	4	25	22	12	6	14
12	21	25		19	**23**	8	4	9	3
8	10	15			11		19	21	13
		18							
		3							
Depot	Depot	Depot	Depot	Depot	Depot	Depot	Depot	Depot	Depot

b)

repeated nodes   inserted nodes

Figure 6 - Eso-spread crossover procedure

The crossover output is the *child population*; some of the children are selected (according to the *mutation rate*) and the *mutation* operator is applied. The mutation operator is needed because the selection and crossover operator, recombining the current population, can remove helpful genetic material. The mutation operator proposed consists in reversing a trip chain. The mutation output is the *children set*, which can be the next parent population.

The problem data concern the number of clients to reach, the goods to deliver and the operation time at each stop, the time windows, fleet size, vehicle capacity. The problem variables are population size, crossover and mutation rate and the max number of iterations.

Calibration

In this section, we present an application test to calibrate the proposed procedure. The problem parameters are the following:

- 24 clients to reach;
- the fleet consists of 5 homogeneous vehicles (NV_{max}=5);
- each vehicle has a capacity of 40 units (b=40).

The problem variables are the following:

- the initial population size is 30 elements;
- crossover rate 60%;
- mutation rate 10%;
- max number of iterations 1000.

The solutions found by the genetic algorithm using the operators described above show in particular (figure 7) the results obtained using respectively the eso-crossover, the eso-crossover and endo-crossover simultaneously and the spread crossover. In all cases, the problem parameters and variables are the same. Note that the spread crossover allows us to find a good solution in a smaller iteration number than the eso-crossover and the combined eso and endo-crossovers.
Figure 8 reports the performance of the spread-crossover obtained by varying the crossover rate and fixing the other variables and parameters. We note that if the crossover ratio increases, in a few iterations a better solution can be found than that found by a smaller crossover ratio.

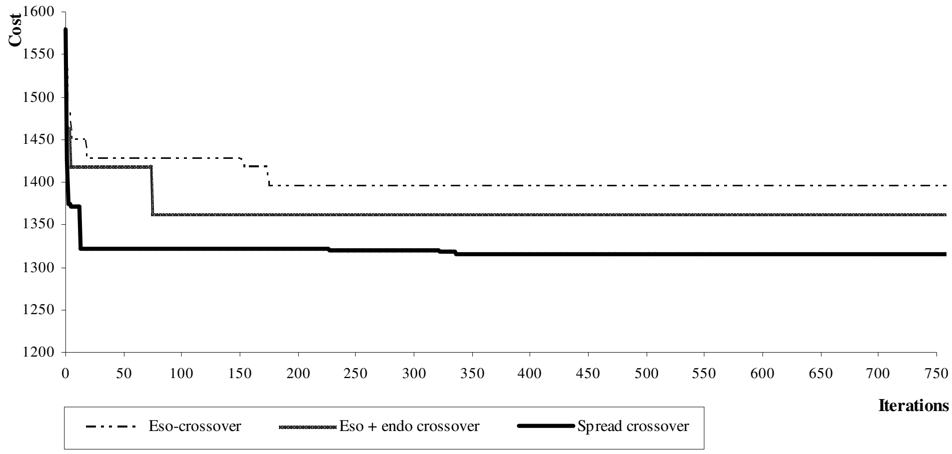


Figure 7. Problem test results varying the crossover operator

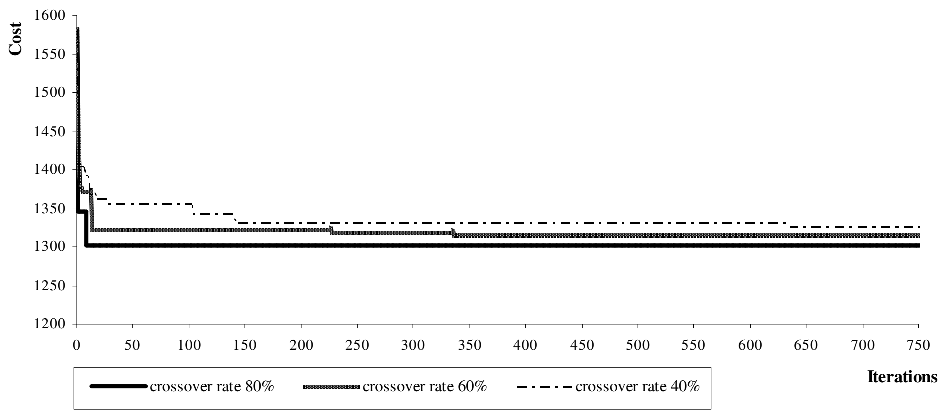


Figure 8. Problem test results using the spread crossover operator

EXPERIMENTATION FOR THE VEHICLE ROUTING PROBLEM

In this section we describe how the proposed procedures were tested. Application to a real case was carried out in Catania, a city in southern Italy. The application involved some carriers who deliver foodstuffs (milk, cheese and so on) to clients.

Real case application was designed to detect the paths and stops of some vehicles (2-6 tons) to deliver dairy products at retailers. The instrumentation consisted of a PDA equipped with GPS tracking (figure 9) and software that mapped the points (to identify the paths) at fixed time intervals (in this case 15 seconds); the stops were identified interactively by the driver. By using the PDA other data were obtained to determine characteristics of the shipment (quantity delivered, vehicle type, etc.) that were not considered in this phase.
The arrival time at the customer was indicated by the driver. The start was derived by analyzing the points that spatially coincide with the stop and assuming that the departure moment is the instant of observation of the last of these points. The stop duration (operation time) was calculated accordingly.
Three vehicles were monitored for four days, their tours starting at 7:00 a.m. and ending at 2:00 p.m. Preliminary analysis was performed considering a single vehicle which, starting from the depot, delivered to 26 clients.

Figure 9. The PDA used

The used software interfaces with the driver through two main forms. The first one allows to record the start/return at the depot (to take the paths) and to indicate the positions where the clients are located. The second one, contains the questions text provided by the analyst. The software is managed by an input file in which parameters (i.e. log interval) and questions type and text are supplied.
There are two question types: L-type questions, are open questions (that is, the driver must write the answer), M-type questions, are multiple-choice questions (that is, the answers were provided by the analyst and the driver chooses one). The questions are activated when the driver pushes a command button provided in the first form.

Costs

The link cost (and the path cost) can be simulated through a model that measures costs against variable characteristics of the problem (link length, flow, and so on). In a previous section we specified a model to simulate the travel time on a link *ij*, in table I the calibrated parameters are reported. The links were divided into four classes, according to their characteristics (the total database is constituted by 244 observations), for each class were calibrated the parameters of the proposed model.

Table I – Link cost function calibration

Link class	β_1 (km/h)	β_2	β_3	β_4	ρ^2
A	54.999	2.551E-05	0.168	0.147	0.976
B	49.953	9.057E-01	1.455	2.244	0.837
C	49.981	5.020E-01	1.942	1.752	0.887
D	31.072	5.000E-02	1.775	2.006	0.689

One to one analysis

The shortest path between every stop (client pair) was determined by considering the cost function reported in a previous section, for the link cost. To perform one-to-one analysis, a graph was developed and the paths between each client pair calculated on the basis of minimum time and minimum length criteria. A choice probability was associated to each path (Russo and Vitetta, 2006). The paths provide the input for many-to-one analysis.
We focused our attention on only one vehicle: this vehicle reached 26 clients (observed trip chain in figure 10). The shortest paths between each pair of users were analyzed in terms of the shortest time and shortest length. It was noted that in some cases the path followed by the vehicle is neither the shortest time path nor that of shortest length: if the vehicle were to use the shortest time path there would be a 4.40% reduction in travel time; were it to use the shortest length there would be a 2.50% reduction in distance travelled (table II).

Table II – One-to-one analysis: path cost

Origin	Destination	Observed Paths		Shortest Paths	
		time (min)	length (km)	time (min)	length (km)
Depot	2	17.48	12.91	17.48	12.91
2	3	0.45	0.31	0.40	0.27
3	4	0.51	0.30	0.51	0.30
4	5	2.10	0.64	2.10	0.64
5	6	0.36	0.26	0.36	0.26
6	7	5.54	4.41	5.54	4.41
7	8	0.12	0.10	0.12	0.10
8	9	3.84	2.98	3.84	2.98
9	10	0.59	0.49	0.59	0.49
10	11	1.01	0.21	1.01	0.21
11	12	1.31	0.45	0.89	0.45
12	13	0.48	0.33	0.48	0.33
13	14	2.93	2.09	2.93	2.09
14	15	0.37	0.32	0.37	0.32
15	16	0.28	0.24	0.11	0.09
16	17	1.33	0.78	1.33	0.78
17	18	2.13	1.69	2.13	1.69
18	19	2.75	2.29	2.75	2.29
19	20	4.38	2.81	4.38	2.81
20	21	1.50	0.76	1.50	0.76
21	22	1.83	1.09	1.83	1.09
22	23	9.90	5.98	9.90	5.98
23	24	1.79	1.39	1.79	1.39
24	25	0.77	0.60	0.77	0.60
25	26	3.38	1.10	2.22	1.39
26	27	6.07	3.94	6.07	3.94
27	Depot	28.43	20.67	25.80	18.86
Total		**101.63**	**69.12**	**97.20**	**67.41**

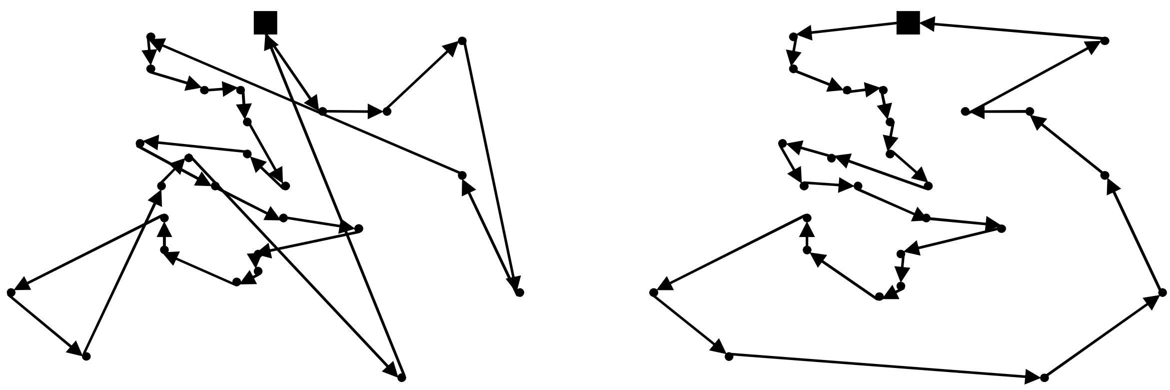
Many-to-one analysis

In figure 9 we report a trip chain followed by a monitored vehicle (9a) and the trip chain obtained by applying the genetic algorithm (9b). From a comparison of trip chain costs (Table 3) it may be noted that the genetic solution leads to a cost reduction of 2.88%.
A vehicle shuttling between the depot and customers (customer - depot - customer) would take approximately 1300 minutes (covering about 900 kilometers) along the shortest path to serve all customers. If the vehicle were to visit all users on any one route (one-to-many), it would save about 81% in time (and 81% in length).
Moreover, a trip chain design in the many to one approach allow a further reduction in the costs. This is an objective of the carriers that allow to optimize the vehicles use and their deployment. The trip chain design can also consider some constraints imposed at political level (i.e. time windows, weight limit and so on) to limit the goods vehicles transit in urban area.

Table III – Solution cost

	time* (min)	length (km)	reduction (%)
Observed	260.23	69.12	0.00
Genetic	252.74	66.24	2.88

*Travel time + operation time



a) b)

Figure 9. Solution comparison (schematic representation)

Below, we analyse and compare the observed trip chain and calculated trip chain. The analysis consists in searching for similarities within the two trip chains (figure 10). We introduce the following definitions:

- identical sub-trip chain: a node sequence that occurs in the observed trip chain and that calculated (not necessarily in the same position);
- inverted node sub-trip chain: node sequences present in the observed trip chain that can be made identical to calculated trip chain sequences by exchanging a single node;
- scattered nodes: these are isolated nodes that cannot be collected in comparable sequences.

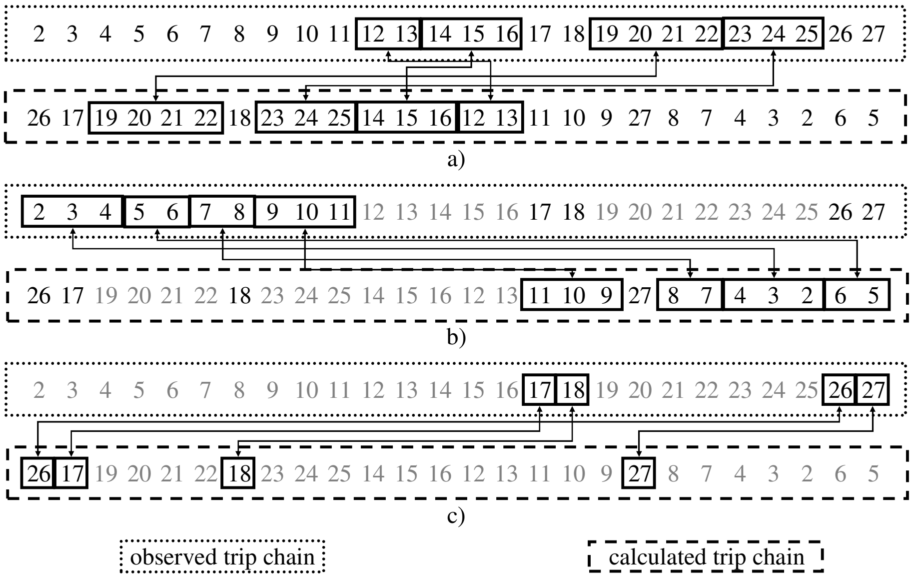


Figure 10. Similarity analysis

This comparison reveals the presence of four identical sub trip chains (figure 10a) (nodes 12/13, 14/16, 19/22, 23/25), four inverted node sub-trip chains (figure 10 b) (node sequence in the calculated trip chain: 11-10-9, 8-7, 4-3-2, 6-5) and four scattered nodes (figure 10c) (17, 18, 26 and 27). The presence of identical sub-trip chains is due to the small difference between the observed and calculated trip chain.

The proposed approach can be extended at the case in which we have *n* vehicles (n≥2). In this case was considered two routes of two distinct carriers to weight the possibility of moving some customers from one operator to another. In table IV have reported the data related to this scenario. In the observed data the first carrier reaches 24 clients, the second carrier 31 clients (then a problem with 55 clients). In the optimum routes we have the moving of a node from the first carrier to the second. The results are (optimal result compared with observed data):

1) an increment (1.81%) in the travel time for the first carrier;
2) a decrement (3.84%) in the travel time for the second carrier;
3) a decrement (1.50%) of the total travel time.

Table IV – Approach extension

	Observed		Optimized		
carrier ID	Clients number	Time* (min)	Clients number	Time* (min)	Time variation (%)
1	24	291.84	25	297.13	+1.81
2	31	414.55	30	398.64	-3.84
Total	55	706.39	55	695.77	-1.50

*Travel time + operation time

CONCLUSIONS

In this paper, a macro-architecture was reported for a model system to simulate goods movements in an urban area. In the macro-architecture proposed, the goods movements were analysed from the upper macro-levels (commodity and vehicle level) to path choice. Path choice was analysed by considering two problems: the one-to-one problem and the many-to-one (or one-to-many) problem. The many-to-one was formulated as a VRP which must be solved in the case of goods distribution. In this paper goods distribution was tackled and a VRPTW was solved to simulate the distribution process. The VRPTW was formulated by referring to some cost terms (travel time, operation time and penalty time). The solution procedure consisted of a genetic algorithm in which some crossovers were tested. Application to a real case was also proposed, based on preliminary data gathered by monitoring some vehicles for the delivery of dairy products. In this phase we reported the trip chain followed by a single vehicle and a case in which we have 2 vehicles in which are optimized simultaneously two trip chains. The route followed by the observed vehicle was compared with that optimized by the proposed procedure, which allows a reduction in both travel time and travel distance. A comparison was also made to highlight the superiority of the many-to-one over the one-to-one approach, noting that the many-to-one approach saves about 90% of time (compared to the one-to-one). The observed trip chain was compared with that found from the application of the GA proposed.

The proposed procedure can be applied in both policy and business sector: in the first case to regulate the goods vehicle access in the urban area (and consequently reducing the interaction between goods vehicles and other traffic components); in the second case is a means for business users to manage the fleet (and reducing the operating costs).

Future developments should focus on improving the database, performing new calibrations of the cost function and adopting a dynamic approach to solving the VRPTW in which observations are allocated to time slots in each of which a function will be performed for calculating the link cost and penalties.

REFERENCES

Alvarenga, G.B., G.R. Mateus and G. de Tomi (2007). A genetic and set partitioning two-phase approach for the vehicle routing problem with time windows. Computers & Operations Research 34, 1561–1584.

Ando, N. and E. Taniguchi (2006). Travel Time Reliability in Vehicle Routing and Scheduling with Time Windows. Networks and Spatial Economics 6, 293-311.

Azi, N., M. Gendreau and J.Y. Potvin (2010). An exact algorithm for a vehicle routing with time windows and multiple routes. European Journal of Operational Research, 178, pp. 755-766.

Badeau, P., F. Guertin, M. Gendreau, J.Y. Potvin and E. Taillard (1997). A parallel tabu search heuristic for the vehicle routing problem with time windows. Transportation Research Part C: Emerging Technologies, 5(2), 109-122.

Baldacci, R., N. Christofides and A. Mingozzi (2008). An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. Mathematical Programming, 115(2), 351-385.

Baker, B. M. and M.A. Ayechew (2003). A genetic algorithm for the vehicle routing problem. Computers & Operations Research, 30, 787–800.

Beckman, M., C.B. McGuire, and C.B. Winsten (1956). Studies in the economics of transportation. New Haven, Yale University Press.

Ben-Akiva, M., M. Bierlaire, H. Koutsopoulos and R. Mishalani (1998). DynaMIT: a simulation-based system for traffic prediction and guidance generation, Proceedings of TRISTAN III. San Juan 1998.

Berger, J., and M. Barkaoui (2004). A parallel hybrid genetic algorithm for the vehicle routing problem with time windows. Computer & Operations Research 31, 2037-2053.

Bin Y., Y. Zhong-Zhen and Y. Baozhen (2009) An improved ant colony optimization for vehicle routing problem. European Journal of Operational Research, 196, pp. 171-176.

Campbell, A.M. and M.W.P. Savelsbergh (2004). A Decomposition Approach for the Inventory Routing Problem. Transportation Science 38(4), 488-502.

Cantarella G. E., G. Pavone and A. Vitetta, (2006). Heuristics for urban road network design: lane layout and signal settings. European Journal of Operational Research, 175, pp. 1682–1695.

Cascetta, E. (2001) Transportation system engineering: theory and methods. Kluwer, Academic Publisher.

Cascetta, E. (2006). Modelli per i Sistemi di Trasporto. Teoria ed Applicazioni. UTET.

Chabrier, A. (2006) Vehicle routing problem with elementary shortest path based column generation. Computer & Operations research, 33, pp. 2972-2990.

Dantzig, G. B. and J. H. Ramser (1959). The Truck Dispatching Problem. Management Science 6 (1), 80-91.

Donati, A.V., R. Montemanni, N. Casagrande, A.E. Rizzoli and L.M. Gambardella (2008). Time dependent vehicle routing problem with a multi ant colony system. European Journal of Operational Research, 26, 1174-1191.

Dorigo, M., M. Birattari and T. Stützle (2006). Ant Colony Optimization: Artificial ants as a computational intelligence technique. http://code.ulb.ac.be/iridia.publications.php

Figliozzi, M. A., L. Kingdon and A. Wilkitzki (2007). Freight Distribution Tours in Congested Urban Areas: Characteristics and Implications for Carriers' Operations and Data Collection Efforts. Transportation Research Board 86th Annual Meeting.

Figliozzi, M. A. (2009). An iterative route construction and improvement algorithm for the vehicle routing problem with soft time windows. Transportation Research Part C: Emerging Technologies, **In Press, Corrected Proof**.

Fisher, M.L. (1994). Optimal solution of vehicle routing problems using minimum k-trees, Operation Research 42, 626-642.

Fisher M. L, K. O. Jörnsten and O. B. G. Madsen (1997) Vehicle routing with time windows: two optimization algorithms. Operations Research, 45 (3), pp. 488-492.

Gendreau, M., J.Y. Potvin, O. Bräysy, G. Hasle, and A. Løkketamgen (2008). Metaheuristics for the vehicle routing problem and its extensions: a categorized bibliography. In: The vehicle routing problem: latest advances and new challenges (B. Golden, S. Raghavan, E. Wasil, ed.), pp. 143-169. Springer.

Goldberg, D. E. (1989). Genetic Algorithms in Search, Optimization and Machine Learning. Addison – Wesley Publishing Company, INC.

Gwiazda, T.D. (2006). Genetic Algorithms References (vol. I and II). Gwiazda T., Poland.

Hanshar, F.T. and B.M. Ombuki-Berman (2007). Dynamic Vehicle Routing Using Genetic Algorithms. Appl. Intell. 27, 89-99.

Harker, P.T. (1985). Predicting intercity freight flows. VNU Science Press.

Holguín-Veras, J. (2002). Revealed Preference Analysis of the Commercial Vehicle Choice Process. Journal of Transportation Engineering 128(4), 336-346.

Holguín-Veras, J. and E. Thorson (2003). Practical Implications of Modeling Commercial Vehicle Empty Trips in transport logistics. Transportation Research Record 1833, 87-94.

Holguín-Veras, J. and G.R. Patil, (2005). Observed trip chain behavior of commercial vehicles. Transportation Research Record 1906, 74-80.

Hoose, N. (1991). Computer image processing in traffic engineering. John Wiley & Sons, New York.

Hu, X., M. Huang, and A.Z. Zeng (2007). An Intelligent Solution System for a Vehicle Routing in Urban Distribution. International Journal of Innovative Computing, Information and Control 3 (1), 189-198.

Hunt, J.D. and K.J. Stefan (2007). Tour-based microsimulation of urban commercial movements. Transportation Research Part B 41(9), 981-1013.

Jones, D.F., S.K. Mirrazavi and M. Tamiz (2002). Multi-objective meta-heuristics: an overview of the current state of the art. European Journal of Operational Research 137, 1-9.

Kennedy, J. and R.C. Eberhart (2001). Swarm Intelligence. Morgan Kaufmann, San Francisco.

Kohl N. and O. B. G. Madsen (1999) 2-Path Cuts for the Vehicle Routing Problem with Time Windows. Transportation Science, 1, pp. 101-116.

Laporte, G. (2007). What You Should Know about the Vehicle Routing Problem. Les Cahiers du GERAD. www.gerad.ca/en/publications/cahiers.php.

Laporte, G., M. Gendreau, J.Y. Potvin and F Semet (2000). Classical and modern heuristics for the vehicle routing problem. International Transaction in Operational Research 7, 285-300.

Manski, C. (1977). The structure of random utility models. Theory and Decision 8, 229-54.

Marinakis, Y. and M. Marinaki (2010). A hybrid – Particle swarm optimization algorithm for the vehicle routing problem. Expert System with Applications 37, 1446-1455.

Montemanni, R., L.M. Gambardella, A.E. Rizzoli and A.V. Donati (2005). Ant Colony System for a Dynamic Vehicle Routing Problem. Journal of Combinatorial Optimization 10(4) 327-343.

Ogden, K.W. (1992). Urban Goods Movement: a Guide to Policy and Planning. Ashgate Publishing Limited.

Potvin, J.Y. (2009). A review of bio-inspired algorithms for vehicle routing. In: Pereira, F.B. and Tavares, J. (eds.), Studies in Computational Intelligence: Bio-inspired Algorithms for Vehicle Routing Problem, pp. 1-34.

Prins, C. (2004) A simple and effective evolutionary algorithm for the vehicle routing problem. Computers & Operations Research 31, 1985-2002.

Qureshi A.G., E. Taniguchi and T. Yamada, (2009) An exact solution approach for vehicle routing and scheduling problems with soft time windows. Transportation Research Part E, 45, pp. 960-977.

Repoussis, P.P. and C.D. Tarantilis (2009). Solving the Fleet Size and Mix Vehicle Routing Problem with Time Windows via Adaptive Memory Programming. Transportation Research Part C: Emerging Technologies, In Press, Corrected Proof, Available online 12 September 2009.

Russo, F. and A. Comi (2007). A model system to simulate urban freight choices, Proceedings of World Conference on Transport Research, Berkeley, CA (USA).

Russo, F., A. Comi and A. Polimeni (2007). Urban freight transport and logistics: Retailer's choices. In: Innovations in City Logistics (Taniguchi, H., Thompson, G. R. Eds.), pp. 401-414. Nova Publisher, New York.

Russo, F. and A. Vitetta (2003). An assignment model with modified Logit, which obviates enumeration and overlapping problems. Transportation 33(4), 347-370.

Russo, F. and A. Vitetta (2005). Reverse assignment: Updating demand and calibrating cost jointly from traffic counts and time measurements. International Symposium on Transportation and Traffic Theory, Maryland University, July 2005.

Russo, F. and A. Vitetta (2006). La ricerca di percorsi in una rete: Algoritmi di minimo costo ed estensioni. FrancoAngeli, Milano.

Russo F. and A. Vitetta (2006). A topological method to choose optimal solutions after solving the multi-criteria Urban Road Network Design Problem. Transportation 33(4), 347-370.

Sheffi, Y. (1985). Urban transportation network. Prentice-Hall Inc., Englewood Cliffs.

Torday, A. and A.G. Dumont (2004). Probe vehicle based travel time estimation in urban networks, Proceedings of TRISTAN V. Guadaloupe.

Toth, P. and D. Vigo (2002). Models, relaxations and exact approaches for the capacitated vehicle routing problem. Discrete Applied Mathematics 123, 487-512.

Vitetta, A., A. Quattrone and A. Polimeni (2008). Safety of users in road evacuation: algorithms for path design of emergency vehicles. In: Urban transport XIV. WIT Press, pp. 727-738 (Brebbia C.A. ed.).

Vitetta, A., A. Quattrone and A. Polimeni (2009). Safety of users in road evacuation: modelling and DSS for paths design of emergency vehicles. In: Sustainable Development and Planning IV, vol. 1 (Brebbia C.A. ed.), pp. 476-485. WIT Press.

Wardrop, J. P. (1952). Some theoretical aspects of road traffic research, Proceedings of the Institute of Civil Engineers Part II (1), pp. 325-378.

Zachariadis, E. E. and C.T. Kiranoudis (2010) An open vehicle routing problem metaheuristic for examining wide solution neighborhoods. Computers & Operations Research, 37, pp. 712-723.