ABSTRACT

The paper introduces two software tools for planning and operating an urban logistic center. The former software, called DiLog, is devised to determine the optimal assignment of delivery points to several possible terminals. It allows the planner locating and sizing the terminals of the urban logistic system. Dylog is a dynamic software tool that allows the manager of an urban logistic base defining in real-time the best solution of the delivering program. The paper discusses also the related modeling framework and provides a concise description of the solution procedures, which are both based on a two-string genetic algorithm that solves the capacitated vehicle routing problem with multiple time windows. Finally, it describes the results obtained by simulating the operations at a logistic center located in the Italian town of Cosenza.

1. INTRODUCTION

Supplying of retail outlets is a process driven mainly by commercial criteria (e.g., just-in-time deliveries, warehouse outsourcing), which often disregards energy, environmental and social impacts produced by transport of goods. In urban areas, significant efficiency and environmental improvements could be obtained by introducing operational centers (called city terminals or urban logistic platforms), located in strategic points, where goods coming from outside are collected, consolidated and delivered to the city center by means of small and clean vehicles optimally charged and operated by applying ITS principles.

However, an urban distribution center introduces a further breakdown into the delivery and transport process, which could produce longer delivery times and increase distribution costs, clashing with the current trend of providing just-in-time supplies.

As a consequence, conditio sine qua non for the success of urban logistic centers is that their operations were so efficient and quick that re-combination of loads does not affect the overall quality of service required by the market.

Achieving this goal requires optimizing the functional lay-out of the logistic center and extensive application of information and communication technologies, like tracking and tracing of vehicles and parcels, individual route guidance, Electronic Data Interchange protocols, real time optimization of routing and scheduling. Since goods can arrive at the logistic center even without previous notice, it is also necessary that all operations will be coordinated and optimized in real-time in order to promptly react to unexpected events.

The paper introduces two software tools carried out jointly by ENEA, the Italian Agency for Energy, New Technologies and Environment, and the Department of Hydraulics, Transportation and Roads at the University of Rome "La Sapienza", Dipartimento Idraulica Trasporti e Strade.
Rome “La Sapienza”. The two software tools are addressed to support planners of freight urban transport and managers of logistic centers, respectively to:

- individuate the optimal lay-out of city logistics systems in a given urban context;
- support delivery operations (delivery tours scheduling and routing) even in real-time.

From a mathematical point of view, both problems are classified as NP-hard combinatorial problems and require heuristic procedures for a solution. Specifically, the operational phase is very complex, as the process continuously evolves during the day and requires taking decisions in a very short time. In fact, the operational scheme considered here assumes that parcels arrive randomly at the city terminal and have to be delivered within the daily time windows required by the final consignee.

Both software tools are based on a mathematical framework that formulates the decision process as a multi-criteria location and routing problem on a graph representing the whole road network. The solution procedures combine deterministic (Analytic Hierarchical Process, clustering techniques) and stochastic (multi-string genetic) algorithms. The software tools have been tested in simulation on two Italian towns, Terni and Cosenza, whose Administrations are taking into consideration innovative solutions for freight transport.

2. CITY LOGISTICS PLANNING

2.1 The Logistic Scheme for Urban Deliveries

The logistic scheme proposed here is based on two fundamental elements, ideally depicted in Fig.1:

- the city terminal, where system performances are improved by means of load concentration, intensive implementation of advanced information systems and cooperative operations of freight transportation;
- the drop-points, where the goods are delivered and consumers pickup their parcels.

In order to reduce new trips and parking needs as well as to minimize the discomfort for the consumer who is involved in parcels pickup, the study idea is to join the pickup and regular and/or enjoyable activity of the consumer (for example, the breakfast at the bar). Thus, commercial and public sites as bars, postal offices, newspapers shops and bookstores have been selected as possible drop-points of the delivery scheme. However, home deliveries are reserved for disabled people and for heavy goods.

2.2 The design procedure

A heuristic procedure for the design of such a logistic scheme integrates together transportation system theory and operational research techniques. The study area is split into “demand zones”, each of them is to be served by one or more drop-points. The demand of BtoC goods in each zone is estimated by applying a standard Logit model (Ben-Akiva and Lerman, 1985). An Analytic Hierarchical Process is applied to compare performances of drop-
points having equal accessibility (Fusco et al., 2003). The network link costs are computed by performing a user equilibrium assignment of passenger cars. The main inputs of the problem are socioeconomic variables of demand zones, Origin/Destination matrix of car trips, road network graph, as well as the fleet size and the set of candidate areas where locating the city terminal. The main steps of the proposed procedure are summarized in the flow-chart depicted in Figure 1.

As results, the solution procedure determines:
1. drop-points selection;
2. size and location of city transit-points;
3. size and composition of the vehicle fleet;
4. optimal routes from each transit-point to each drop-point.

2.3 Transit-point selection and sizing

Once drop-point locations, demand and vehicle types have been defined, the optimal set of logistic transit-points has to be carried out. The solution procedure searches simultaneously for both optimal facilities location and vehicle routing. In other words, it looks for the solution that minimizes the amount of investment and operating costs of both transit-points and vehicles and external costs due to pollution from vehicles. Not only delivery transport but also feeding to transit-points from external zones should be considered. A general solution is represented in Figure 2. Here, each drop-point is associated with one transit-point and one delivery tour (obviously one transit-point can be associated with more than one tour).

Such a solution is feasible if following constraints are kept:
- the total demand does not exceed vehicle capacity;
- tour time does not exceed driver shift duration;
- total daily demand of the drop-point set associated to one transit-point does not exceed the transit-point capacity.
After the drop-point/transit-point/tour associations have been defined and checked, the optimal route between one transit-point and each drop-point belonging to its set can be determined by solving the Travel Salesman Problem (TSP), considering real traffic conditions on the road network. Now all the elements (transit-point sub-set and related demand amount, distance covered, speed and travel time) exist to compute the value of the disutility function and compare the solutions against each other. As the problem is NP-hard, exact algorithms are feasible only for small/medium sized problems; as a consequence, heuristic procedures have to be used. To solve this non-linear problem we applied a double-string genetic algorithm that considers possible tours between drop-points and then searches for sub-optimal facility location and route design simultaneously. The results from the algorithm are transit-point selection and sizing, drop-point clustering, tours and number of necessary delivery vehicles, as well as the amount of all costs considered within the performance function. A concise flow chart of the genetic algorithm is shown in Figure 3.
2.4 Costs and impacts estimation

The present procedure for logistic systems design requires the computation of investment, operational and external costs in order to allow a comparison among different solutions and find-out a good one. In order to estimate the investment costs for constructing the logistic bases, unit costs (per storehouse unit area/volume) have to be fixed, according to the market values of grounds in the study area. Similarly, fleet buy-out costs and operative vehicle costs can be computed once the investment per vehicle and an average operative cost per kilometer have been fixed. Personnel costs can be estimated using a given value of the hourly wage. Environmental costs were derived from pollutant emissions estimation by means, for instance, of COPERT functions, starting from vehicle typology, distance covered and average speed. Instead of costs per unit emission, cost per unit mileage can
be used to determine the external costs of air pollution, so that emissions estimation can be avoided.

Other costs to consider are related to drop-points. Actually, drop-points do not require any construction investment as, according to the scheme proposed here, they exploit existing facilities, which currently perform functions other than parcel picking. In spite of that, a cost for the new service supplied has to be provided, in terms of whether fare per delivered parcel or monthly wage. Such a cost, as well as the other cost of the delivery service, should be put down to the logistic operators in return of reductions in operation costs thanks to last-mile cost savings.

2.5 Check of consistency and feedbacks

The solution procedure computes the location and size of transit-points, drop-point clusters and optimal delivery routes simultaneously, for a given demand pattern and an assumed vehicle fleet. Thus, once a solution has been obtained, a feedback should be carried out to verify if different hypotheses on fleet size and composition could improve the solution found. As a matter of fact, the number of vehicles and then the number of drivers greatly affects the total cost of the system.

3. DILOG: A SOFTWARE FOR PLANNING THE URBAN FREIGHT DISTRIBUTION

The software tool DiLog (Distribution Logistics) is developed in C++ and is provided with a graphical interface carried out in Delphi 7 environment. The optimization program is composed by two modules:

- a module for the drop-point/transit-point/tour association, which applies the double strung genetic algorithm described in the previous section;
- a module for the tour optimization, which applies a “greedy” algorithm to solve the well known Travel Salesman Problem (TSP) and determine so the optimal order to visit the drop-points of each tour.
The user can choose between two different versions of the genetic algorithm:
- the **simple version** for each generation creates a completely new population of individuals, with the only noticeable exception of the best individual of the population, which is kept also in the next generation.
- the **steady state version** overlaps successive populations and allows the user specifying the percentage of replacement. At each generation, the algorithm creates a temporary population of individuals, adds it to the previous one, and removes the worst individuals, keeping unchanged the size of the population.

Data input concern the total expected demand at each drop point, the daily cost of each terminal, the cost matrix for each origin-destination pair, the weights of the different elements of the objective function, the unit operating cost of vehicles, the average fixed cost, the average speed, the average time for each delivery, the average volume of a delivery, the average unloading time, the average cost for the crew, the total area of the terminal, the external costs of truck transport, the duration of service time, and the vehicle capacity.

A relevant feature of DiLog is to allow the analyst introducing a specific solution, which might be his or her current preferred solution. Such a solution will be included into the initial population and should be improved by the genetic algorithm.

The software displays the progression of the objective function during the evolution of the population. The current solution is shown in the window on the left. The program computes the total cost of daily distribution and the tours from each terminal (in figure 4, denoted by the numbers 27, 28, 29). In the example shown, which refers to small electric vehicles having capacity of...
3.14m$^3$, the first tour starts from terminal 29 and visits drop points numbered 0, 6, 10, 17, 1; the total volume of deliveries is 3.07m$^3$. A second tour is needed to complete deliveries from terminal 29. It concerns the only drop point 2, whose deliveries sum up to 0.92m$^3$. Five tours start from terminal 28. They are composed, respectively, by drop points numbered: 8, 5, 3, 4; 7, 14; 9, 16, 19; 13, 15, 22; 23, 24; 25, 26. The volume of deliveries of these tours range from a minimum of 2.09 and a maximum of 3.06m$^3$. The two tours from terminal 27 include, the former, the drop points numbered 21, 18, 11, 12 (whose total volume of deliveries equals the capacity of the vehicle) and, the latter, the drop point number 20, whose delivery volume is 0.53 m$^3$.

4. REAL-TIME MANAGEMENT OF THE OPERATIONS AT THE URBAN LOGISTIC CENTER

4.1 The dynamic capacitated vehicle routing problem with time windows

The capacitated vehicle routing problem with time windows has been shown to be a non convex NP-hard problem, for which there is no guarantee that a solution exists. Exact algorithms have been proposed to solve static problems with up to 100 destinations, so that many authors prefer heuristic approach (Desrosiers et al., 1992).

In a dynamic context, the problem is further complicated because of an additional constraint on the starting time of tours. Moreover, both travel times and the demand can change continuously during the day and this requires re-calculating the solution.

Thus, the solution procedure has to be so efficient to provide a good feasible solution for large scale problems of several hundreds or thousands of deliveries in no more than about 30 minutes. As genetic algorithms simulate an evolutionary process, they are flexible enough to be applied in real-time by continuously updating the data and adjusting the solution.

The reader can refer to Fusco and Valentini (2004) for the mathematical formulation of the dynamic capacitated vehicle routing problem with multiple time windows.

The solution procedure is based on a bi-level genetic algorithm, whose chromosomes are defined by following a path representation (Choi et al., 2003) and provide, at a given time instant, the delivery list of each vehicle. The population is updated every time the data change because a tour starts or a delivery arrives. The structure of the general solution procedure (Fig.2) takes planned deliveries as initial data, applies a clustering algorithm to generate the first population and then runs the optimization algorithm to compute sub-optimal delivery tours.

©Association for European Transport and contributors 2006
4.2 Initializing algorithm and generation of first population

The initializing algorithm first applies AGNES clustering algorithm (Kaufmann and Rousseeuw, 1990) and then combines a random and a serial scheme to generate the initial population.

The clustering algorithm splits all the possible destinations both in the time and in the space, by minimizing the average distance between destinations of the same cluster. As each destination is assigned to one cluster, it is possible also to define the cluster of a tour, as the prevailing cluster of destinations belonging to it. This information is exploited by the solution procedure, by assuming that a rational solution is to assign a destination to a tour of the same cluster. The procedure of random selection assigns a vehicle by random to each delivery and generates a new tour when a given quota of the vehicle capacity is exceeded. The procedure of serial selection assigns a generic delivery to one tour, which is taken with probability $P_1$ among those belonging to the same cluster of that delivery and with probability $(1-P_1)$ to the others. In any case, only tours having available capacity are considered and a new tour is generated if necessary.

4.3 Optimization algorithm

The optimization algorithm computes sub-optimal delivery tours by applying a first genetic algorithm to solve the classic Vehicle Routing Problem (VRP), a second genetic algorithm that solves a Travel Salesman Problem (TSP) for each tour of the solution of VRP and then computing their optimal starting time. Both genetic algorithms have the usual framework composed by: crossover, mutation, constraints verification, fitness computation, reproduction and
population update. However, the peculiarities of the dynamic VRP problem require the introduction of some modifications, which concern:

*Crossover.* As different individuals of a solution of the VRP (i.e., different tours) may have a different number of chromosomes, to avoid repeating or missing some customers an incidence matrix has been introduced, which associates tours and customers univocally, and then makes it possible tie-break crossover on a randomly selected column.

*Mutation.* Mutation alters the structure of chromosomes in order to explore some new portion of the search space. Information regarding clusters of tours are here exploited to guide the process toward presumably better solutions. For each individual, a chromosome selected at random (i.e.: a delivery in a tour) is assigned to another tour, which is taken with probability $P_2$ among those belonging to the same cluster of that delivery and with probability $(1–P_2)$ to the others.

*Computation of tour starting time.* Since early arrivals always would require waiting for the beginning of the time window, the starting time is computed as the earliest time instant that complies at least one of time window constraints.

*Fitness computation and reproduction.* Fitness of all individuals is defined as the sum of the following terms: the operative costs of the delivery process, the costs related to delays of deliveries (if any), the fixed cost associated to any tour, and the gains obtained by carrying out the deliveries.

The bi-level structure of the solution procedure implies that the fitness is firstly computed for the inner problem (i.e.: for different individuals of TSP with time windows that take the delivery list of the routing problem as an input) and then for the main outer VRP by taking for each tour the fitness value obtained by solving TSP. After fitness evaluation, all members of the current population are ranked by their fitness value and best individuals are selected for reproduction by applying a stochastic sampling.

### 4.4 Population update

The current solution is continuously updated by checking whether the starting time of a tour has been reached or a new arrival has reached the logistic center. In the first case, provided that a given minimum number of generations has been processed, the program prints the loading list of the tour and removes it from the current population. In the second case, it codes the new deliveries in terms of chromosomes, computes an initial solution for them and adds such a solution to the current population.

### 5. THE DYLOG SOFTWARE

Dylog is a software tool intended for logistic operators, in particular for managers of urban logistic base. It allows them defining in real-time the best solution of delivering program, in terms of time, costs and logistics utilization. Unlike Dilog, Dylog is a dynamic software and can deal with both planned and unexpected goods arrivals at the logistic base as well as vehicles dispatching and personnel availability.

Furthermore, Dylog contains a simulation tool that allows users estimating time and costs for a certain sequence of goods arrivals, including also a part of randomly deliveries generated by the program to simulate unexpected
arrivals at the logistic center. In such a way, it can support strategic decisions about logistic features setting up.

Finally, Dylog is arranged to display in real-time the current positions of vehicles on a geo-referenced map.

Dylog is made up of two components:
- Calculation subroutines, developed in C++ language;
- User interface, developed in Delphi 7 language.

The calculation subroutines consist of:
- an algorithm for delivery tours optimization, which assigns the deliveries to the vehicles available at the logistic base;
- an algorithm for route optimization, which solves a classic Travel Salesman Problem (TSP) for each set of deliveries defined by the previous subroutine;
- a procedure for vehicle departures scheduling;
- a procedure for initial deliveries clustering, whose role is to support the first optimization algorithm.

As for the user interface, it is made by a main dialogue window that shows the current status of the logistic hub (Figure 6). Push buttons depicted in Figure 7 allows the user entering into the four main Dylog functionalities (optimization, simulation, vehicle position monitoring and data input).

<table>
<thead>
<tr>
<th>Delivery Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
</tr>
<tr>
<td>----</td>
</tr>
<tr>
<td>001</td>
</tr>
<tr>
<td>002</td>
</tr>
<tr>
<td>003</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Delivery Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
</tr>
<tr>
<td>----</td>
</tr>
<tr>
<td>001</td>
</tr>
<tr>
<td>002</td>
</tr>
<tr>
<td>003</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Current Vehicle Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
</tr>
<tr>
<td>----</td>
</tr>
<tr>
<td>001</td>
</tr>
<tr>
<td>002</td>
</tr>
<tr>
<td>003</td>
</tr>
<tr>
<td>004</td>
</tr>
<tr>
<td>005</td>
</tr>
<tr>
<td>006</td>
</tr>
<tr>
<td>007</td>
</tr>
<tr>
<td>008</td>
</tr>
<tr>
<td>009</td>
</tr>
</tbody>
</table>

©Association for European Transport and contributors 2006
Dylog inputs consist of three sets of data, namely:
- Deliveries data;
- Vehicle fleet characteristics;
- Reference travel times.

Data on deliveries regard sender, carrier and receiver names and addresses, goods and parcels characteristics, requested delivery time-windows (Figure 8). Unlike other similar programs devised for solving the vehicle routing problem, Dylog allows the receiver selecting multiple separate time windows during the same day.

Data on vehicle fleet concern vehicle capacity, both in volume and weight. Travel times are related to every possible origin–destination pair within the delivery area and should be computed in a pre-processing analysis (Figure 9).
Apart from area-specific data, the initialization procedure of Dylog allows the user defining:

- algorithm parameters, such as crossover, mutation and replacement probability for each of the genetic algorithms applied to either the vehicle routing or the TSP (Figure 10);
- operational parameters, such as the average time necessary to load and unload a vehicle or to deliver a parcel.

After inputting activities, either the optimization or the simulation procedure can be launched.

In case of simulation, a dialogue window opened by the related button allows the user setting the simulation speed, that is the ratio between real and virtual...
events durations (Figure 11); of course, the higher is the speed the less confident are the simulation results. Moreover, the simulation window shows both the delays of expected arrivals at the logistic hub and the delays of deliveries with respect to the time window.

![Simulation dialogue window.](image)

In case of optimization, unexpected deliveries arriving during the day have to be inputted in real-time and optimization algorithms restarts automatically taking into account both new inputs and previous temporary solutions. Output visualization regards deliveries, vehicles and tour status, which are updated continuously in the main dialogue window (Figure 6). As for the deliveries, it shows whether if they are still waiting in the base, running on tour, delivered or refused. As for the vehicles, the window indicates if they are still available at the base or on tour and which tours have been assigned to. Finally, regarding current tours, the Dylog main dialogue window visualizes their planned departure time and predicted arrival time. The set of the deliveries assigned to each tour, with the related input data, is listed in an inferior window that can be opened with a double click. Vehicles position can be visualized on a geo-referenced map (Figure 12): in case of simulation it shows the calculated position; in case of optimization the actual position can be monitored, if the fleet is equipped with a GPS locating device and either a GPS/GPRS or a UMTS cellular phone.

©Association for European Transport and contributors 2006
Both for simulation and optimization, the on-going trend of the fitness value is plotted in a graph that changes dynamically while the optimization process runs (Figure 13).
6. SIMULATION OF REAL-TIME OPERATIONS AT AN URBAN LOGISTIC CENTER BY DYLOG

Preliminary numerical tests have been carried out by simulating the dynamic delivery procedure to an Urban Logistic Center (ULC) located in the town of Cosenza, a town of 75,000 inhabitants in Southern Italy. The road network is modeled by a graph composed by 491 nodes and 1,029 links. The fleet is composed by 9 vehicles: 4 diesel powered vans having capacity of 3,000 kg and 7.5m$^3$ and 5 small electric vehicles of 1,500kg and 3.5m$^3$. In Cosenza there are 1,391 retailer shops that attract 65,500 kg and receive 511 deliveries every day, on average. By assuming that ULC deals with 200 deliveries in a generic day, we generated 100 planned and 100 non planned deliveries. Each of them is composed by a number of items ranging from 1 to 10 and has time windows varying from a minimum size of 4 and a maximum of 12 hours. Each item can weigh from 1 to 20 kg and has a volume ranging from 0.01 to 0.04m$^3$.

An example of simulation results is shown in Fig.14, which depicts the number of deliveries arrived, released and waiting at ULC, the number of vehicles in tour and the value of the fitness, which is normalized with respect to the number of deliveries at ULC. Planned deliveries are available at ULC at the beginning of the service time (6 a.m.). Non planned deliveries arrive at ULC in 4 different shipments: at 8:40, 11:20, 13:10 and 15:10. At the beginning of operations, the fitness has very low improvements (1.4% after 170 generations) and this highlights the effectiveness of the clustering algorithm applied to find out the initial solution.
As each new arrival reaches the logistic center, the fitness function increases, but returns back to similar values after only few minutes. Specifically, the first two unpredicted arrivals at 8:40 and 11:20 are very well absorbed by the algorithm and do not affect fitness values conclusively. Further unexpected arrivals at 13:10 occur at a very unfavorable time instant, because just few minutes before two vehicles had left the logistic center, so that there are fewer available resources. The fitness value has a sharp increase up to 4 times the previous values. In such a condition, the optimization algorithm reveals to be both efficient and effective, as it takes only 7 minutes (in simulation) to reduce the fitness as 70%. It is worth noting that the speed simulation was about 2 times more than the real time. Thus, in real applications the algorithm could process 2 times more computations and would achieve so the same result in a 2 times shorter time, that is in only 3.5 minutes. After that critical condition, several tours start and, each time a vehicle leaves the logistic center or a group of items is delivered, the value of the average fitness changes. At 15:10, a new group of unexpected deliveries arrives at the terminal, when only 1 vehicle is available. Improvements of the fitness are quite small and so its unitary value (fitness per delivery) is slightly increasing, because of the decreasing number of waiting deliveries. As the number of deliveries to release diminishes, the fitness function decreases and the process ends when the last item is delivered.

Sensitivity analysis on algorithm parameters underlines that the algorithm is robust with respect to the probabilities of mutation and cross-over, for both routing and TSP genetic algorithms (values of about 0.80 being sub-optimal). With respect to usual static applications of genetic algorithms, a real-time application requires much smaller values of those parameters that would improve algorithm effectiveness but reduce its efficiency. In our tests we experienced better solutions by using small values for the population size (less than 100 individuals) as well as by reducing the number of generations of the TSP algorithm (less than 50). In fact, the peculiar framework of the solution procedure exploits the clustering algorithm to find a good initial solution and benefits from genetic algorithms almost to update the current solution after a heavy modification of initial data.
Fig. 14. Simulation results of the real-time optimization of deliveries at the logistic center.

7. CONCLUSIONS

In this paper we have described two software tools for planning and operating an urban logistic center. The former software, called DiLog, is devised to determine the optimal assignment of the delivery points to several possible terminals. It allows then the planner locating and sizing the terminals of the urban logistic system. The latter software, called Dylog, is a procedure for the real-time management of deliveries at an urban logistic center, where items can arrive continuously, even without notice. In these conditions, the promptness of the real-time procedure is crucial to ensure effective operations. Many numerical tests have been conducted by simulating the operations at a logistic center during a typical day. Results have shown that the algorithm succeeded in improving the solution in very few minutes, even when strong perturbations occur.

8. BIBLIOGRAPHY


