A solution for the optimisation of freight transport in urban areas

R. Raicu[1], S. Raicu[2] & E. Rosca[2]
[1]*Transport Systems Centre, University of South Australia, Australia*
[2]*School of Transportation Engineering,*
Polytechnic University of Bucharest, Romania

Abstract

The paper presents an optimisation method for vehicle routing with time windows and scheduling of freight distribution/collection in urban areas. The differences with the classical approaches are outlined. The optimal routing solution that minimises the total travel distance is determined by investigating trips nature, obtaining a better initial solution and turning the problem into a transportation one. Based on the generated set of routes, the scheduling routine computes the minimum number of vehicles that could perform the container distribution/collection and the schedule for each vehicle. An alternative scenario with a reduced number of vehicles is compared with the initial one, taking into consideration vehicles utilisation, hiring costs, and penalty costs for the cancelled trips. The results presented in this paper are from a study carried out for the metropolitan area of Bucharest, where the method proposed for freight transport optimisation was successfully implemented.
Keywords: urban freight transport, optimisation, vehicle routing with time window, scheduling.

1 Introduction

Transport companies, customers, local and central authorities, for different reasons, are interested in the development of integrated logistics to satisfy the requirements of ongoing transport development. The optimisation of freight distribution/collection in urban areas implies solving transport routing and scheduling problems. In general, the objectives of these two complex combinatorial problems consist of minimising the fleet size, the number of

WIT Transactions on The Built Environment, Vol 89,
www.witpress.com, ISSN 1743-3509 (on-line)
doi:10.2495/UT060341

'unloaded trips', the total haulage distance, the transport costs or an aggregated function of the above mentioned.

The Vehicle Routing Problem (VRP) involves finding a set of routes, starting and ending at a depot, which services a set of customers. The problem is defined on a graph $G=(V,A)$, where $\mathrm{V}=\{\mathrm{v}_0,...,\mathrm{v}_\mathrm{n}\}$ is a vertex set including the depot (v_0) and the customers $(\mathrm{v}_\mathrm{i},\mathrm{i}=1,...,\mathrm{n})$, and $A=\{(v_i,v_j)\,|\,i\neq j;v_i,v_j\in V\}$ is the arc set. Each arc set $(\mathrm{v}_\mathrm{i},\mathrm{v}_\mathrm{j})$ has associated non-negative values c_{ij} representing distance, travel cost or travel time. If $\mathrm{c}_{\mathrm{ij}}=\mathrm{c}_{\mathrm{ji}}$ for all $\mathrm{v}_\mathrm{i},\mathrm{v}_\mathrm{j}\in\mathrm{V}$, then the problem is called symmetrical. Also, if $\mathrm{c}_{\mathrm{ik}}+\mathrm{c}_{\mathrm{kj}}\geq\mathrm{c}_{\mathrm{ij}}$, then the triangle inequality is satisfied for vertices $\mathrm{v}_\mathrm{i},\mathrm{v}_\mathrm{j},\mathrm{v}_\mathrm{k}$. Each customer has a given demand q_i and the total demand hauled by any vehicle may not exceed the vehicle capacity Q. The objective function is to minimise the total travel distance, cost or time. When for each customer, the start of service must be within a given time interval $[a_i,b_i]$, the problem turns into a Vehicle Routing Problem with Time Windows (VRPTW). A vehicle is allowed to arrive before the beginning of the time window and wait at no cost, but is not permitted to arrive after the end of the time window.

Kolen *et al* [1] have presented the first optimisation method for the VRPTW. The method computes lower bounds using dynamic programming and state space relaxation. Branching decisions are taken on route-customer allocations. Fisher [2] describes an algorithm based on the K-tree relaxation of the VRPTW. Desrochers *et al* [3] have applied column generation to the VRPTW with a free number of vehicles. This column generation is in fact equivalent to the Dantzig-Wolfe decomposition. The master problem consists in finding a minimum cost set of paths, among all generated paths, that ensures the service of all the customers. For computational reasons, Desrochers *et al* [3] solve the Linear Programming (LP)-relaxation of a Set Covering Problem (SCP) instead. This does not change the final result, if the triangle inequality of time and cost is satisfied.

The Scheduling Problem (SP) deals with finding the minimum number of vehicles, which could cover the routes generated in VRPTW, with respect to the time window constraints. The problem is equivalent to finding the minimum number of subsets the set of all routes could be divided into, satisfying the time constraints Bodin [4].

The paper focuses especially on VRPTW and SP problems for the optimisation of containers transport. The classical VRPTW problem is modified such as the transportation of containers takes place as follows: from the depot (container terminal) to customers, and from customers to the depot.

2 Containers distribution/collection model

As shown in Figure 1, a container terminal (T) has to serve a set of customers $C = \{C_i \mid i = 1,..., n\}$, using a minimum number of hired vehicles within a time window $[0, \tau]$. At the beginning of the time window, there are q_i containers at the terminal to be hauled to the customer C_i, and p_i containers at customer C_i to be hauled to the terminal.

The time window interval is related to the schedule of container trains arriving and departing from the terminal. The assumption that the distances and the travel time matrix are symmetrical is made. The vehicles are hired for the entire time window interval, and in addition to the hiring costs, the travel costs are proportional to the travel distances. Each vehicle has a capacity of one container.

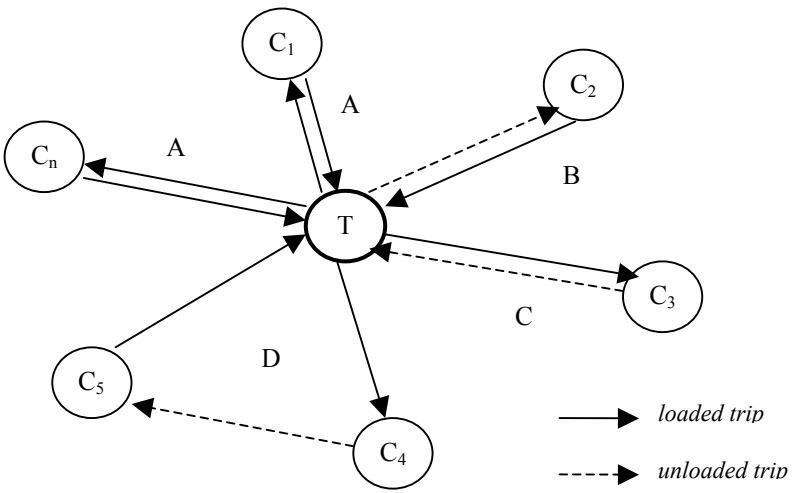

Figure 1: Types of vehicle trip.

Figure 1 shows that there are four types of trip for container distribution:

- type A trip – the vehicle hauls a container both ways
- type B trip – the vehicle travels unloaded to the customer and returns with a container to the terminal
- type C trip – the vehicle hauls a container to the customer and returns unloaded to the terminal
- type D trip – the vehicle hauls a container to a customer, travels unloaded to another customer and returns to the terminal with a new container.

The optimisation algorithm for container distribution is divided into two phases: routing and scheduling.

The routing phase can be mathematically expressed as:

$$\min F = 2\sum_{i=1}^{n}(\alpha_i + \beta_i + \gamma_i)d_i + \sum_{\substack{i,j=1 \\ i\neq j}}^{n}\delta_{ij}(d_i + d_{ij} + d_j),$$

subject to:

$$\alpha_i + \beta_i + \sum_{\substack{j=1 \\ j\neq i}}^{n}\delta_{ji} = p_i, \quad i = 1,...n \tag{1}$$

$$\alpha_i + \gamma_i + \sum_{\substack{j=1 \\ j\neq i}}^{n}\delta_{ij} = q_i, \quad i = 1,...n \tag{2}$$

$$\alpha_i, \beta_i, \gamma_i, \delta_{ij} \geq 0, \quad i, j = 1,...n, \tag{3}$$

where:

$\alpha_i, \beta_i, \gamma_i$ are the number of type A, B and respectively C trips for customer C_i

δ_{ij} – the number of type D trips, involving an unloaded trip from customer C_i to customer C_j

d_i – the travel distance between the container terminal T and customer C_i

d_{ij} – the travel distance between customers C_i and C_j.

The objective function states that the total travel distance should be minimised. The above constraints refer to the containers that have to be hauled from customers (1), and to customers (2). Non-negativity constraints are also added (3).

As constraints (1) and (2) state, increasing α_i by 1 unit, means to decrease the sum of the other terms by 1 unit. It is not difficult to demonstrate that in the objective function this is translated into a reduction of the total travel distance. Therefore, α_i must have the maximum possible value $\alpha_i = \min(p_i, q_i)$. Subtracting α_i trips of type A for each customer, two sets of customers are obtained:

- S = {Ci} – the set of customers that still have containers to be sent to the terminal:

$$\beta_i + \sum_{\substack{k=1 \\ k\neq i}}^{n}\delta_{ki} = p_i - q_i$$

- L={Cj} – the set of customers that still have to receive containers from the terminal:

$$\gamma_j + \sum_{\substack{k=1 \\ k \neq j}}^{n} \delta_{jk} = q_j - p_j$$

As one can see, if the triangle inequality is satisfied $(d_i + d_j \geq d_{ij})$, decreasing β_i and γ_j by 1 and increasing δ_{ji} by 1, further reduces the objective function by $(d_i + d_j - d_{ij})$. Thus, the optimisation problem could be transformed into a transportation problem with the L set of customers representing the origins and the K set of customers representing the destinations. If "demand" equals "production" $\sum_{i \in S} (p_i - q_i) = \sum_{j \in L} (q_j - p_j)$, the transportation problem is balanced, otherwise is unbalanced.

The paths generated by the transportation problem, together with the α_i trips for each customer represent the vehicle routing problem solution.

The scheduling routine finds the minimum number of hired vehicles v, such as all the trips are performed within the $[0, \tau]$ time window. Let M be the set of all trips. The scheduling problem is stated as:

$$\min v,$$

subject to:

$$\bigcup_{i=1}^{v} M_i = M \tag{4}$$

$$M_i \bigcap M_j = \varnothing, \quad i \neq j \tag{5}$$

$$\sum_{m_j \in M_i} \theta_j \leq \tau, \quad i = 1,...v \tag{6}$$

where:

M_i is a sub-set of M

θ_j – the m_j trip time.

The number of vehicles v should satisfy $v\tau \geq \sum_{m_j \in M} \theta_j$. The lower bound of v is:

$$v_{min} = \left\lfloor \frac{\sum_{m_j \in M} \theta_j}{\tau} \right\rfloor + 1$$

where $\lfloor \ldots \rfloor$ is the integer part.

If it is possible to divide the M set of trips in v_{min} sub-sets, so the constraints (4–6) are satisfied, then a feasible solution $v^* = v_{min}$ is obtained. Otherwise, the number of vehicles is increased by 1, and the procedure is repeated until a value v^* satisfying all the constraints is found. For the same v^*, many possible partitions of the set M might exist, and new conditions, such as the uniform utilisation of vehicles, might be added.

3 Case study

The algorithm described before has been developed to optimise the distribution/collection of containers in the metropolitan area of Bucharest (Figure 2). The container terminal is located in the outer suburbs of Bucharest. Figure 2 shows a situation with 12 customers with different locations in the city area that have to be served within a time window of 8 hours.

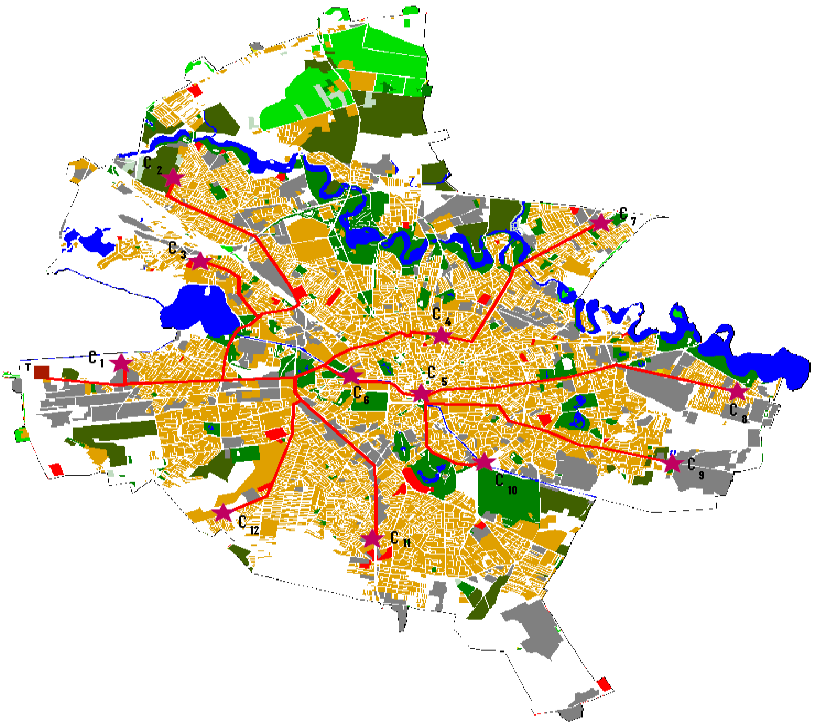

Figure 2: Spatial distribution of customers.

The computer program developed for the optimisation of container distribution/collection requires the following input-data:

- the number of containers pi to be hauled from customer Ci to the terminal T, and the number of containers qi to be hauled from the terminal T to customer Ci (Table 1)
- the database containing customer–customer and terminal–customer distances and travel times
- the time window interval, the additional time for loading/unloading the containers, and other commercial procedures at the terminal, and at the customers.

Table 1: Containers initial state.

Customer	Containers to be received from the terminal (q_i)	Containers to be sent to the terminal (p_i)	Trips of type A (α_i)
C_1	3	5	3
C_2	7	7	7
C_3	4	5	4
C_4	7	6	6
C_5	5	6	5
C_6	5	2	2
C_7	2	5	2
C_8	7	4	4
C_9	6	7	6
C_{10}	5	4	4
C_{11}	7	6	6
C_{12}	6	3	3
Total	**64**	**60**	**52**

Table 2: Demand, supply and customer to customer distances.

		Destinations						Supply (containers)
		C_1	C_3	C_5	C_7	C_9	Γ	
Origins	C_4	29	30	29	42	47	0	1
	C_6	20	24	24	41	43	0	3
	C_8	51	52	50	58	52	0	3
	C_{10}	34	38	35	47	46	0	1
	C_{11}	25	31	29	48	46	0	1
	C_{12}	22	28	27	46	48	0	3
Demand (containers)		2	1	1	3	1	4	12

Consequently, one obtains a transportation problem as shown in Table 2.

The cell located at the intersection of line Ci with column Cj represents the travel distance [km] in a type D trip on the Terminal → Customer Ci → Customer Cj → Terminal route $\left(d_i + d_{ij} + d_j\right)$. The problem is unbalanced and a "dummy" destination, customer Γ, is introduced, with a demand of 4 containers and 0 length distances connecting the origins. The solution of the transportation problem is shown in Table 3.

Table 3: Transportation problem solution.

		Destinations						Supply (containers)
		C_1	C_3	C_5	C_7	C_9	Γ	
Origins	C_4	–	–	–	1	–	–	1
	C_6	–	1	–	2	–	–	3
	C_8	–	–	–	–	–	3	3
	C_{10}	–	–	–	–	1	–	1
	C_{11}	–	–	–	–	–	1	1
	C_{12}	2	–	1	–	–	–	3
Demand (containers)		2	1	1	3	1	4	12

The value of the cell located at the intersection (Ci x Cj) represents the number of type D trips to be performed (δ_{ij}). The value of the cells in column Γ represents the number of type C trips to be performed (γ_i).

From Table 1 and Table 3, one obtains the final solution of the routing problem, consisting of 52 trips of type A, 8 trips of type D and 4 trips of type C. The travel times along selected routes are shown in Table 4.

Table 4: Routes travel times [min.].

Route	Travel time	Route	Travel time
$T - C_1 - T$	20	$T - C_{10} - T$	80
$T - C_2 - T$	46	$T - C_{11} - T$	36
$T - C_3 - T$	30	$T - C_{12} - T$	24
$T - C_4 - T$	60	$T - C_4 - C_7 - T$	63
$T - C_5 - T$	52	$T - C_6 - C_3 - T$	36
$T - C_6 - T$	50	$T - C_6 - C_7 - T$	61
$T - C_7 - T$	88	$T - C_{10} - C_9 - T$	69
$T - C_8 - T$	108	$T - C_{12} - C_1 - T$	33
$T - C_9 - T$	92	$T - C_{12} - C_5 - T$	40

Each container needs 15 minutes for loading/unloading and other commercial procedures. The total travel time along all routes is 4835 minutes. Considering a time window of 8 hours, the minimum number of vehicles is v_{min}=11. The computational program succeeds in finding a scheduling solution with 11 vehicles.

As Figure 3 shows, it was attempted to maximise the vehicles utilisation as much as possible. Note that for one of the vehicles the utilisation does not justify the hiring costs for the entire time interval.

Assuming the set of trips can be split into pre-emptive and non pre-emptive trips, then the last vehicle might be eliminated if the hiring cost exceeds the penalty costs due to failure of performing some non pre-emptive trips within the time window. The cancelled trips are performed within the next time window as pre-emptive ones. Thus, the optimisation could be extended to cover two time windows, if the number of containers to be transported within the next time window is known.

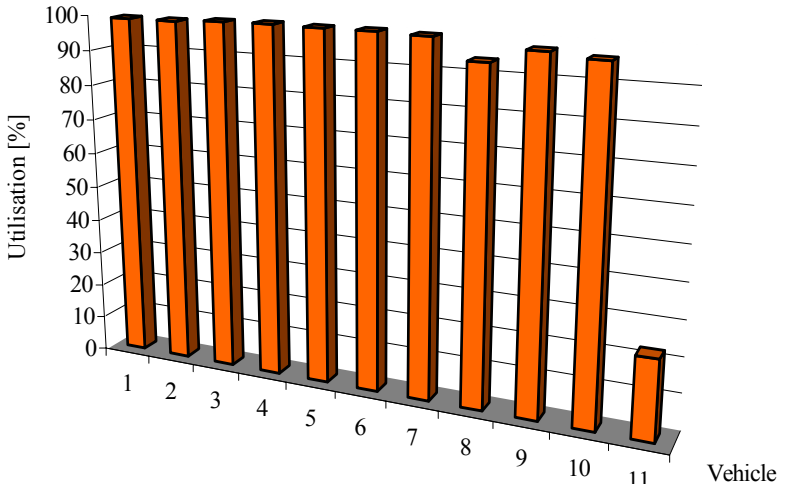

Figure 3: Vehicles utilisation.

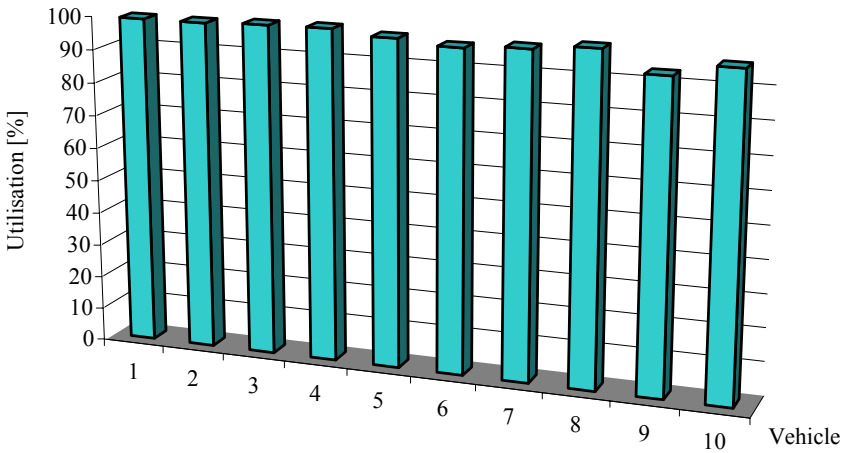

Figure 4: Vehicles utilisation (non pre-emptive trips case).

Considering as non pre-emptive trips those trips that haul containers from the terminal to the customers, the problem might be solved with 10 vehicles. The non pre-emptive trips are cancelled starting with the most time consuming type C trips, and if this is not sufficient some type D trips are turned into type B trips to be completed, and type C trips to be cancelled.

Figure 4 shows the utilisation of all 10 vehicles, when one type C trip (T – C8 – T: travel time 123 minutes) has been cancelled.

The computer model was developed using Java 2™ capabilities, and Microsoft Access database to store input-data. The tests performed on Bucharest area, with up to 50 customers provided almost "instantaneous" solutions on a Pentium III PC (computing time and results displaying in less than 1 second).

4 Conclusions

Both mathematical and computer model enable the activities of distribution and collection of containers from and to the customers, with minimum of resources managed by the terminal.
The solution presented would also alleviate congestion on the main urban road arteries during peak hours.

The vehicle routing solution provides the set of routes to be covered, minimising the total travel distance. The scheduling solution offers the minimum number of vehicles and the transport schedule to be carried out by each hired vehicle. An alternative with a reduced number of vehicles and cancelled trips is computed. The economic validation of this alternative assumes that the hiring cost of a vehicle would exceed the penalty costs of the cancelled trips.

References

[1] Kolen, A., Kan, A.R., & Trienekens, H., Vehicle Routing with Time Windows, *Operations Research*, **35**, pp. 256-273, 1987.
[2] Fisher, M., Optimal Solution of Vehicle Routing Problems using Minimum K-Trees, *Operations Research*, **42**, pp. 626-642, 1994.
[3] Desrochers, M., Desrosiers, J., & Solomon, M., A New Optimisation Algorithm for the Vehicle Routing Problem with Time Windows, *Operations Research*, **40**, pp. 191-211, 1992.
[4] Bodin, L.D., Twenty Years of Routing and Scheduling, *Operations Research*, **38**, pp. 571-579, 1990.