Journal of Food Engineering 51 (2002) 85–91

JOURNAL OF FOOD ENGINEERING

www.elsevier.com/locate/jfoodeng

Distribution of fresh meat

C.D. Tarantilis, C.T. Kiranoudis *

Department of Chemical Engineering, National Technical University of Athens, Heroon Polytexniou 9, Zografou Campus, GR 15780 Athens, Greece

Received 14 October 2000; accepted 23 January 2001

Abstract

This paper deals with a real-life distribution problem formulated as an open multi-depot vehicle routing problem (OMDVRP) that was encountered by a major Greek industry distributing fresh meat from depots to its customers (butchers' shops) located in an area of the city of Athens. To solve the problem, a new stochastic search meta-heuristic algorithm termed as the list-based threshold accepting (LBTA) algorithm is proposed. The proposed routing plan gives answers to a number of operational decision problems and provides significant economic benefits for the company. © 2001 Elsevier Science Ltd. All rights reserved.

Keywords: Logistics; Distribution management; Optimization; Metaheuristics

1. Introduction

In today's fresh meat industry, distribution cost constitutes a significant part of the operational costs of a company. Its share has experienced a steady increase, since faster, more frequent, and timely shipments are required as a result of request for total quality management and sharp global-size competition. In addition, perishability issues arising in cases of fresh meat render the time period between preparation and trading date to be of major interest for both producers and traders, enforcing the companies to employ efficient distribution systems (Tarantilis & Kiranoudis, 2001).

This paper deals with a real-life vehicle routing problem (VRP) that occurs in a major Greek industry distributing fresh meat from depots to its customers (butchers' shops) located in an area of the city of Athens. The management of the company estimated that the fixed cost of the vehicles involved in the distribution process of fresh meat is high enough, and therefore decided to assign the distribution task to hired vehicles. The fleet of the hired vehicles is parked at two distribution centres located next to the abattoirs of the company. The authors were asked to frame a set of vehicle routes that minimises the total travelling and vehicle operating costs of the distribution.

2. Problem review

The fact that the fleet of vehicles is hired and the distribution centre of the company is not singly led the authors to formulate this real-life problem as a combination of an open VRP (OVRP) (Sariklis & Powell, 2000) and a multi-depot VRP (MDVRP) (Wren & Holliday, 1972). Both of these discrete optimisation problems have the following common characteristics:

- vehicles are homogeneous with known capacities;
- the number of vehicles is unrestricted;
- each customer is visited once and only once by exactly one; vehicle and its demand is totally satisfied;
- the customers visited in each route have total demand less than or equal to the capacity of the vehicle assigned to serve the route;
- each vehicle makes one trip only;
- the objective is to find the set of routes, and minimise the total distance travelled.

The concept of the OVRP is involved in the formulation of the real-life problem since it allows the hired vehicles to be assigned to routes in which they do not have to return to the company's distribution centre, after serving the customers, providing the company with the set of routes that minimises the total distance travelled. Furthermore, since the distribution process of the company operates from two depots, our problem borrows ideas from the concept of the MDVRP.

Since both the OVRP and MDVRP are characterised as hard combinatorial optimisation problems (Sariklis & Powell, 2000; Salhi & Sari, 1997), their solution has been

* Corresponding author. Tel.: +30-1-772-1503; fax: +30-1-772-3228.
E-mail address: kyr@chemeng.ntua.gr (C.T. Kiranoudis).

PII: S0260-8774(01)00040-1

Notation

$c(s)$	cost of the solution s
k	last node in a route
LBTA	list-based threshold accepting
L_s	values in storage scheme
m	move
MDVRP	multi-depot vehicle routing problem
$N(s)$	neighbourhood of the solution s
OMDVRP	open multi-depot vehicle routing problem
OVRP	open vehicle routing problem
s	current solution
s'	neighbour solution of the current solution s
S	solution space
SA	simulated annealing
$S(m)$	subset of S
T_h	threshold value
T_{ho}	initial value of threshold
T_{Norm}	normalised threshold value
$T_{New\ Norm}$	normalised threshold value in the Step 3
i, j	customers of a route
Z	set of all moves of a problem

achieved mainly by non-mathematical programming methods that describe how to find good solutions in a reasonable amount of time (Chao, Golden, & Wasil, 1993). These methods were termed as heuristics or heuristic algorithms. Nevertheless, the lack of flexibility of heuristics when formulation of the problem had changed, their unsuitability for experimentation, and the fact that they were not able to produce very often high-quality solutions, led the researchers to produce higher-level procedures based on generic principles for developing heuristics capable of solving a large range of problems more efficiently than the specific simple heuristics do. These procedures compose a new type of advanced heuristic algorithm named as meta-heuristic algorithm (Hindsberger, 1998).

The list-based threshold accepting (LBTA) algorithm is such a meta-heuristic and is presented in this paper for solving a combination of the OVRP and MDVRP defined as open multi-depot vehicle routing problem (OMDVRP).

3. The LBTA algorithm

The LBTA algorithm is a meta-heuristic algorithm belonging to the class of threshold accepting-based algorithms (Dueck & Scheuer, 1990). Algorithms of this class iteratively search the solution space guided by a deterministic control parameter in the same units as the cost function to reveal promising regions for better configurations. This parameter is reduced throughout the algorithm and is termed as threshold. In the optimisation process of a typical threshold accepting algorithm, when a tentative solution configuration is generated by using an appropriate move or blend of moves, the configuration is tested with respect to feasibility constraints (e.g., total demand of customers served must be lower than the vehicle capacity) and to the move acceptance criterion (cost difference between new and previous solutions must be lower than threshold).

A *move* m is a function $m : S(m) \rightarrow S$ where $S(m)$ is a subset of the solution space S. A solution s' is called *neighbour* of the solution s if $s' = m(s)$. $N(s)$ is the *neighbourhood* of a solution s if $N(s) = \cup_{\{m\in Z\backslash s\in S(m)\}} m(s)$, where Z is the set of all moves of the problem. If the newly generated configuration satisfies the previously mentioned criteria, it replaces the old configuration and gives birth to a new one, within the general framework of the algorithm. The move acceptance criterion, however, does not guarantee that the newly created configuration will result in an overall better performance compared to its previous one, as long as cost is taken into consideration. This is clearly shown by the inequality that expresses the move acceptance criterion

$$c(s') - c(s) < T_h, \tag{1}$$

where $T_h > 0$. Implementing the inequality (1), a typical threshold accepting method accepts all improving moves and deteriorating moves as long as the cost $c(s')$ of a new solution $s' \in N(s)$ does not exceed the cost $c(s)$ of the current solution s plus a certain positive threshold that is reduced throughout the algorithm, aimed at preventing the local search from becoming trapped prematurely in a local minimum, that is to say there is no other solution $s' \in N(s)$ such that

$$c(s') \leqslant c(s). \tag{2}$$

The determination of the threshold value can interfere and transform the inequality (2) into (1), satisfying the move acceptance criterion, allowing the algorithm to continue performing local search, thus seeking better solutions. In addition, it is obvious by observing the inequality constraint (1) that the value of the threshold determines the size of the variation between proposed and current solutions. Therefore, during the optimisation process of a typical threshold accepting algorithm, the threshold is gradually lowered in such a way that it is ensured that a move that would make the current solution significantly worse is never accepted. Both the initial value of threshold and the percentage of the threshold reduction are determined by the user.

The innovation of the LBTA algorithm, originally presented in this paper, over a typical threshold accepting algorithm is based on the fact that the threshold values used in the implementation of the move acceptance criterion are determined by a list that is rejuvenated and adapted according to the topology of the solution space of the problem. The introduction of the list in the concept of the LBTA algorithm mainly contributes two main advantages over a typical threshold accepting and other types of algorithms belonging to the general class of simulated annealing (SA) (Kirkpatrick, Gelatt, & Vecchi, 1983):

- The search parameters involved in the threshold reduction strategy, such as the initial value of threshold and the percentage of the threshold reduction, are determined by the algorithm automatically, without the intervention of the user.
- The normalisation of the threshold values stored in the list of the algorithm, and the implementation of the "conservation of threshold" criterion (see algorithm description below) during the optimisation process allow this new algorithm to accept cost-increasing neighbouring solutions of the current solution, facilitating escape from local minimum points, thus increasing the possibility of finding a better, and even a global optimal solution.

The way the above principles are embodied in the LBTA algorithm is described as follows:

Phase 1

In the first phase, a list filling procedure with threshold values is conducted.

Step 1 (Initialisation phase). Having already selected an initial positive value of threshold ($T_{\text{ho}} > 0$), an initial solution is produced by firstly allocating customers to the closest distribution centre and secondly dispatching to each of them a vehicle.

Step 2. In this step of the LBTA algorithm, a local search procedure is conducted in order to compute the threshold values that represent the list of the algorithm.

Local search uses a blend of 2-Opt (Croes, 1958), 1-1, and 1-0 Exchange (Waters, 1987) moves distinguished in terms of exchanges of edges or nodes performed to convert one tour into another. These moves aim to improve the current routing plan and they are described as follows:

- 2-*Opt move*. The 2-Opt move is implemented in the case of a single route as follows. Suppose a single route consists of the following set of nodes in the given order $(\text{depot}, 1, 2, 3, \ldots, k)$, and let $\{(i, i+1); (j, j+1)\}$ be a set of two edges belonging to this route that form a criss-cross. The 2-Opt move eliminates the criss-cross and reverses a section of the route by deleting the edges $(i, i+1), (j, j+1)$ and replacing them with $(i, j), (i+1, j+1)$ to reconstruct the route. In the case of multiple routes, edges $(i, i+1)$, and $(j, j+1)$ belong to different routes but they form a criss-cross again. The 2-Opt move is applied exactly in the same way as in the case of the single route. This move is demonstrated in Fig. 1.

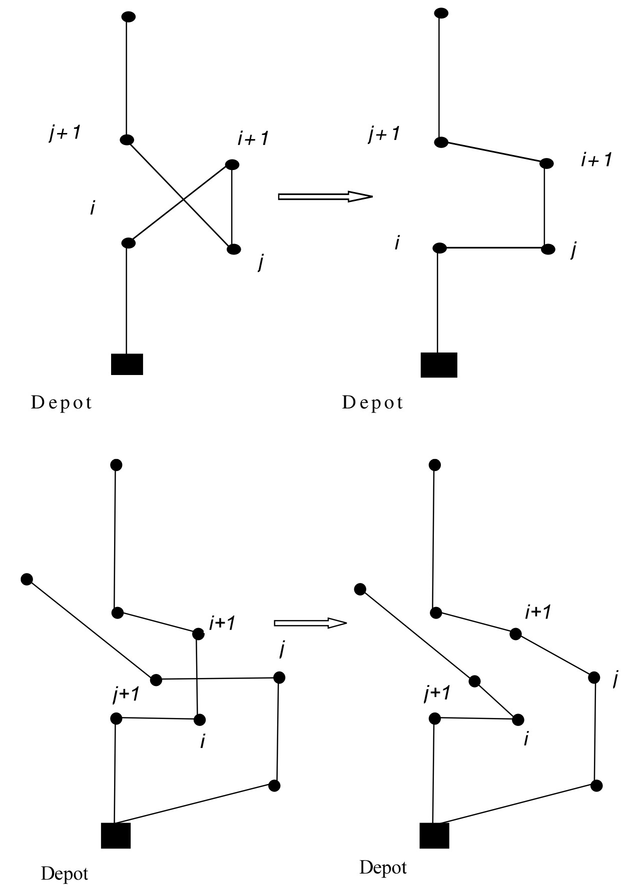

Fig. 1. The 2-Opt move for single and multiple routes used during the optimisation procedure of the LBTA algorithm.

- 1-1 *Exchange move*. The 1-1 Exchange move swaps two nodes from the same route. In fact, if it is supposed that the initial tour consists of the following set of nodes $(\text{depot}, \ldots, i-1, i, i+1, \ldots, j-1, j, j-1, \ldots, k)$, the improved one is constructed as $(\text{depot}, \ldots, i-1, j, i+1, \ldots, j-1, i, j+1, \ldots, k)$. The same procedure is conducted in the case of multiple routes but the swapping of nodes takes place between different routes. This move is demonstrated in Fig. 2.
- 1-0 *Exchange move*: The 1-0 Exchange move transfers a node from its position in one route to another position in either the same or a different route. Consequently, while the initial tour was $(\text{depot}, \ldots, i, i+1, \ldots, j-1, j, j+1, \ldots, k)$, the improved one is constructed as $(\text{depot}, \ldots, i, j, i+1, \ldots, j-1, j+1, \ldots, k)$. This move is demonstrated in Fig. 3.

The type of the move is selected by employing a stochastic move generation algorithm where the selection follows the uniform distribution. The customers (nodes) involved in the implementation of the above moves are also chosen stochastically. Then, local search procedure starts as follows:

- A neighbour s' of a current solution s is selected by using one of the local search moves

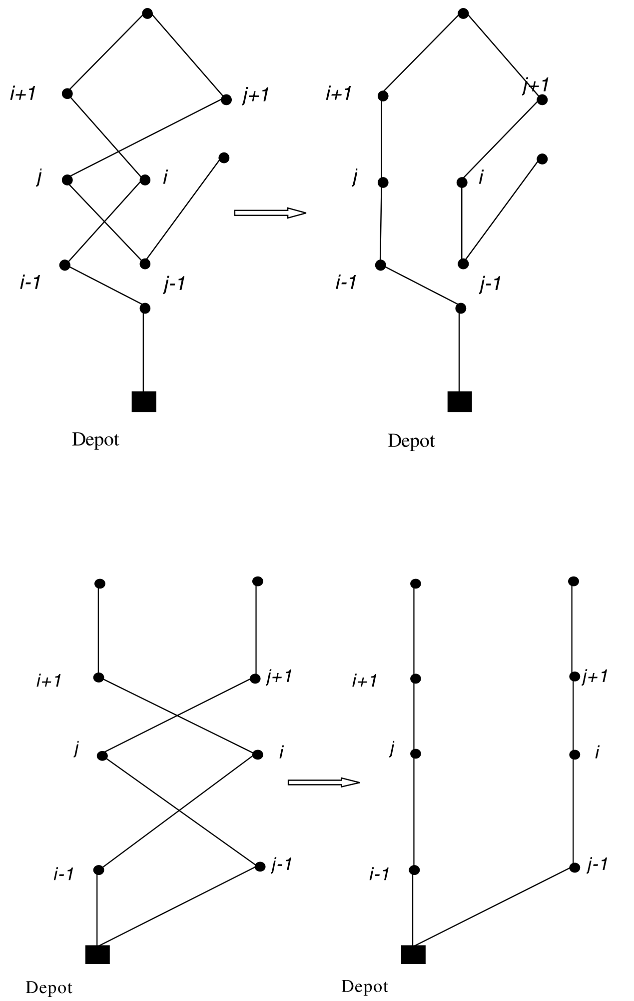

Fig. 2. The 1-1 Exchange move for single and multiple routes used during the optimisation procedure of the LBTA algorithm.

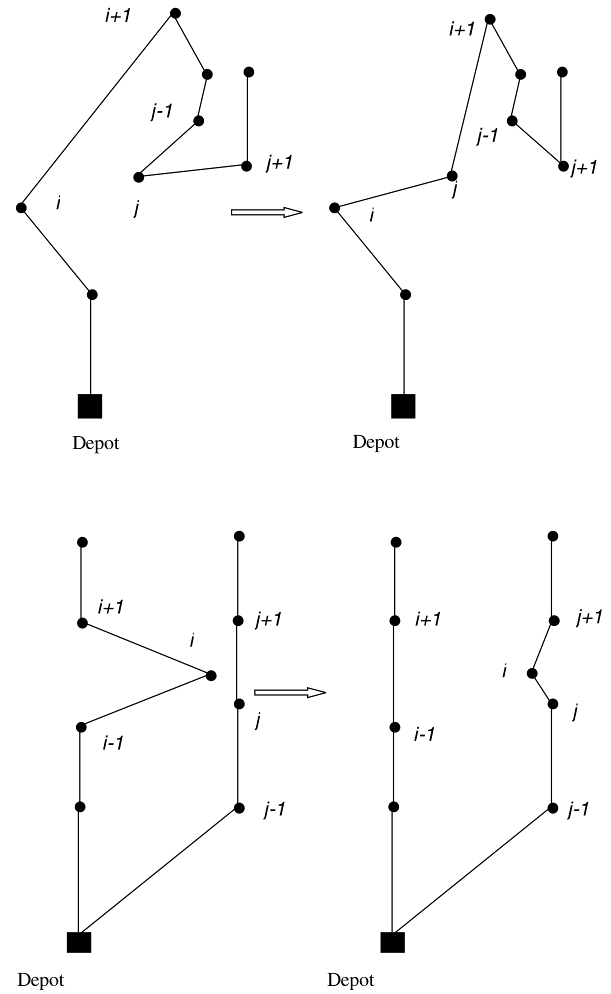

Fig. 3. The 1-0 Exchange move for single and multiple routes used during the optimisation procedure of the LBTA algorithm.

- The relative cost deviation between proposed and current solutions is calculated and normalised by means of the following criterion:

$$T_{\text{Norm}} = \frac{c(s') - c(s)}{c(s)}. \tag{3}$$

- If T_{Norm} is positive and lower than the maximum element of the list, T_{Norm} is inserted in the list. This normalisation procedure favours a slow reduction of the threshold value in the iterative procedure.
- This iterative procedure is repeated for each of the moves mentioned in the previous section until the list is exhausted.

The list serves as a memory of the variability of local function values, stored in the form of value changes from each old configuration. The storage scheme employs up to L_s (user-defined parameter) values in a binary tree list. Storage and retrieval from such a scheme have logarithmic complexity as opposed to linear complexity per operation in a simple linear storage scheme.

Phase 2

In this phase, the main optimisation algorithm is described and a threshold controlling procedure is conducted in the list of the algorithm.

Step 3. In this step, an initial solution is produced in the same way as in Step 1. Then the local search method is applied as follows:

- Using one of the local search moves mentioned in the previous section, a solution s' is generated as a neighbour of a current solution s.
- Threshold values generated with the iteration procedure are normalised, in the same way as in Step 2, and the move acceptance criterion is checked for satisfaction, using the maximum element of the list:

$$T_{\text{New Norm}} < T_{\text{Max Norm}} \Rightarrow \frac{c(s') - c(s)}{c(s)} < T_{\text{Max Norm}}$$
$$\Rightarrow c(s') - c(s) < c(s) * T_{\text{Max Norm}}. \tag{4}$$

- If this criterion is satisfied, then s is set to s' and the maximum element of the list is rejected from it, with

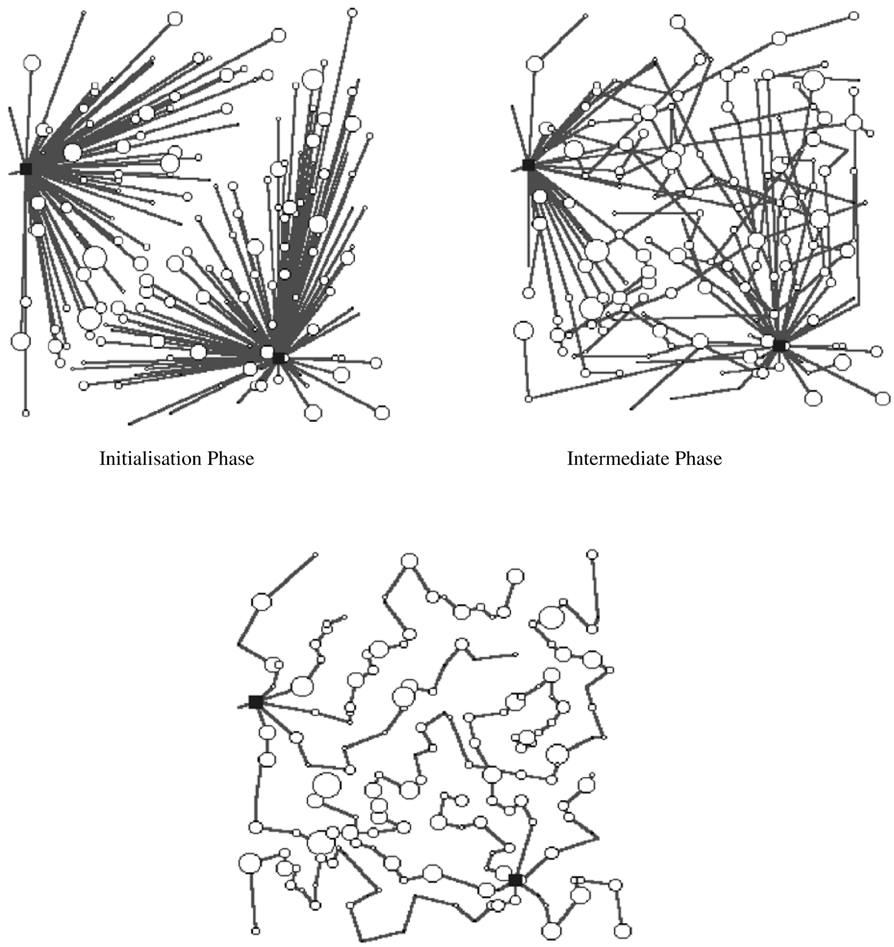

Fig. 4. The optimisation procedure of the LBTA applied to the real-life problem.

the new value of normalised threshold being appropriately inserted. If the move acceptance criterion is not satisfied, the selected move is rejected.

- The same procedure is repeated for every one of the above moves selected, until the relation (4) is not satisfied, for a number of feasible moves (stopping criterion).

Step 4 (Conservation of threshold strategy). In order to facilitate escape from local minimum points and increase the possibility of finding a better solution, the algorithm attempts to keep the values of list high enough by placing in the list the highest of the normalised threshold values found after a predefined number of feasible moves. This strategy is termed the conservation of threshold strategy and represents a delay factor in the list value adaptation.

4. Case study

This study refers to a real-life distribution problem encountered by a major Greek industry distributing fresh

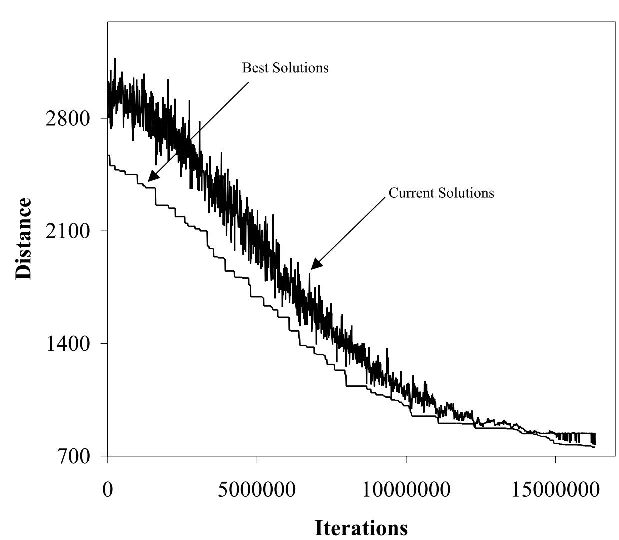

Fig. 5. The interaction between current and best solutions for the real life problem using Euclidean distances.

Table 1
Solution of the real-life problem using Euclidean distances

174 customers – 12 vehicles																	
Depot	Sequence of customers																
1	39																
1	97	76	143	13	45	109	166	24	139	63	23	104	44				
1	106	145	40	29	42	57											
1	41	105	14	92	151	6											
1	75	130	91	170	158	125	55	136	5	165	93	108	56	157	25	112	137
	110	15	164	59													
2	65	66	16	189	198	180	188	174	194	179	183	196	197	177	193	178	175
	184	186	182	192	185	187	190	176	191	195	181	199	156	33	51	129	46
	90																
1	150	107	38	12	96	54	77	81	52	83	8	82	159	124	80	95	133
	123	20	144	62	113	21	7	153	127	73	61	103	22	146	71	27	141
	102	4	89	138	64	116	28	58									
2	173	132	2	48	142	111	47	115	140								
2	68	114	17	34	122	50	10	36	120	35	149	30	86	87	169	168	9
	101	37	167	119	11	78	31										
2	67	148	98	154	121	79	85	128	84	3	100	53					
2	49	162	18	32	118	163	147	60									
2	155	171	126	19	131	117	94	152	43	88	134	26	74	161	72	135	1
	70	172	99														
Total distance travelled = 755.95																	

meat from depots to 174 customers (butchers' shops) located in the area of the city of Athens. The authors were asked to plan the efficient utilisation of the vehicles by designing a routing plan that minimises the total distance travelled subject to customer and vehicle constraints.

Real-life routing is slightly different from routing using Euclidean distances as is presented in the previous section of this paper. Data management plays a major role in the efficient functioning of a distribution system and this becomes more substantial when the distribution takes place within a detailed road network like the one of Athens.

In order to produce an efficient routing plan, apart from software that will implement the routing algorithm, information must be collected about:

- the location of the distribution centres;
- the location of the butchers' shops within the road network of the region;
- the quantity of fresh meat that have to be distributed to each shop;
- the capacity of the vehicles used;
- the spatial characteristics of the road segments of the network examined;
- the speed of the vehicles, considering the spatial characteristics of the roads and the surrounding area;
- the node numbers of the distribution centres;
- the node numbers of the customers to be served
- the node number of the erased or added node in the cases of creating a new vehicle routing scenario.

The decision support system for solving the vehicle routing problem of this study exploits this information and combines it with spatial characteristics of the area of Athens, generating automatically the vehicle routing plan by using the LBTA algorithm.

Early findings indicate considerable improvements in the operational performance of the company, reducing the total distance travelled by the vehicles by 17% in comparison with the previous algorithm used by the company for the same distribution plan. The optimisation procedure of the LBTA algorithm applied to the real-life problem is described in Fig. 4.

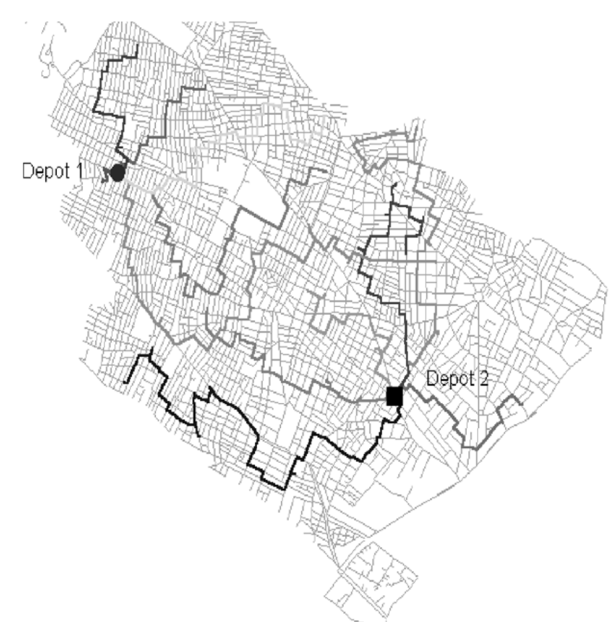

Fig. 6. The proposed routing plan for the real-life problem (road network).

Interaction between current and best solutions found during the optimisation process of the LBTA algorithm for the real-life problem is demonstrated in Fig. 5.

The solution of this case study is shown in Table 1, providing information on:

- the customers assigned to each distribution centre;
- the vehicle routes (which vehicle serves each customer);
- the total duration (or distance travelled) of all routes developed.

A graphical solution of the case study of this paper is shown in Fig. 6. Contrary to the Euclidean distance results that are presented in Table 1, no exact routing data are presented for proprietary reasons of the company. The computation time presented in Table 1 has been clocked on a computer with Pentium III/550 MHz-128MB RAM, using a computer program written in Microsoft Visual C++/version 6.0.

5. Conclusions

A new stochastic search meta-heuristic algorithm named as List-Based Threshold Accepting algorithm for solving a real-life OMDVRP is presented in this paper. A case study of a Greek industry distributing fresh meat from its distribution centres to its customers (butchers' shops) located in the area of the city of Athens is also presented. The computational performance proves that the algorithm can be employed to solve real-life problems providing very good results in fast computational speed by tuning just one of its parameters (list size). The company management was satisfied with the performance of the algorithm and endorsed it for the daily operations.

References

Chao, I. M., Golden, B. L., & Wasil, E. (1993). A new heuristic for the multi-depot vehicle routing problem that improves upon best-known solutions. *American Journal of Mathematical and Management Sciences*, *13*, 371–401.

Croes, G. (1958). A method for solving traveling salesman problems. *Operations Research*, *6*, 791–812.

Dueck, G., & Scheuer, T. (1990). Threshold accepting. A general purpose optimisation algorithm appearing superior to simulated annealing. *Journal of Computational Physics*, *90*(1), 161–175.

Hindsberger, M. (1998). *Metaheuristics an introduction*. Technical Report, Danish Defence Research Establishment.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimisation by simulated annealing. *Science*, *220*, 671–680.

Salhi, S., & Sari, M. (1997). A multi-level composite heuristic for the multi-depot vehicle fleet mix problem. *European Journal of Operational Research*, *103*, 95–112.

Sariklis, D., & Powell, S. (2000). A heuristic method for the open vehicle routing problem. *Journal of the Operational Research Society*, *51*, 564–573.

Tarantilis, C. D., & Kiranoudis, C. T. (2001). A meta-heuristic algorithm for the efficient distribution of perishable foods. *Journal of Food Engineering*, *50*, 1–9.

Waters, C. D. J. (1987). A solution procedure for the vehicle scheduling problem based on iterative route improvement. *Journal of Operational Research Society*, *38*, 833–839.

Wren, A., & Holliday, A. (1972). Computer scheduling of vehicles from one or more depots to a number of delivery points. *Operational Research Quarterly*, *23*, 333–344.