RAIRO Operations Research
RAIRO Oper. Res. **38** (2004) 345–360
DOI: 10.1051/ro:2004029

A THRESHOLD ACCEPTING APPROACH TO THE OPEN VEHICLE ROUTING PROBLEM

Christos D. Tarantilis[1], George Ioannou[1],
Chris T. Kiranoudis[2]
and Gregory P. Prastacos[1]

Abstract. In this paper we consider the operational planning problem of physical distribution *via* a fleet of hired vehicles, for which the travelling cost is solely a function of the sequence of locations visited within all open delivery routes, while vehicle fixed cost is inexistent. The problem is a special class of vehicle routing and is encountered in the literature as the Open Vehicle Routing Problem (OVRP), since vehicles are not required to return to the depot. The goal is to distribute in an optimal way finished goods from a central facility to geographically dispersed customers, which pose daily demand for items produced in the facility and act as sales points for consumers. To solve the problem, we employ an annealing-based method that utilizes a backtracking policy of the threshold value when no acceptances of feasible solutions occur during the search process. Computational results on a set of benchmark problems show that the proposed method consistently outperforms previous algorithms for solving the OVRP. The approach can serve as the means for effective fleet planning in real-life problems.

Keywords. Distribution, vehicle routing, logistics.

1. Introduction

Product proliferation, short product life, customer responsiveness and a constant thrive for expanded market share define a very competitive business environment that places heavy constraints and requirements on today's enterprises, which

[1] Management Sciences Laboratory, Athens University of Economics and Business, Greece; e-mail: tarantil@aueb.gr
[2] Department of Process Analysis and Plant Design, National Technical University of Athens, Greece.

are also responsible, in the majority of the cases, for the delivery of products or services to the final customers. Distribution is a critical problem within operations management and its applications are interrelated to transportation, stock control and warehousing. Planning delivery operations involves decision-making issues that relate to the sequence of tasks that must be performed in order to minimize operational costs and achieve desired service levels.

In this paper we address the planning problem of the physical distribution of goods to customers' locations, after all production, packaging and other value-added activities have been completed. The problem is of particular importance since the relevant operational costs (*i.e.*, labour, fuel and other associated costs) constitute a large proportion of the overall field expenses of a company [13]. The problem becomes much more relevant, from a practical perspective, when the vehicle fleet is hired, *i.e.*, vehicles do not constitute company assets. In such cases effective planning is a critical success factor for the operational effectiveness and the resulting service level, since non-company resources are responsible for the physical interface with the final customer.

The real-life situation we address in this paper can be accurately described as follows. Consider a central warehouse where items that have to be delivered to a set of customers are consolidated. Customers are geographically dispersed within a distance radius that allows for demand to be satisfied through daily deliveries. We assume that customer demand is known when a delivery schedule is determined, as is the distance or the travel time between the warehouse and each customer's location, as well as between each pair of customers' locations.

The daily delivery process is performed according to the following steps: products are loaded on appropriate hired vehicles at the warehouse and, subsequently, they are transported *via* a road network to the customers' locations. At each location, quantities that equal customer demand are unloaded from the vehicle, paperwork (shipping documents, various bills and invoices) is filled and exchanged, and then vehicles travel to subsequent customers' locations where the process is repeated. Since vehicles are hired and not owned, after all deliveries have been performed (end of a shift), the vehicles do not return to the warehouse.

The problem is a special class of the well-known Vehicle Routing Problem (VRP) [7, 18] called Open VRP (OVRP) [15]. The depot of the OVRP is the shipping area of the warehouse, where vehicles are loaded up to (or sometimes below, according to customer requests) their capacity and they perform open tours during which they visit several customers and deliver products that equal each customer's demand. The depot represents the originating (but not terminating) node of all tours. The goal is to determine the number of vehicles that are required to service all customers with the minimum operating cost, *i.e.*, in the minimum time or following the paths of minimum distance.

From a graph theory point of view, the difference between the OVRP and the classical VRP is that a solution of the OVRP consists of a set of Hamiltonian paths, instead of Hamiltonian cycles. In addition, since the OVRP sub-problem of finding the best Hamiltonian path for each set of customers assigned to a vehicle

is NP-hard [16], and it can be converted into an equivalent Hamiltonian cycle, the overall OVRP is also NP-hard [2].

Due to the computational complexity of the OVRP, heuristic and metaheuristic methods have been employed to solve practical, real-life instances of the problem. Heuristics are solution derivation methods that perform relatively limited exploration of the search space, aiming at producing a relatively good solution as quickly as possibly. Metaheuristics constitute the most promising and effective type of solution methods for all variants of the VRP [6, 17]. Metaheuristics are general-purpose mechanisms for solving hard optimization problems, which perform a thorough exploration of the solution space and guide intelligently the search process by using diversification and intensification strategies, and allowing deteriorating or even infeasible intermediate solutions to be considered during the search process. Below, we provide a brief survey of the relatively limited heuristic and metaheuristic methods proposed to-date to solve the OVRP.

Bodin *et al.* [1] were the first to address real life applications modelled as OVRPs. The authors considered an express airmail distribution problem including many practical side constraints such as delivery or pickup time windows, caps on the total route length, and capacities. Two routing problems were separately solved: One for deliveries and another for pickups, both using the Clarke and Wright algorithm [5] with appropriate modifications to account for the openness of the routes (no return to the depot) and for all side constraints.

Almost two decades later, Sariklis and Powell [15] re-examined this practical variant of the VRP and named it Open VRP. They presented a two-phase heuristic technique based on minimum spanning tree (MST) derivations, and following the "cluster first – route second" approach. The clustering procedure provided an initial assignment of customers to clusters, which was further improved by reassigning customers among them. In the routing phase, the concept of a *chain* was introduced, as an ordered sequence of customers which starts at the depot, visits several customers once and terminates at a final customer (no the depot). Each of the formed clusters was revisited during the routing phase to determine the least cost chain *via* the solution of an associated MST problem.

Finally, Brandao [2] proposed a tabu search metaheuristic for solving the OVRP, in which several features from previous tabu search implementations for the classical VRP were used [10, 11]. The neighbourhood structure of this algorithm was defined using insertion and swap moves between different routes, while intermediate infeasible solutions (in terms of capacity or maximum route length) were considered. In order to manage infeasibilities, the authors introduced the concept of penalizing the objective function using two penalty terms for measuring overcapacity and over-duration respectively. In order to prevent cycling, a vertex that was moved from one route to another at iteration t was prohibited from being reinserted in the first route until iteration $t + \beta$ (prohibition of reverse moves), where β was a fixed integer or a randomly drawn one from a specific interval.

In this paper a simple and effective variant of the threshold accepting algorithm, called Backtracking Adaptive Threshold Accepting (BATA), for solving the OVRP is presented. The basic innovation of the BATA method is the employment of a

backtracking policy for the threshold value when no acceptances of feasible solutions are encountered during the search process. According to the proposed policy, the threshold is not only reduced in a monotonic fashion during the optimisation procedure, but it may incorporate an occasional increase in its value, called backtracking. Adopting this non-monotonic schedule results in an oscillating strategy that achieves a dynamic balance between diversification (when the threshold value is high enough) and intensification (when the threshold value is low) of the search process. Our major contribution is to show that BATA, this lean and parsimonious method can outperform more complicated strategies, such as tabu search, for the solution of the OVRP.

The remainder of the paper is organized as follows: Section 2 presents BATA, the proposed annealing-based method. In Section 3, the computational results of BATA on well-known benchmark problems extracted from the literature [4] are provided and compared with previously published work. Finally, Section 4 summarises our conclusions and offers pointers to further research.

2. The proposed threshold accepting approach

In the early 70s, the most popular heuristics developed for solving *NP*-hard problems, such as the VRP, were based on local search improvement techniques. Local search is an iterative procedure that, starting from an initial feasible solution, progressively improves it by applying a series of local modifications called *moves*. As the local search evolves, the set of *moves* that can be applied to a current solution s, define a set of neighbouring solutions, $N(s)$, which is a subset of the search space (*i.e.*, all possible solutions than can be visited during the search) consisting of solutions generated by applying a single transformation to s. According to local search rationale, at each iteration, the search proceeds to an improving neighbouring feasible solution until it arrives at a local optimum of unknown quality.

Heuristic methods were the heart of traditional optimisation approaches until the development of the Simulated Annealing (SA) metaheuristic [14] in 1983. SA is inspired by the physical annealing process emanating in statistical mechanics. The approach uses local search while offering the possibility of accepting, in a controlled manner, worse solutions. This feature allows SA to escape from low quality local optima. More precisely, at each iteration of SA, a neighbour $s' \in N(s)$ of the current solution s is generated stochastically and a decision is made whether s' will replace s. If s' is better than s, *i.e.*, $\Delta = c(s') - c(s) \leq 0$ (for a minimization problem), the search moves from s to s'; otherwise, the search proceeds to s' with a probability $e^{(-\Delta)/t}$. This probability depends on: a) the degree of the degradation Δ (the smaller the Δ, the greater the accepting probability), and b) a control parameter t called temperature (higher temperatures lead to higher accepting probabilities and *vice versa*). The temperature is controlled by a cooling schedule specifying how this parameter would be progressively reduced. Typically, SA stops when a fixed number of non-improving iterations is realized

within a temperature, or when a pre-specified number of iterations is reached. SA is proven to converge to an optimal solution in infinite computational time.

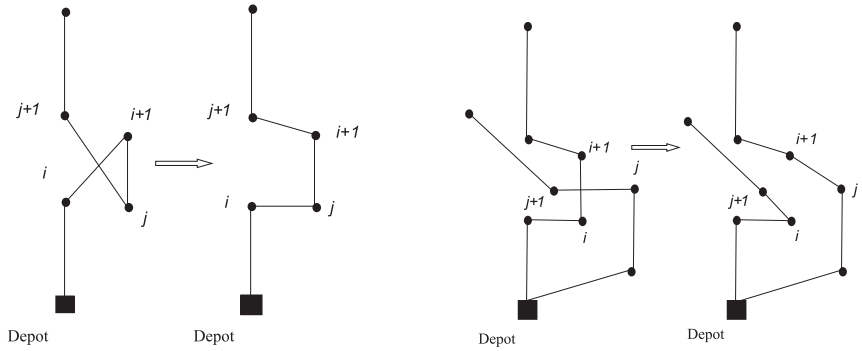
Threshold accepting (TA) [9] is a modification of SA, which eliminates the stochastic element in accepting worse solutions by introducing a deterministic threshold, denoted $T_h > 0$, and accepts a worse solution if $c(s') - c(s) \leq T_h$ (the inequality represents the *move acceptance criterion*). During the optimisation process the threshold level is gradually reduced, just like the temperature parameter in SA. As long as the value of T_h is high, the local search performed is not goal oriented, thus achieving high diversification and low intensification of the process. However, as the algorithm evolves and T_h is reduced, the balance between diversification and intensification changes until the typical threshold accepting algorithm behaves nearly like a local descend search method.

The key advantages of TA are its simple structure, general applicability and computational effectiveness on different combinatorial optimization problems [3]. According to Golden *et al.* [12], these characteristics should belong to any VRP-heuristic in the future. Following this rationale, the TA-based approach for solving the OVRP, named Backtracking Adaptive Threshold Accepting (BATA) method includes the following steps:

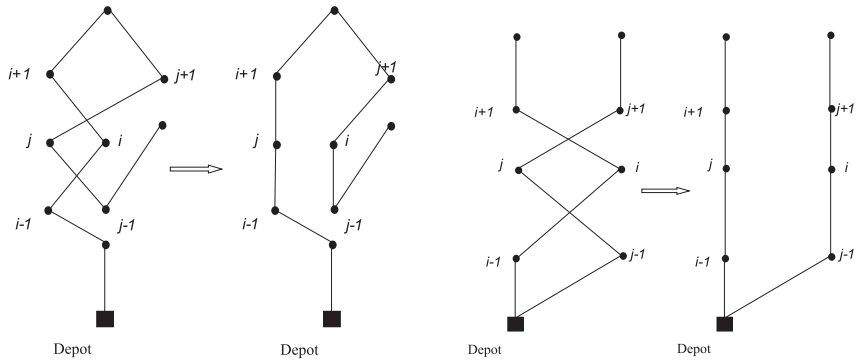
Step 1 (Initialisation)**.** Produce an initial feasible solution by dispatching one vehicle to each customer. Select an initial threshold value T_{ho}.

Step 2 (Local search)**.** Begin with the initial solution s of Step 1 and perform a blend of moves by exchanging edges (2-opt move) [8] or nodes (1-1 and 1-0 Exchange moves) [19] to reach a solution $s' \in N(s)$ (*i.e.*, in the neighbourhood of the current one). Moves are specified as follows:

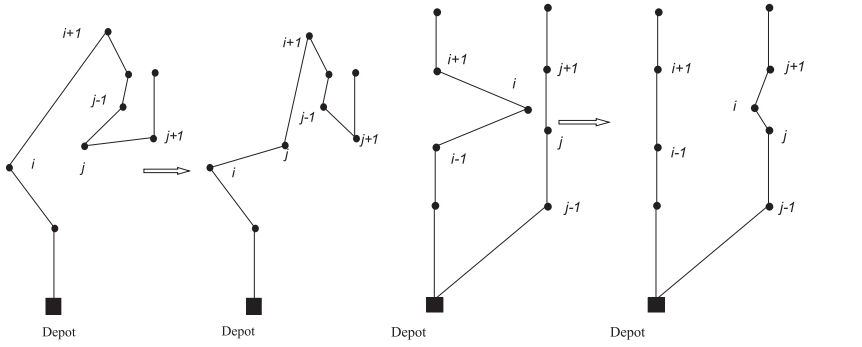
- 2-Opt move. Suppose a single route consists of the following ordered set of nodes (depot, $1, 2, 3, \ldots, k$), where k is the last customer to be serviced. Let $\{(i, i{+}1), (j, j{+}1)\}$ be a set of two edges belonging to this route that form a criss-cross. The 2-Opt move eliminates the criss-cross 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 case of multiple routes, edges $(i, i{+}1)$, and $(j, j{+}1)$ belong to different routes and the 2-Opt move is applied exactly as in the case of a single route.
- 1-1 Exchange move, which swaps two nodes from the same route. If the initial route consists of the nodes (depot, $\ldots$, i-1, i, $i{+}1$, $\ldots$, j-1, j, j-1, $\ldots$, k), the improved one is constructed as (depot, $\ldots$, i-1, j, $i{+}1, \ldots$, j-1, i, $j{+}1, \ldots$, k). The same procedure is applied in the case of multiple routes, but swapping of nodes takes place between different routes.
- 1-0 Exchange move, which transfers a node from its position in one route to another position in either the same or a different route. Consequently, while the initial route is (depot, $\ldots$, i, $i{+}1, \ldots$, j-1, j, $j{+}1, \ldots$, k), the improved one is (depot, $\ldots$, i, j, $i{+}1$, $\ldots$, j-1, $j{+}1, \ldots$, k).



2-opt move



1-1 Exchange move



1-0 Exchange move

FIGURE 1. The local search moves performed within BATA.

The customers involved in the implementation of the above moves and the type of moves are selected stochastically. The local search moves described above are demonstrated in Figure 1.

Step 3 (New acceptances based on the move acceptance criterion)**.**

Step 3(a). Generate a neighbour solution s' from the current solution s by applying one of the local search moves shown in Figure 1.
Step 3(b). Decide on the acceptance of the proposed solution s' based on the *move acceptance criterion* expressed as $c(s') - c(s) \leq T_h$, where T_h is the current threshold value (equal to T_{h0} at the first iteration).

- If $c(s') - c(s) \leq T_h$, the proposed solution s' is accepted; thus $s = s'$. Then the threshold value is lowered as follows: $T_{hNEW} = T_r{}^*T_h$, $T_h = T_{hNEW}$, where T_r is a parameter denoting the threshold reduction rate. If $c(s) < c(s_{best})$, then $s_{best} = s$.
- If $c(s') - c(s) > T_h$, then the proposed solution s' is not accepted while the threshold value T_h is raised (or backtracked) as follows: $T_{hNEW} = T_h/T_r$ $(T_h/T_r - T_h)^*(1\text{-}T_b)$, $T_h = T_{hNEW}$, where T_b is the percentage of threshold backtracking. The increased threshold value (after backtracking) must *always* be *smaller* than the one before the backtrack step. In this way, BATA attempts to find the smallest possible value for the threshold so as to achieve accepted feasible moves without causing excessive computational effort to the optimisation procedure.

– Go to Step 3(a) using the updated threshold value and repeat Step 3(a) and Step 3(b).

Step 4 (stopping criterion)**.** The BATA optimization process is stopped when although the value of the threshold T_h has been backtracked, no feasible move (which gives the proposed neighbour solution s') can satisfy the move acceptance criterion $c(s') - c(s) \leq T_h$ for a number of consecutive iterations. Finally, report the best solution found.

The following example illustrates the BATA mechanism for the OVRP: Let $T_r = 0.9999$, $T_b = 0.98$ and $T_{ho} = 48$ and suppose there are feasible moves satisfying the move acceptance criteria in Step 3. Hence, the value of the threshold is lowered and it is set to:

$$T_{hNEW} = T_{ho}^* T_r = 48^*0.9999 = 47.9952, \text{ and } T_h = T_{hNEW}.$$

BATA

Step 1. Produce an initial feasible solution and select an initial threshold value

Step 2. Conduct Local Search by employing a blend of local search moves

Step 3. **a.** Generate a neighbour solution s' of the current solution s
b. Check if $c(s')$-$c(s) \leq T_h$
If YES, accept s' and set $s'=s$. Then, reduce the threshold value. Check if $c(s)<c(s_{best})$, then $s_{best}=s$.
If NO, reject s'. Then, raise the threshold value.
Repeat Step 3a and Step 3b

Step 4. Report the best solution found.

FIGURE 2. The Backtracking Adaptive Threshold Accepting method.

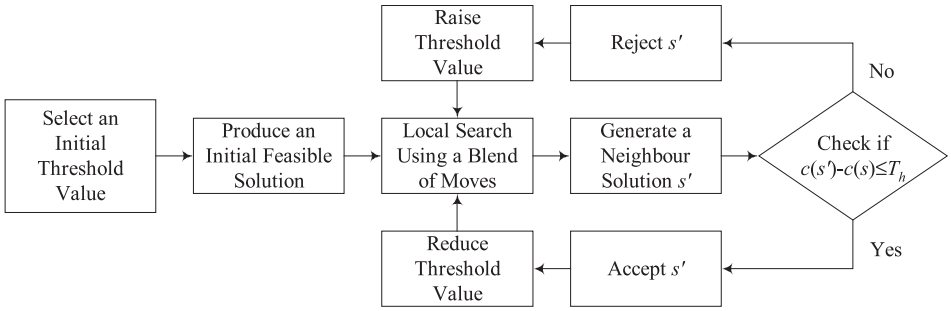


FIGURE 3. BATA Flowchart.

If no feasible move satisfies the move acceptance criterion for $T_h = 47.9952$, then T_h must be backtracked to a value "close" to its previous one and specifically to the value of:

$$T_{hNEW} = T_h/T_r(T_h/T_r - T_h)^*(1 - T_b) = 47.9952/0.9999 \\ - (47.9952/0.9999 - 47.9952)^*(1 - 0.98) = 47.9999.$$

Obviously, the increased value of the threshold is *always smaller* than the one before the backtracking step, since:

$$T_{ho} = 48 > 47.9999 > T_{hNEW} = 47.9995.$$

The main steps of BATA are summarized in Figure 2.

To graphically illustrate the mechanics of the BATA method, we provide in Figure 3 a flowchart that shows all the steps and decisions involved within the proposed algorithm as well as their natural flow and interrelationships.

3. Analysis of the BATA parameters

The majority of metaheuristics for the vehicle routing problem include several parameters that need to be set before the algorithm is executed. This is indeed the case with BATA, which employs four such parameters, namely: Initial Threshold Value (T_{ho}), Maximum number of feasible moves satisfying the move acceptance criterion within the Inner Loop (*Maxup*), Percentage of threshold backtracking based on the previous iteration (T_b), and Rate of threshold reduction (T_r). A common and well accepted way to approach the interrelationships between the parameters of a metaheuristic, and their effect to the evolution process and the resulting solution quality, is the empirical analysis of "hard" problem instances. Thus, we concentrated on the analysis of the BATA parameters based upon the large scale instance of Christofides *et al.* [4] (Problem 5, including 200 customers).

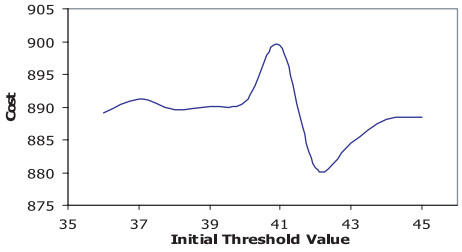
The process we employed for the analysis is as follows: first we defined a set of initial parameter values *ad hoc*. Then we used a sequential approach to fine tune one parameter on a reasonable value range, and repeated the process for the remaining ones keeping the previously reached best values for parameters already examined. For the problem instance under considerations, the initial values were chosen as follows: $Maxup = 25$, $T_b = 0.98$ and $T_r = 0.99999$. Based on these values, we examined the effect of varying the values of T_{ho} in the range [36 ... 45]. The results are shown in Figure 4a. Subsequently, as shown in Figure 4b, we kept the value of T_{ho} that provided the minimum cost ($T_{ho} = 42$), and varied value of *Maxup* in the range of [20, ...,30].

The parameter-tuning procedure was carried on in the same manner trying to reach the best values of T_b, and T_r in the ranges $T_b \in [0.90, \ldots, 0.99]$ and $T_r \in [0.9999, \ldots, 0.99999]$, respectively. The results are shown in Figures 4c and 4d. A key observation from the curves that relate parameters value and solution quality is that no specific trend can be established. This is usually the case with metaheuristics employing a stochastic evolution process.

4. Computational results

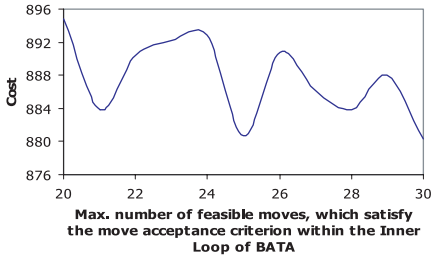
The proposed algorithm, BATA, for the OVRP was coded in MS Visual C++ 6.0 and run on a Pentium II/400-128MB RAM. The computational experiments were performed as follows:

Parameters	Values
T_{ho}	[36…45]
Maxup	25
T_b	0.98
T_r	0.99999



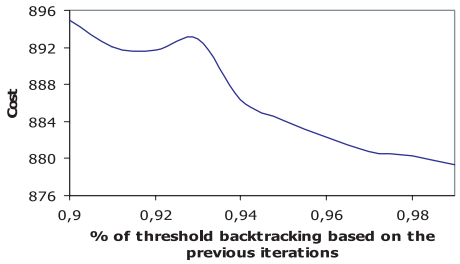
(a) Changing the values of T_{ho}

Parameters	V alues
T_{ho}	42
Maxup	[20,…,30]
T_b	0.98
T_r	0.99999



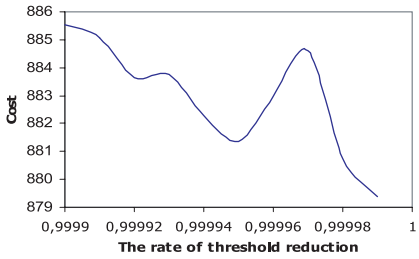
(b) Changing the values of *Maxup*

Parameters	V alues
T_{ho}	42
Maxup	30
T_b	[0.9…0.99]
T_r	0.99999



(c) Changing the values of T_b

Parameters	V alues
T_{ho}	42
Maxup	30
T_b	0.99
T_r	[0.9999….0.99999]



(d) Changing the values of T_r

FIGURE 4. Sensitivity analysis of BATA parameters for Christofides *et al.* [4] problem 5.

TABLE 1. Standard parameter setting of BATA on Christofides *et al.* [4] data sets.

BATA Parameter	Description	Value
Maxup	Maximum number of feasible *moves* satisfying *the move acceptance criterion* within the Inner Loop	30
T_{ho}	Initial threshold value	42
T_r	Threshold reduction rate	0.99999
T_b	Percentage of threshold backtracking based on the previous iteration	0.99

4.1. Benchmark problems

BATA was tested on the well-known benchmark problems of Christofides *et al.* [4]. These problems include between 50 and 199 nodes (customers) as well as the depot, and have capacity restrictions but no route length constraints or node service times. Problems 1–5 contain customers randomly located over a square, while problems 11–12 have clustered customers. Euclidean distances are used in all problems, and all computations were performed with full double precision.

4.2. Parameter setting

The parameter setting of BATA resulting from the analysis of Section 3 was deemed as the "standard" parameter value set and was applied to solve all the benchmark problems of Christofides *et al.* [4]. The "standard" parameter values are shown in Table 1. Note that these values represent the best combination of parameters for problem 5.

4.3. Results and comparison with previous methods

Table 2 compares the results obtained by BATA, using the standard parameter setting of Table 1, to those derived using the OVRP solution approach of Sariklis and Powell heuristic [15], and the tabu search metaheuristic of Brandao [2]. Close examination of the results shows that BATA *improves upon five out of seven solutions* generated by other two approaches and provides higher quality solutions for problems 1, 2, 5, 11 and 12. Thus, for the data tested, the proposed algorithm consistently outperforms the Sariklis and Powell heuristic [15], improving the mean of the solutions on the examined benchmark problems by 39.15%, as well as the Brandao tabu search method [2] by 0.92%, even though tabu search is generally considered more effective than annealing-based metaheuristics for solving all variants of the method [7].

TABLE 2. Comparison between literature methods and the BATA.

Problem	*Number of customers*	*Sariklis & Powell* [15]	*Brandao* [2]	*BATA*
1	50	488.2	416.06	412.96
2	75	795.3	567.14	564.06
3	100	815.0	640.42	642.42
4	150	1034.1	734.72	736.89
5	199	1349.7	914.51	879.37
11	120	828.3	683.02	679.60
12	100	882.3	534.40	534.24
Mean		**884.7**	**641.47**	**635.65**
Percentage Above BATA Mean		**+39.15%**	**+0.92%**	

Italics indicate the problems where BATA outperforms other approaches for OVRP.

TABLE 3. CPU time (seconds) for BATA and the other literature methods.

Computational Platform		*Pentium 133 MHz*	*Pentium III 500 MHz*	*Pentium II 400 MHz*
Problem	*Number of customers*	*Sariklis & Powell heuristic*	*Brandao metaheuristic*	*BATA metaheuristic*
1	50	0.22	17.3	38.62
2	75	0.16	40.2	68.89
3	100	0.94	27.3	56.54
4	150	0.88	40.7	81.69
5	199	2.20	47.6	98.13
11	120	1.54	21.3	37.67
12	100	0.76	65.1	84.54

Table 3 reports the computational times of BATA and of the other two methods developed for the OVRP on various platforms [2, 15]. Regarding the BATA CPU times, despite being quite satisfactory, a fair comparison in terms of computational efficiency is rather difficult and probably not useful, due to the several factors that affect the computational speed, such as the method design, the data structures, coding skills, compiler and hardware.

Finally, Table 4 presents all the routes of the best BATA solutions for the problems of Christofides *et al.* [4] that we examined.

TABLE 4. Routes of best BATA solutions on Christofides *et al.* [4] problem instances.

Problem 1 - 50 customers

Route Index	Sequence of Nodes												
1	0	11	38	9	50	16	2	29	21	34	30	0	
2	0	6	14	25	13	41	19	42	40	0			
3	0	46	12	5	49	10	39	0					
4	0	47	18	4	17	37	44	15	45	33	0		
5	0	27	48	8	26	7	23	24	43	0			
6	0	32	1	22	31	28	3	20	35	36	0		

Total Distance = 412.96

Problem 2 - 75 customers

Route Index	Sequence of Nodes												
1	0	12	40	9	39	72	31	0					
2	0	45	29	5	37	20	70	60	71	36	69	0	
3	0	17	3	44	32	50	18	25	55	0			
4	0	4	30	48	47	21	61	0					
5	0	67	46	8	19	54	0						
6	0	75	68	2	74	28	62	22	0				
7	0	34	52	27	13	57	15	0					
8	0	7	35	53	11	14	59	0					
9	0	33	73	1	43	41	42	64	0				
10	0	6	51	16	63	23	56	49	24	0			
11	0	26	58	10	38	65	66	0					

Total Distance = 564.06

Problem 3 - 100 customers

Route Index	Sequence of Nodes																		
1	0	15	43	42	87	2	57	41	22	75	74	72	73	21	40	58	53	0	
2	0	62	10	70	30	20	51	9	81	33	50	1	69	27	0				
3	0	67	39	23	56	4	25	55	54	24	29	77	76	28	0				
4	0	64	49	36	47	46	8	45	17	84	5	60	83	18	89	0			
5	0	38	14	44	91	100	98	37	92	97	95	94	13	0					
6	0	86	16	61	85	93	59	99	96	6	0								
7	0	32	90	63	11	19	48	82	7	88	31	52	0						
8	0	65	66	71	35	34	78	79	3	68	80	12	26	0					

Total Distance = 642.42

Problem 4 – 150 customers

Route Index	Sequence of Nodes																		
1	0	67	23	56	39	139	4	110	149	26	105	53	0						
2	0	38	140	44	119	14	142	42	144	87	137	13	112	0					
3	0	64	49	143	36	47	124	48	82	7	106	52	146	0					
4	0	123	19	107	11	62	148	88	31	127	27	0							
5	0	51	103	9	120	81	33	102	50	111	28	0							
6	0	46	8	114	125	45	84	5	118	60	83	18	89	0					
7	0	43	15	57	2	115	145	41	22	133	75	74	72	73	21	40	58	0	
8	0	25	55	130	54	134	24	29	121	68	150	80	109	12	138	0			
9	0	17	113	61	85	93	59	104	99	96	6	147	0						
10	0	10	108	126	63	90	32	131	128	20	30	122	70	101	1	69	132	0	
11	0	66	71	65	136	35	135	34	78	129	79	3	77	116	76	0			

Total Distance = 736.89

TABLE 4. *Continued.*

Problem 5 - 200 customers

Route Index	Sequence of Nodes																
1	0	100	98	37	151	92	97	117	95	94	183	112	156	0			
2	0	23	186	56	197	72	198	180	105	53	0						
3	0	11	175	107	19	123	182	7	194	106	153	52	146	0			
4	0	159	62	148	88	31	190	127	167	27	0						
5	0	170	25	55	165	130	54	177	150	80	68	116	184	28	0		
6	0	46	174	45	125	8	114	199	83	60	118	166	89	0			
7	0	43	15	145	41	22	133	75	74	171	73	21	40	58	152	0	
8	0	164	34	78	169	121	29	24	163	134	109	12	154	138	0		
9	0	17	113	86	141	16	61	173	84	5	147	0					
10	0	192	119	44	191	91	193	85	93	59	104	99	96	6	0		
11	0	67	39	187	139	155	4	110	179	195	149	26	0				
12	0	38	140	14	142	42	172	87	144	57	178	115	2	137	13	0	
13	0	131	32	181	63	126	90	108	10	189	70	101	162	69	132	0	
14	0	66	188	20	128	160	30	122	1	176	111	0					
15	0	9	120	81	185	79	129	3	158	77	196	76	0				
16	0	64	49	143	36	47	168	124	48	82	18	0					
Total Distance = 879.37																	

Problem 11 – 120 customers

Route Index	Sequence of Nodes																					
1	0	8	12	13	14	15	11	10	9	7	6	5	4	3	1	2	88	0				
2	0	108	118	18	114	90	91	89	92	87	86	111	82	0								
3	0	103	104	99	100	116	98	110	115	97	94	93	96	101	102	0						
4	0	51	50	49	47	46	44	41	37	38	39	42	48	45	43	40	95	0				
5	0	56	55	58	60	63	66	64	62	61	65	59	57	54	53	52	105	0				
6	0	83	113	117	84	85	112	81	119	0												
7	0	120	0																			
8	0	68	73	76	77	79	80	78	75	72	74	71	70	69	67	107	106	0				
9	0	29	32	28	35	36	34	31	30	33	27	24	22	25	19	16	17	20	23	26	21	109	0
Total Distance = 679.60																						

Problem 12 - 100 customers

Route Index	Sequence of Nodes																
1	0	24	25	27	29	30	28	26	23	22	21	20	0				
2	0	72	61	64	68	69	66	62	74	63	65	67	0				
3	0	11	9	8	6	7	5	3	4	2	1	75	0				
4	0	40	41	42	44	45	46	48	51	50	52	49	47	43		0	
5	0	12	14	16	15	19	18	17	13	10	0						
6	0	80	79	77	73	70	71	76	78	81	86	87	90	0			
7	0	82	83	84	85	88	89	91	0								
8	0	92	93	94	95	96	97	100	99	98	0						
9	0	31	35	37	38	39	36	34	33	32	0						
10	0	60	58	56	53	54	55	57	59	0							
Total Distance = 534.24																	

5. CONCLUSIONS

In this paper we have proposed a metaheuristic method, called Backtracking Adaptive Threshold Accepting (BATA) for solving the Open Vehicle Routing Problem (OVRP). The OVRP is of particular importance for fleet planning as it allows companies to adopt the practice of hiring a fleet of vehicles to service their customers and assigning these vehicles to open routes (*i.e.*, routes that do not close at the depot). The fundamental contribution of BATA is the employment of the backtracking policy of the threshold value when no acceptances of feasible solutions occur during the search process. The excellent performance of BATA and its

simplicity shows that it can be employed (without demanding high coding skills) as a powerful tool for effective fleet planning in real life problems.

In terms of future research directions, as an important number of industries model their distribution operations as the OVRP (since they do not own a vehicle fleet), we strongly believe that the goal should be to design simple (*i.e.*, containing few parameters) and effective (*i.e.*, resulting in high quality solutions) metaheuristics for solving this interesting and practical distribution problem.

Acknowledgement. The authors are indebted to the Editor-in-Chief Professor Lemaire and the anonymous reviewers for their useful comments and suggestions that helped improve the manuscript. This work was partially supported by the General Secretariat of Research and Technology under grant NM-67.

References

[1] L. Bodin, B. Golden, A. Assad and M. Ball, Routing and scheduling of vehicles and crews: the state of the art. *Comput. Oper. Res.* **10** (1983) 63–211.

[2] J. Brandão, A tabu search algorithm for the Open Vehicle Routing Problem. *Eur. J. Oper. Res.* **157** (2004) 552–564.

[3] O. Bräysy, J. Berger, M. Barkaoui and W. Dullaert, A threshold accepting metaheuristic for the Vehicle Routing Problem with Time Windows. *Central Eur. J. Oper. Res.* **11** (2003) 369–387.

[4] N. Christofides, A. Mingozzi and P. Toth, The vehicle routing problem, in *Combinatorial Optimization*, edited by N. Christofides, A. Mingozzi, P. Toth and C. Sandi. Wiley, Chichester (1979) 315–338.

[5] G. Clarke and J.W. Wright, Scheduling of vehicles from a central depot to a number of delivery points. *Oper. Res.* **12** (1964) 568–589.

[6] J.-F. Cordeau, M. Gendreau, A. Hertz, G. Laporte and J.S. Sormany, New Heuristics for the Vehicle Routing Problem, in *Logistics Systems: Design and Optimization*, edited by A. Langevin et D. Riopel. Kluwer (to appear).

[7] J.-F. Cordeau, M. Gendreau, G. Laporte, J.-Y. Potvin and F. Semet, A guide to vehicle routing heuristics. *J. Oper. Res. Soc.* **53** (2002) 512–522.

[8] G. Croes, A method for solving traveling salesman problems. *Oper. Res.* **6** (1958) 791–812.

[9] G. Dueck and T. Scheuer, Threshold accepting: a general purpose algorithm appearing superior to simulated annealing. *J. Comput. Phys.* **90** (1990) 161–175.

[10] M. Gendreau, A. Hertz and G. Laporte, A tabu search heuristic for the vehicle routing problem. *Manage. Sci.* **40** (1994) 1276–1290.

[11] M. Gendreau, A. Hertz and G. Laporte, New insertion and postoptimization procedures for the traveling salesman problem. *Oper. Res.* **40** (1992) 1086–1094.

[12] B.L. Golden, E.A. Wasil, J.P. Kelly and I.-M. Chao, *The impact of metaheuristics on solving the vehicle routing problem: Algorithms, problem sets, and computational results*, edited by T.G. Crainic and G. Laporte. Fleet Management and Logistics, Kluwer Academic Publishers, Boston (1998) 33–56.

[13] B. Golden and A.A. Assad, *Vehicle Routing: Methods and Studies*. Elsevier Science Publishers, Amsterdam (1988).

[14] S. Kirkpatrick, C.D. Gelatt Jr. and M.P. Vecchi, Optimization by simulated annealing. *Science* **220** (1983) 671–680.

[15] D. Sariklis and S. Powell, A heuristic method for the open vehicle routing problem. *J. Oper. Res. Soc.* **51** (2000) 564–573.
[16] M. Syslo, N. Deo and J. Kowaklik, *Discrete Optimization Algoritms with Pascal Programs*. Prentice Hall, Inc. (1983).
[17] C.D. Tarantilis, C.T. Kiranoudis and V.S. Vassiliadis, A list based threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem. *J. Oper. Res. Soc.* **54** (2003) 65–71.
[18] C.D. Tarantilis and C.T. Kiranoudis, BoneRoute: An adaptive memory-based method for effective fleet management. *Ann. Oper. Res.* **115** (2002) 227–241.
[19] C.D.J. Waters, A solution procedure for the vehicle scheduling problem based on iterative route improvement. *J. Oper. Res. Soc.* **38** (1987) 833–839.