A Modified Osman's Simulated Annealing and Tabu Search Algorithm for the Vehicle Routing Problem

Caston Sigauke[a] and Habib M. Talukder[b]

Abstract

The basic vehicle routing problem is concerned with finding a set of routes for a fleet of vehicles which have to service a specified number of retail outlets from a central depot. A vehicle is allowed a maximum of two trips a day. In our analysis a vehicle delivers and collects empty containers. For the empty containers to be collected in a specified route total volume should not exceed the vehicle capacity. Similarly, the total quantity demanded on that route should not exceed the vehicle capacity. Our objective is to design a set of least cost vehicle routes for a given set of customer requirements.

Introduction

The basic vehicle routing problem is concerned with finding a set of routes for a fleet of *m* identical vehicles which have to service *n* customers from a central depot v_o, with $v_1, v_2, ..., v_n$ representing customers. The vehicle routing problem has been studied extensively for the past thirty or so years. It is a hard combinatorial optimisation problem. As such only relatively small vehicle routing problems can be solved to optimality. However there have been exceptional cases where large problems have been solved to optimality. Current researchers are concentrating on the development of approximation algorithms, heuristics and metaheuristics. Gendreau, Laporte and Hertz (1994) developed a tabu search heuristic for the vehicle routing problem with capacity and route length restrictions. Barnes and Carlton (1995) presented a reactive tabu search approach to the vehicle routing problem with time windows. Rochat and Taillard (1995) developed a probabilistic tabu search technique to diversify, intensify and parallelize almost any local search for almost any vehicle routing problem. Osman (1993) developed a hybrid simulated annealing and tabu search algorithm for the vehicle routing problem. In their studies they considered the single trip case and that the vehicles were of the same capacity.

In our study we considered the multi-trip case with vehicles of different capacities. A given vehicle *j* has capacity c_j. A vehicle is allowed a maximum of two trips a day. A non-negative demand quantity $d_i (i = 1,2,...,n)$ at each retail outlet is known in advance. We have also considered that a vehicle delivers the product and collects empty containers. The number of empty containers $e_i (i = 1,2,...,n)$ at each retail outlet is known in advance. For the empty containers to be collected in a specified route, the total volume of empty containers to be collected together with the load to be delivered at each customer should not exceed the vehicle capacity. A non-negative travel time c_{ij} is associated with every arc (v_i, v_j) between any two adjacent customers v_i and v_j. A non-negative service time δ_i is associated with customer *i*. The main objective is to design a set of least cost vehicle routes for given customer requirements.

Background information

The company considered in this study is in the alcoholic beverage industry. Although the organisation is in the process of computerising its routing system, currently they are using manual methods for vehicle routing and scheduling.

[a]Department of Applied Mathematics, [b]Department of Applied Physics, National University of Science and Technology P.O. Box AC 939 Ascot Bulawayo Zimbabwe

The product is distributed to customers from two production plants and from several distribution centres known as depots. The lead-time for delivery from a depot is shorter than from a brewery. This is due to the fact that the delivery is from existing stock. But from a brewery we have to wait at times for the production to be completed.

However, delivery from a brewery is cheaper than delivery from a depot due to lower handling costs. In this study we have considered a case where the product is delivered to retail outlets from one of the organisation's breweries. As service reliability is considered to be a key factor for the company, management undertakes regular studies to ensure fast delivery to customers while keeping distribution costs as low as possible.

The customers or retail outlets are divided into five sales territories. The organisation uses two types of systems, i.e. the pre-sell and the driver-sell. The dominant system currently is the pre-sell. The driver–sell system was introduced recently and is in its infancy stages. In this study we concentrate only on the pre-sell distribution system. The sales department through phone calls receives orders. After the sales orders have been received, a summary of these sales orders is then sent to the distribution planners known as dispatchers, who then carry out the routing and scheduling of vehicles. The deadline for receipt of orders is midday. This leaves the afternoon for the distribution planners to design routes, carry out loading and prepare order-picking notes for the following day's work.

In general, the construction of vehicle routes is a complicated task due to complexities arising from fleet size, the number of customers, capacity constraints, vehicle access restrictions to some retail outlets, and the fact that such routes must be constructed in a relatively short period of time. This, therefore, calls for the use of cheap and efficient computerised packages for vehicle routing.

Currently, the distribution planners are using the cluster-first-route-second method to design their vehicle routes. A driver is then assigned a vehicle with a crew of four members. They then deliver the product to the retail outlets in the given cluster. The driver determines the sequencing of customers in the sales territory he is serving. A driver takes with him a route card on which he writes a summary of the retail outlets he has visited in his trip. In most cases when they deliver they also collect empty containers. This therefore makes it a deliver and pick up system. On average the turn around time is four and half-hours per trip. Drivers are expected to work for nine hours a day. Overtime is also allowed but a driver may not work for more than four hours overtime. A driver is therefore expected to have two trips a day except in situations when he has been assigned a rural trip. In this study we concentrate only on the multi-trip case in an urban setup.

We have modified Osman's hybrid simulated annealing and tabu search algorithm to find feasible vehicle routes for each of the sales territories of the organisation. The following modifications are made:

(a) Starting solution: our initial solution is a combination of the cluster-first-route-second and the nearest neighbour concept. The cluster-first-route-second heuristic is a two phase method in which we first assign customers to vehicles and then determine the sequence within each vehicle route.

(b) We consider vehicles of different capacities (in our study we considered three vehicle types).

(c) We consider a deliver and pick system (deliver the product and pick empty containers in the same trip).

(d) $n(k) = k\alpha$, where $n(k)$ is the number of iterations to be performed before the algorithm terminates, h is a parameter to be determined and $h \in (0,1)$ and α is the total number of feasible exchanges. In this study we have arbitrarily set $h = 0.5$. Studies are still underway to determine a way of coming up with the best value of h.

(e) Derivation of α is done in section 4.

Mathematical model

The following assumptions have been made:

- the vehicles used have different capacities,
- multi-trips are allowed,
- the deliver and pick up system has been considered,
- the pre-sell system is only investigated and
- delivery is done from a production plant.

Notation

The following notation has been used in the mathematical development of the problem.

N = the set of n customers, $N = \{1, 2, \ldots, n\}$.

d_i = the demand of the i^{th} customer, $i \in N$.

V = the set of m vehicles.

c_j = the capacity of the j^{th} vehicle, $j = 1,2,\ldots,m$.

R_p = the set of customers in route p.

S_p = subset of R_p.

$c_{j,k+1}$ = excess capacity of vehicle j to accommodate empties from k number of customers, $k = 0,1,2,\ldots,x$.

$c_{j,0}$ = excess capacity of vehicle j when leaving the depot.

x = number of customers in route p.

P = the set of p routes, $P = \{1,2,\ldots,p\}$

δ_i = the service time of customer $i \in N$.

e_i = the number of empty containers to be collected from customer i during a delivery trip.

c_{ij} = the travel time/distance from customer i to customer j, $c_{ij} = c_{ji}$ $\forall i, j \in N$

$C(R_p)$ = the total cost/length of an individual tour. This cost includes the travel times c_{ij} and the service times δ_i.

U = the pre-specified upper bound on the maximum tour length. This is determined by adding the normal working hours per day to the total allowable overtime per day and then divide by 2 to allow for multi-trips.

S = the feasible solution which is defined as $S = \{R_1, R_2, \ldots, R_m\}$.

$C(S)$ = the total sum of each individual tour length, $C(R_p), \forall p \in P$.

Objective function

Minimise $C(S) = \sum_{p \in P} C(R_p)$

That is minimise the total sum of each individual tour length, $C(R_p), \forall p \in P$

Constraints

(a) Time Constraint

$$C(R_p) = \sum (c_{ij} + \delta_i) \le U, \forall p \in P$$

That is the total time travelled and service time on route p should not exceed a pre-specified upper bound U.

(b) Capacity Constraints

(i) $$\sum_{i \in R_p} d_i - c_j \le 0, \forall p \in P$$

(ii) $$c_{j,0} = c_j - \Sigma d_i ,$$

(iii) $$c_{j,k+1} = d_i + s_k - e_i \ge 0, \quad k = 0,1,2,\ldots,x$$

Solution procedure

The algorithm proposed in this study uses a λ - interchange generation mechanism. This generation mechanism describes how a solution S can be altered to generate another neighbouring solution say S'. Osman and Christophides in [6] developed this mechanism. This generation mechanism works on the following basic principles:

We first select two routes p and q. Let $S_p \subset R_p$ and $S_q \subset R_q$. Then a λ-interchange between this pair, R_p and R_q consists of swapping the customers of S_p with those of S_q as long as this is feasible. The sizes of S_p and S_q should be in such a way that $|S_p| \leq \lambda$ and $|S_q| \leq \lambda$, where $|S_p|$ is the cardinality of S_p and λ is the maximum number of customers to be exchanged. This interchange process will result in new route sets, $R'_p = (R_p - S_p) \cup S_q$ and $R'_q = (R_q - S_q) \cup S_p$. S_p and S_q can either be empty which would mean that the interchange would only simply include shifting customers from one route to the other. This process results in a large number of combinations of route pairs and as a result choices of S_p and S_q are also usually large. Because of this problem the λ-interchange generation procedure is usually implemented with λ = 1 or 2. In our study, we have used λ = 1. From our study we derived the maximum number of feasible exchanges as follows:

Let $|R_1| = n_1, \ldots, |R_m| = n_m$, where $|R_m|$ is the number of customers in route m

Considering the pair (R_p, R_q)

(i) For the $(1,0)$ shift process we have at most n_p shifts

(ii) For the $(0,1)$ shift process we have at most n_q shifts

(iii) For the $(1,1)$ interchange we have at most $n_p n_q$ feasible exchanges.

(iv) Therefore the maximum number of feasible exchanges is given by

$$\alpha = n_p + n_q + n_p n_q$$

The algorithm is summarised below. However, we need to explain the search process

Search process

In order to develop our algorithm, we need a few more notation to explain the search process. We perform a test cycle of search over the neighbourhood *N(S)* of the initial solution without performing the exchanges in order to obtain the largest and smallest Δ_{max}, Δ_{min} change in objective function values and an estimate of the total number of feasible exchanges α.

Δ_{max} = largest change in objective function value in a test cycle of the search over the neighbourhood $N(S)$ of the initial solution.

Δ_{min} = smallest change in objective function value in a test cycle of the search over the neighbourhood $N(S)$ of the initial solution.

S = initial solution.

N (S) = neighbourhood of *S.*

T_s = starting temperature which is a control parameter. i.e. Δ_{max}

T_f = final temperature, i.e. Δ_{min}

T_k = temperature at iteration *k*.

T_r = temperature reset variable.

α = total number of feasible exchanges.

k = iteration *k*, k = 1 represents initialisation process.

S_b = best solution found so far.

C(S) = objective function value of solution *S.*

n(k) = number of iterations to be performed before the algorithm terminates.

<u>Simulated Annealing and Tabu Search Algorithm</u>

Step 1. Generate an initial solution *S* by the cluster-first- route-second heuristic.

Step 2. *Parameter Initialisation*

Set $T_s = \Delta_{max}$
$T_f = \Delta_{min}$
$T_r = T_s$
$T_k = T_s$
$S_b = S$
$k = 1$
$n(k) = 0.5\alpha$

Step 3. *Next solution*

Explore the neighbourhood of *S* using the 1-interchange scheme and select a solution $S' \in N(S)$ in ordered search. Compute Δ = $C(S') - C(S)$ according to the evaluation of the cost of a move, which is the insertion - deletion procedure.

Step4. *Comparison of Solutions*

If $\Delta \leq 0$ then S = S'
    Else
     Generate a random number $r \in U(0,1)$
       If $r \leq e^{\frac{\Delta}{T_k}}$ then S = S'
            Else retain *S*
      Endif
Endif

If $C(S') < C(S_b)$ then
    $S_b = S'$ and $T_b = T_k$ (the temperature at which the best solution is found)
    Else retain *S*.
Endif

Step 5. *Temperature update*

We update the temperature parameter according to:

(a) Normal decrement rule

$$T_{k+1} = \frac{T_k}{1+\beta_k T_k}, where$$

$$\beta_k = \frac{|T_s - T_f|}{\left(0.5n\alpha + n\sqrt{k}\right)T_s T_f}$$

or

(b) Occasional increment rule

If a cycle of search is completed without accepting any 1-interchange move, update as

$T_r = max \{\frac{T_r}{2}, T_b\}$ and set $T_k = T_r$

Set $k = k+1$

Step 6. *Termination test*

Stop if $k = n(k)$ or if there is a very small change in objective function values. Then report the best solution S_b , otherwise, go to step 3.

Computational Experience

The algorithm was tested on six routes of the company under study. Problem sizes ranged from 1 to 6 customers. Figure 1 shows the sequencing of customers in route 1.

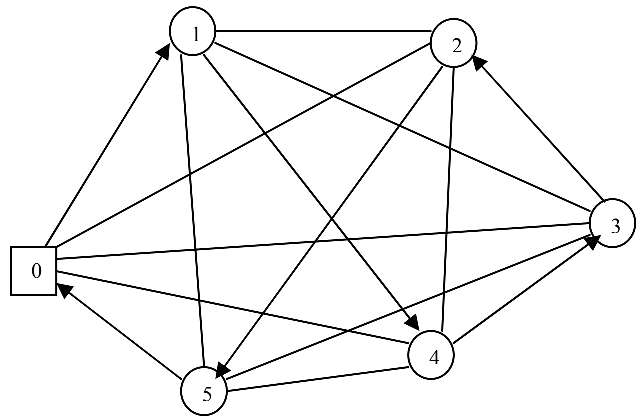

Figure 1. Sequencing of customers in route 1

The capacity ratio was calculated as follows:

$\rho = \frac{\sum (d_i + e_i)}{2c_j}$, Average Service Time per Crate (ASTC) $= \frac{\sum \delta_i}{|R_p|}$ and

$$\delta_i = (d_i + e_i)ASTC + 4$$

ρ = 0.972, R_1 = 0-1-4-3-2-5-0, Σd_i = 675, Σe_i = 686, $\Sigma \delta_i$ = 311, we choose type B (700 cases)

Then ΣC_{ij} = 10 +7 + 2 + 7 + 10 + 20 = 56, and $C(R_1)$ = 311 + 56 = 367

Summary results of the six routes are shown in the table 1.

We were able to reduce the number of vehicles by one and the average capacity ratio of the vehicles increased from 0.615 to 0.925. The total travel time was also reduced from 1083 minutes to 763 minutes.

Table 1: Summary results of six routes

P	Problem size (x)	Demand d_i	Collected containers e_i	Vehicle capacity c_j (crates)	Travel time(mins) c_{ij}	Service time(mins) δ_i	Route length(mins) $C(R_p)$	Capacity ratio ρ
1	5	675	686	700	56	311	367	0.972
2	6	640	579	650	53	286	339	0.938
3	3	547	795	700	35	299	334	0.839
4	2	365	613	650	40	216	256	0.752
5	1	0	968	800	10	210	220	0.605
6	3	600	634	650	40	276	316	0.949

Comparative Analysis and Conclusions

In this study, we modified Osman's hybrid simulated annealing and tabu search algorithm for the capacitated vehicle routing problem. The objective was to determine feasible routing patterns for an organisation in the alcoholic beverage industry. The details of the algorithm and the actual company data used in the calculations are given in Sigauke (2000). These showed that the algorithm outperformed the manual schedules currently being used in the organisation. Benefits to be derived from the use of this algorithm are as follows:

(i) Since fewer vehicles are required for the same amount of demand, it is easier to pull trucks off the road for repair and maintenance, which helps to extend vehicle life. Furthermore older vehicles can operate close to the depot to ensure easy and cost-effective recovery in case of breakdowns.

(ii) The developed algorithm is less dependent on one person for scheduling. If the dispatcher is away, the system can still run smoothly, when proper software is available. It is still being developed.

Acknowledgements

This paper is based on lengthy conversations with numerous individuals. We are grateful, in particular to the management of the organisation where this study was carried out and National University of Science and Technology for providing facilities.

References

[1] J. W. Barnes and W. B. Carlton, " Solving the Vehicle Routing Problem with Time Windows Using Reactive Tabu Search", presented at the Fall INFORMS National Meeting in New Orleans, Louisiana, 1995

[2] M. Gendreau, A. Hertz and G. Larporte, " A Tabu Serach Heuristic for the Vehicle Routing Problem", Management Science, Vol. 40, No. 10, pp 1276-1290,1994.

[3] F. Glover and M. Laguna, Tabu Search, Kluwer Academic Publishers, 1997.

[4] I.H. Osman., "Metastrategy .Simulated. Annealing and Tabu Search Algorithms for the Vehicle Routing Problem", Annals of Operations Research, No 41, pp 421-451,1993.

[5] Y. Rochat and E. Taillard, "Probabilistic Diversification and Intensification in Local Search for Vehicle Routing", Journal of Heuristics, Vol. 1, No. 1, pp 147-167, 1995.

[6] C. Sigauke, " Efficient use of vehicle routing in logistics and distribution management with a capacity constraint, an application of some neighbourhood search techniques", M.Sc. dissertation, NUST, 2000.