Int. J. Open Problems Compt. Math., Vol. 7, No. 4, December 2014
ISSN 2074-2827; Copyright ©ICSRS Publication, 2014
www.i-csrs.org

Applying Tabu Search in Finding an Efficient Solution for the OVRP

Abdel Rahman Hedar, Mohammed Abdallah

Faculty of Computers and Information, Assiut University, Assiut, 71526, Egypt
Faculty of Sciences, Assiut University, Assiut, 71526, Egypt
e-mail: mohammed_pg0838243@science.au.edu.eg

Abstract

In the open vehicle routing problem (OVRP), the target would be to reduce the amount of vehicles after reducing the whole distance (or time) travelled. Every route begins at the depot and comes to an end at a customer, going to many customers, every once, en route, without returning to the depot. The demand of every customer has to be completely fulfilled by a single vehicle. The total demand serviced by each vehicle has to not go over vehicle capacity. A highly effective tabu search for open vehicle routing called Three Strategies Tabu Search (TSTS) heuristic for this problem is suggested. The TSTS depends on three strategies MOVE, EXCHANGE and SWAP. Computational results on fourteen standard benchmark problem instances demonstrate that the suggested TSTS is comparable in terms of solution quality for the best performing published heuristics.

Keywords: *Open Vehicle Routing Problem, Tabu Search*

1 Introduction

The supply of products or the delivery of services is essential for both customers and contemporary business actions, considering the appropriate working costs make up a large portion of the total field expenses of a company. The problem condenses from a useful point of view once the vehicle fleet is employed, which is, vehicles do not amount to company property [23]. In these cases, efficient organizing is a crucial good results element with the functional effectiveness and the ensuing service level, given that non-company assets have the effect of the physical interface using final customer [21].

The open vehicle routing functional structure is confronted with a business that either does not own a vehicle fleet whatsoever, or its fleet is improper or insufficient in order to meet the demand of its customers [23]. Therefore, this company needs to deal with all or part of its distribution activities to external carriers. These contractors have their own vehicles, they pay their own vehicle costs (e.g. capital city, functioning, servicing and rebate), and they in addition usually think about a reimbursement type depending on mileage [15]. Anytime the company does not need the service provider or the vehicle back at the depot, the routes and then the vehicles must not range from the vehicle voyage after the last delivery (i.e. the returning voyage to the depot) which will increase more mileage towards the reparations type.

The allocation associated with supply features to third party logistic (3PL) suppliers is a advantageous company process for many corporations. As an example, whenever a business has its fleet and customer demand differs over time, the perfect solution is for the open vehicle routing problem (OVRP) will supply the appropriate combined possessed and hired vehicles [21]. Likewise, companies which have numerous deliveries experience have a similar kind of problem. Even though hiring vehicles is costlier for each device distance traveled (DT), numerous expenses, for instance funds, upkeep and devaluation expenses, tend not to occur [24, 25]. Standard real-life OVRP examples would be the home delivery associated with packages and newspapers [22]. Lately, [20] created a web-based choice support system for any real-life OVRP app in regards to the supply of lubrication goods. In all of the cases, the companies that do not work in the delivery corporation utilize their particular vehicles and do not go back to the depot, however the bill is founded on the total DT from the depot towards the last customer. Whenever many deliveries tend to be finished, the travel distance and period linked to every vehicle is logged and drivers are liberated to get back to the favored location, since this part of travel is not refunded. Moreover, the usual assumption is that the price of one more vehicle will certainly overbalance any traveling costs that might be saved by its use. The other demonstrates the tradeoff between the vehicle hiring cost and transport cost indicated in terms of DT.

The OVRP can be defined as follows: It is provided an undirected graph G=($\grave{V}$, E) in which $\grave{V}$=$(0, 1, ..., n)$ is the set of $(n+1)$ vertices and E is the set of edges. Vertex 0 symbolizes the depot, and the vertex set V=$V \setminus 0$ refers to n customers. A nonnegative cost d_{ij} is connected with each edge i, $j \in E$. Each customer $i \in V$ needs a supply of q_i units from depot 0 (we assume q_0=0), and set of m similar vehicles of capacity Q positioned at depot 0 is employed to provide the customers. A route is described as a minimum cost simple cycle in G moving from the depot 0 and so that the whole demand of the customers visited does not surpass the vehicle capacity Q. The objective of the OVRP would be to design essentially m routes in order that all customers are visited precisely once and the sum of the route costs is minimized. The OVRP is NP-hard as it is natural generalization of the traveling salesman problem (TSP). This paper evolves efficient tabu search (TSTS) for solving OVRP. It is worth noting that tabu search have been used on many various vehicle routing problems [4, 7, 8].

The remaining of the paper is arranged as follows: (Section 2) gives a thorough literature review on strategies suggested for the OVRP. (Section 3) shows how to formulate initial solution and fitness function. (Section 4) gives a brief description on Tabu Search (TS). (Section 5) elaborates the suggested hybrid TSTS and offers an in depth explanation of most algorithmic components. Computational experiments evaluating the evidence of strategy and the quality of the proposed method, plus a relative efficiency examination, tend to be shown in (Section 6). Finally, in (Section 7) conclusions are driven and suggestions for additional research are recommended.

2 Literature Review

Because of its broad applicability and excessive complexity, the OVRP possesses produced significant research considering each its modeling and solution features. Over the last ten years, tabu search, deterministic annealing, large neighborhood search and branch-and-cut, amongst other methods, have been successfully applied to the OVRP. Though optimal solutions can be had applying specific strategies, the computational time needed to solve properly large problem instances continues to be beyond reach. For that reason, the concentrate of most researchers is provided to the design of metaheuristic methods effective at generating premium quality around optimum solutions with sensible computational problems. Nevertheless, research on the large-scale data set of [15] is limited so far.

[3] are one of the primary to handle real-life apps which fit the OVRP functional framework. They look at an exhibit airmail supply problem having many aspect restrictions, for instance delivery and pickup time windows, capacity on the total route length, vehicle capacities, open routes along with features. Two routing problems (deliveries and pickups) are individually solved, while using the well-known savings heuristic of [6] .

[23] produced a two-phase structure heuristic, cluster first route second (CFRS), for the OVRP with capacitated vehicles and unrestricted route lengths. From the first phase, clusters of customers are created contemplating only the capacity of the vehicles. The clusters are consequently well balanced and enhanced by reassigning customers. In the second phase, open routes are designed through solving a minimum spanning tree problem (MSTP). Fines used to

modify the MSTP solution and to retrieve the feasibility.
[4] suggested a TS metaheuristic algorithm. Initial solutions are produced when using the closest neighbor heuristic and the K-tree technique. Every solution is presented to an unstringing and stringing strategy for route advancement. The suggested rendering utilizes the $1-0$ Transfer (insert) and the $1-1$ Exchange (swap) neighborhood structures [12]. A specific characteristic is which in-feasible intermediate solutions (in conditions of capacity and maximum route length) are also regarded. The objective function is penalized to manage the infeasibilities, while two penalty conditions for the over-capacity and over-duration are presented.
[19] produced an adaptive large neighbor-hood search (ALNS) strategy, which includes some big neighborhoods which contend to change the existing solution. A predefined list of N^- and N^+ providers are selected to remove and insert customers through the existing solution. At every iteration, the altered solution is approved if specific requirements identified by a simulated annealing (SA) [13] get better at construction are fulfilled. A adaptive layer randomly controls that neighborhood to chose (roulette wheel) having a prejudice to its past efficiency. For diversification, a noise function is utilized in all providers. Eventually, with regards to fleet size minimization, one more stage just before LNS is employed.
Finally, [14] produced a branch-and-cut solution way of the OVRP. When compared with additional precise methods suggested in the literature for the sealed and asymmetric capacitated VRP, many adjustments are presented concerning the integer programming formula, the reducing airplanes and the separating algorithms. Perfect solutions are developed for each small- and medium-scale OVRP instances, as the comparative difficulty between the open and closed route variations of the problem is furthermore examined.

3 Initial Solution

Any kind of tabu search algorithm demands an initial solution since that search procedure starts, specifically the meaning of the neighbourhood and the resultant first move. The majority of the research which implement tabu search, and particularly these specialized to VRP problems, spend little awareness of the initial solution. Commonly, the initial solution is insignificant, such as determining every customer to a route, or is acquired using a fast and well-known heuristic. We believe the primary reasons behind this are already the fact that the initial solution has hardly any impact on the quality of a final solution, and the requirement of obtaining a beginning solution rapidly, leaving behind all the work progress for the tabu search algorithm. This process has not prevented the accomplishment of very good results, since the tabu search is actually effective any time efficiently used. Nevertheless, even as will

be presented in the article, the initial solution could be computed rapidly and still provide an essential factor to improve a final solution.
In order to obtain an initial solution, we have developed a procedure, that constructs a feasible solution to the OVRP by finding a feasible assignment of customers to vehicles. The customers are assigned randomly to a selected route. The proposed procedure can be summarized as follows:
Step 1: Make a number of routes equal to number of vehicles (all vehicle have the same capacity, N_v and n represent the number of vehicles and the number of nodes sequentially).
Step 2: For $i = 1$ to $n - 1$, do
select a random node, assign it to current route. If the capacity of vehicle reached the maximum, assign node to a new route.
Step 3: For $i = 1$ to N_v, do
calculate length of each route.
Step 4: For $i = 1$ to N_v, do
select the first node in route and swap it by all next node, in each swap find length of route. Choose the shortest length of route.

3.1 Formulation of the Fitness Function

A fitness function F is a specific type of objective function which quantifies the optimality of solution. The shorter the route, the higher is the fitness value. Therefore, we designed the fitness function as follows :

$$F = TD \prod_{i=1}^{N_v} \max[1, (RQ/Q)], \qquad (1)$$

where TD is total length of route, RQ is all demand of nodes in route, and Q is the capacity of each vehicle i.

4 Tabu Search Optimization

Fred Glover suggested in 1986 anew strategy, that he referred to as tabu search, allowing hill climbing to conquer local optima. Actually, several components of this first tabu search suggestion, plus some components of later preparations, had been presented in [9], which includes short-term memory to avoid the change of current moves, and longer-term frequency memory to strengthen desirable elements. The essential process of tabu search is to follow the search when a local optimum is encountered by enabling non-improving moves; cycling back to formerly visited solutions is prohibited by the use of memories, called tabu lists, which record the current history of the search. The main notion to exploit info to guide the search may be from the informed search methods recommended in the late 1970s in the field of artificial intelligence ([17]). It is intriguing to note that, in 1986 also, Hansen proposed a method similar to tabu search, that he named steepest ascent/mildest descent([10]). It is additionally vital that you remark which Glover did not see tabu search as a appropriate heuristic, but instead being a metaheuristic, i.e. a general technique for driving and managing "inner" heuristics particularly customized for the problems available. The steps of TS [11]are

1. Choose an initial solution x_0. Set the Tabu List (TL) to be empty, and set the counter $k := 0$.

2. Generate neighborhood moves list $M(x_k) = \{\grave{x} : \grave{x} \in N(x_k)\ \}$, based on tabu restrictions, where $N(x_k)$ is a neighborhood of x_k.

3. Set x_{k+1} equal to the best trial solution in $M(x_k)$, and update TL.

4. If stopping conditions are satisfied, terminate. Otherwise, go to Step 2.

5 Three Strategies Tabu Search Algorithm

As earlier began, the initial solution attained by the construction heuristic identified in (section 3) is improved by means of TSTS method. As when it comes to every local search method, the solution search space is discovered by executing moves from current solution to the subsequent one. Two methods of each Move, Exchange and Swap are employed, in addition to intersection procedure.

5.1 Move Strategy :

$Method\ one$: suppose that customer x is in route R_m, and customer y in route R_n. Using the move strategy, customer x could be removed from route R_k and inserted following customer y in route R_n. After that we could have the new routes denoted as $\grave{R}_m = (0, ..., x-1, x+1, ...)$ and $\grave{R}_n = (0, ..., y, x, y+1, ...)$.
$Method\ two$: liked first method but the idea was how to choose the two customers. The chosen here was made according to the length between them thus we need to find the smallest length pair of customers so we take first customer from R_m and find the length between it and all customer in R_n and so on until last customer in first route was reached.

5.2 Exchange Strategy :

Method one : the method is described as exchange of customers between two different routes. For example we swap customers $x \in R_m$ and $y \in R_n$ position.
Method two : As we done in second method of move strategy to find the smallest length between two customer then swap position.

5.3 Swap Strategy :

Method one : suppose that two customers w and z are in route R_m. Using the swap strategy, the position of the two customers are swaped so if the route $R_m = (0, ..., w, ..., z, ...)$ then after swapping operation the route is $R_m = (0, ..., z, ..., w, ...)$.
Method two : suppose that three customers y, w and z are in route R_m. Using the swap strategy, find all probabilities of swapping these three customers. In each probability compute the total length of route then choose the smallest length of route. As example if $R_m = (0, ..., y, ..., w, ...z, ...)$ hence first probability may be $R_m = (0, ..., w, ..., y, ..., z, ...)$, second probability is $R_m = (0, ..., w, ..., z, ..., y, ...)$ and so on until last probability.

5.4 Intersection procedure :

In this procedure we attempt to reduce the length of routes in solution therefore, we apply two methods.
Method one (four customers): suppose that two customers x and $y \in R_m$ and other two customers w & $z \in R_n$, then we remove the intersect arcs between four customers by using the first method of exchange strategy.
Method two (six customers): It is the same as the first method but the difference is in the number of customers and the idea that how to choose them. We have three types through which we can choose, firstly three customers from each route, secondly two customers from route and four from the other and thirdly two customers from different the three routes.

5.5 Stopping Criteria

Because the algorithm is actually open-ended, therefore the stopping criteria are usually needed. It might run forever since the optimum is unidentified. Due to this problem, algorithm stops searching just after it finishes diversify on the repeating and we additionally limit the maximum iteration up to 1000 in order to avoid wasting time. This is due to, in case when there are a great number of repetitions, the algorithm might run forever until it finished diversify but also in the same time the solution are not enhanced after such number of iteration. Because the optimum is not known, therefore the maximum number of iteration is necessary.

6 Computational Experiments

The suggested method have been tested on the well-known bench-mark data sets of [5, 15]. Particularly, there are 14 problems denoted as $C1-C14$ coming from [5]. The set range in size from 50 to 199 customers, suppose Cartesian coordinates and Euclidean distances.

Parameter Setting and Tuning : The suggested solution strategy exposes three parameters, specifically P_m, P_e and P_s. This parameters specify the proportion of every strategy to occur inside range of solutions. The three parameters values modifications from 0 to 1, where P_m, P_e and P_s represent Move, Exchange and Swap strategies.

6.1 Computational Results

Computational results for the OVRP are ranked according to a hierarchical objective function. The primary objective is to minimize the total Nv, the secondary objective is to minimize the total traveled distance. However, one should note that these two objectives are often conflicting since the reduction of the total Nv may increase the total traveling distance. The best results are marked in bold indicate that, for the instance, the TSTS provided a new best result. The proposed algorithmic framework was coded in Matlab and executed on a Pentium i5 2.4 GHz computer system, with 6 GB of RAM, under Windows 7, for solving the sets of 14 problem, in total, VRP benchmark instances.

Table 1 : illustrates the characteristics of the fourteen problem instances of [5] where N and Q represent the number of customers and capacity of vehicle Sequentially.

Table 2 : summarizes the results obtained from the application of the proposed solution method, abbreviated as TSTS algorithm as the first method of the three strategies, on the problem instances of [5]. Furthermore, the detailed results of the best performing metaheuristic implementations from other authors are also provided, using the following abbreviations: SA and TS [18], genetic algorithm (GA) [2], scatter search algorithm combined by ACO (SS-ACO) [27], particle swarm intelligent (PSO) [1], genetic algorithm combined with particle swarm intelligent (GAPSO) [16] and (HEAS) [26] in addition to the Best Known Result (BKR). In addition to a multi-start version, where the

algorithm is repeated 10 times and the best solution is kept. The table demonstrates the efficiency and effect of the proposed solution method compared to the existing state-of-the-art of the OVRP. It produces two new best solutions.
Table 3 : summarizes the results obtained from the application of the proposed solution method, abbreviated as TSTS algorithm as second method of three strategies. The table is divided into three groups according to parameters setting. In the first group we set one parameter equals to 1 and other two parameters equal to 0 (i.e $P_s = 1$ and $P_e = P_m = 0$), but in the second group we set one parameter equal to 0.5 and other two parameters equal to 0.25 (i.e $P_s = 0.5$ and $P_e = P_m = 0.25$) and finally we set one parameter equals to 0.2 and other two parameters equal to 0.4 (i.e $P_s = 0.2$ and $P_e = P_m = 0.4$) as shown in the table. we mentioned that the solution match most of the 14 best known solutions but from three groups we investigated that the best known result is achieved when P_m has a large ratio which means move strategy produced most of the best solutions.
Table 4 : summarizes the results obtained from the application of the suggested solution method, abbreviated as TSTS algorithm as the second method of the three strategies after set the parameter P_m to take a high ratio in solution comparing to others two parameters P_e and P_s as shown in table. This method produced the best known results except instances $C5$.
Table 5 : show the results for first method of intersection procedure after sets the parameter P_m equal to 1 and the others two parameters P_e and P_s equal to 0. This method produced the best known results except three instances $C1$, $C5$ and $C11$.
Table 6 : show the results for second method of intersection procedure after sets the parameter P_m equal to 1 and the other two parameters P_e and P_s equal to 0. This method produced the best known results except four instances $C1$, $C3$, $C5$ and $C11$.
Table 7 : show the results for two methods of intersection procedure after set three parameter P_e, P_m and P_s equals to (1/3). This method produced the best known results for six only instances $C6$, $C8$, $C9$, $C10$, $C13$ and $C14$.
The above computational experiments illustrate how the suggested solution strategy is both effective and efficient in finding top quality solutions with sensible computational problems. The strategy appears to scale effectively based on the total number of customers. The latter additionally suggests how the parameter settings employed for the computational experiments provided a good compromise between solution quality. Finally, since these results acquired having fixed parameter settings overall problem instances, the robustness of the method is apparent.

Table 1: Benchmark data sets of [5]

Instance	N	Q	Nv
C1	50	160	5
C2	75	140	10
C3	100	200	8
C4	150	200	12
C5	199	200	16
C6	50	160	6
C7	75	140	10
C8	100	200	9
C9	150	200	13
C10	199	200	17
C11	120	200	7
C12	100	200	10
C13	120	200	11
C14	100	200	11

Table 2: Detailed results for the second methods TSTS

Instance	SA	TS	GA	SS-ACO	PSO	GAPSO	HEAS	BKR	TSTS
C1	528	524	524.61	524.61	524.61	524.61	524.61	524.61	**338.83**
C2	838	844	849.77	835.26	844.42	835.26	847.14	835.26	**638.15**
C3	829	835	840.72	830.14	829.4	826.14	712.36	826.14	1152.87
C4	1058	1052	1055.85	1038.2	1048.89	1028.42	1066.89	1028.42	1922.62
C5	1376	1354	1378.73	1307.18	1323.89	1294.21	1311.35	1291.45	3164.97
C6	555	555	560.29	559.12	555.43	555.43	555.43	555.43	**293.85**
C7	909	913	914.13	912.68	917.68	909.68	909.68	909.68	**579.51**
C8	866	866	872.82	869.34	867.01	865.94	865.94	865.94	1019.39
C9	1164	1188	1193.05	1179.4	1181.14	1163.41	1162.89	1162.55	1897.12
C10	1418	1422	1483.06	1410.26	1428.46	1397.51	1404.75	1395.85	2938.75
C11	1176	1042	1060.24	1044.12	1051.87	1042.11	1042.11	1042.11	2173
C12	826	819	877.8	824.31	819.56	819.56	840.64	819.56	1035.42
C13	1545	1547	1562.25	1556.52	1546.2	1544.57	1545.93	1541.14	1774.82
C14	890	866	872.34	870.26	866.37	866.37	866.37	866.37	1000.76

Table 3: Detailed results for three proposed strategies according to parameter setting

(P_s, P_m, P_e)	$(1,0,0)$	$(0,1,0)$	$(0,0,1)$	$(.5,.25,.25)$	$(.25,.25,.5)$	$(.25,.5,.25)$	$(.4,.2,.4)$	$(.4,.4,.2)$	$(.2,.4,.4)$
c1	833.20	587.90	1281.68	600.50	703.23	696.23	687.37	598.79	600.33
c2	1337.53	825.73	1882.41	839.69	909.84	827.70	899.90	976.00	870.43
c3	1588.03	852.92	2814.83	890.18	848.51	818.96	866.34	915.92	839.49
c4	2383.91	1063.17	3989.29	1077.46	1097.48	1122.80	1162.24	1123.95	1100.40
c5	3115.46	2164.52	5333.63	2903.77	3023.60	2929.22	3066.25	3027.64	3446.97
c6	849.92	**470.49**	1113.23	**463.14**	**450.43**	**486.88**	**441.43**	**441.90**	**438.01**
c7	1366.82	**808.79**	1836.10	963.64	1008.30	935.14	916.11	935.37	952.55
c8	1617.80	**812.41**	2562.67	**779.27**	**830.27**	**824.50**	**784.59**	**831.03**	**772.15**
c9	2395.35	**988.82**	3587.84	**803.25**	**985.37**	**1036.55**	**1047.41**	**1005.09**	**1036.29**
c10	3094.21	**1290.84**	4369.87	1475.70	**1385.45**	**1342.65**	1418.82	**1305.77**	**1342.03**
c11	2229.07	843.01	5389.57	1391.45	1620.15	1168.33	1566.96	1367.59	1422.27
c12	1923.73	**628.76**	3090.08	**693.50**	**665.62**	**681.40**	**718.11**	**628.24**	**657.03**
c13	2631.44	**847.99**	4846.54	**960.17**	**976.59**	**948.37**	**941.47**	**897.00**	**962.05**
c14	2064.10	**642.92**	3048.76	**600.14**	**633.84**	**614.39**	**673.81**	**620.36**	**656.80**

Table 4: Detailed results for Move strategy according to parameter setting

(P_m, P_e, P_s)	$(1,0,0)$	$(.9,.05,.05)$	$(.8,.1,.1)$	$(.7,.15,.15)$	$(.6,.2,.2)$	$(.5,.25,.25)$
c1	595.25	551.12	529.34	**516.85**	572.58	696.23
c2	836.71	809.22	905.57	960.03	**564.76**	827.7
c3	852.92	**783.91**	834.92	831.47	854.18	818.96
c4	1085	1065.28	1034.67	983.92	**946.94**	1122.8
c5	2252.13	2726.57	3220.42	3451.23	3135.09	2929.22
c6	470.49	445.87	**435.31**	474.54	443.81	486.88
c7	**814.57**	878.37	896.21	971.73	835.96	935.14
c8	812.41	**755.73**	779.21	775.1	804.08	824.5
c9	988.82	977.04	**973.75**	955.68	1022.8	1043.25
c10	**1290.84**	1335.93	1317.03	1402.14	1299.22	1393.79
c11	**851.46**	1152.84	1268.48	1194.45	1167.23	1168.33
c12	**628.76**	642.33	651.91	706.67	675.01	681.4
c13	**847.99**	953.29	**939.72**	**874.31**	**855.63**	948.37
c14	642.92	639.84	635.76	650.55	**611.59**	614.39

Table 5: Computational result for method one of intersection procedure

Instance	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11	c12	c13	c14
TSTS	646.97	**746.84**	**808.60**	**1020.80**	2172.30	**549.24**	**705.13**	**746.30**	**882.86**	**1150.60**	1185.50	**370.89**	**747.99**	**428.99**

Table 6: Computational result for method two of intersection procedure

Instance	TSTS (2, 2, 2)	TSTS (2, 4)	TSTS (3, 3)
c1	646.97	694.77	646.97
c2	834.12	**738.14**	839.49
c3	862.99	866.04	981.05
c4	1160.20	**1020.80**	1090.10
c5	2214.90	2172.30	2243.20
c6	685.04	**549.24**	649.97
c7	830.59	**705.13**	877.52
c8	**746.30**	**746.30**	**746.30**
c9	938.05	**882.86**	995.74
c10	**1122.50**	**1122.50**	**1122.50**
c11	1079.10	1079.10	1146.50
c12	**370.89**	**370.89**	**370.89**
c13	**777.43**	**747.99**	1049.50
c14	**385.16**	**385.16**	580.51

Table 7: Computational result for intersection procedure

Instance	TSTS (2, 2)	TSTS (2, 2, 2)	TSTS (2, 4)	TSTS (3, 3)
c1	682.97	708.72	682.97	777.47
c2	1310.70	1310.70	1347.90	1310.70
c3	1327.60	1352.50	1391.40	1378.80
c4	1788.70	1788.70	1788.70	1865.10
c5	4667.00	4635.50	4667.00	4716.40
c6	**455.75**	**455.75**	**455.75**	**455.75**
c7	1177.10	1265.40	1254.90	1265.40
c8	**739.53**	**739.53**	855.23	825.87
c9	**964.99**	1077.00	1095.60	1019.70
c10	1175.00	1238.30	**1321.90**	1251.70
c11	3273.30	3315.00	3370.00	3546.10
c12	1065.00	1065.00	1065.00	1138.10
c13	**1207.70**	1373.80	1219.40	1227.00
c14	563.16	**523.20**	**523.20**	711.52

7 Conclusion

The paper introduced algorithm TSTS for solving the OVRP. The basic tabu search framework is utilized to direct the local search process. Results on well-known benchmark data sets of the literature had shown the competition and precision of the suggested approach with fixed settings of parameters and sensible computational problems. In comparison to the state-of-the-art, it turned out to be effective providing many new best solutions, although it acquired the minimum released fleet size for each small and large-scale problem instances. Additionally, an overall solution framework is presented, which can be used to handle additional vehicle routing problem versions or related discrete permutation flow combinatorial optimization problems. A research direction worth going after is towards analysis of more complex recombination operators which will include intelligent pattern-identification systems.

8 Open Problem

we offer algorithm TSTS for fixing the OVRP. The fundamental tabu search structure is utilized to direct the local search process. Benefits in well-known benchmark data sets from the literature proven the competition and also accuracy and reliability of the suggested approach with fixed settings of parameters and realistic computational problems. In comparison to the state-of-the-art, this became effective generating many new best solutions, although it obtained the minimum published fleet size with regard to each small and large-scale problem instances. Moreover, a general solution framework is presented, that may be used to treat other vehicle routing problem variants or related discrete permutation flow combinatorial optimization problems. A research direction worth pursuing is towards the analysis of more complex recombination operators which will include intelligent pattern-identification systems. An interesting perspective for future research is to extend the model and the heuristic approaches to

- Improvement of currently present metaheuristics TSTS in order to efficiently solve other VRP.
- Design of effective hybrid metaheuristics though decreasing complexity and also to suggest new metaheuristics actual life purposes.
- Concern with the improvement of more intelligent parameter to manage the utilization of heuristics in a more adaptable way.
- Using different constraint-handling techniques such as penalty functions and filter sets.

References

[1] J. Ai and V. Kachitvichyanukul, *Particle swarm optimization and two solution representations for solving the capacitated vehicle routing problem*, Computers and Industrial Engineering, 56 (2009), pp. 380–387.

[2] B. Baker and M. Ayechew, *A genetic algorithm for the vehicle routing problem*, Computers and Operations Research, 30 (2003), pp. 787–800.

[3] L. Bodin, B. Golden, A. Assad, and M. Ball, *Routing and scheduling of vehicles and crews: the state of the art*, Computers & Operations Research, 10 (1983), pp. 63–211.

[4] J. Brando, *A tabu search algorithm for the open vehicle routing problem*, European Journal of Operational Research, 157 (2004), pp. 552–564.

[5] N. Christofides, A. Mingozzi, and P. Toth, *Combinatorial optimization*, Chichester, UK: Wiley, 1979, ch. The vehicle routing problem, pp. 315–338.

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

[7] Z. Fu, R. Eglese, and L. Y. O. Li, *A new tabu search heuristic for the open vehicle routing problem*, Journal of the Operational Research Society, 56 (2005), pp. 267–274.

[8] ——, *Corrigendum: A new tabu search heuristic for the open vehicle routing problem*, Journal of the Operational Research Society, 57 (2006), pp. 1018–1018.

[9] F. Glover, *Heuristics for integer programming using surrogate constraints*, Decision Science, 8 (1977), pp. 156–166.

[10] P. Hansen, *The steepest ascent mildest descent heuristic for combinatorial programming*, Capri, Italy, 1986, Numerical Methods in Combinatorial Optimization.

[11] A. Hedar, *Studies on Metaheuristics for Continuous Global Optimization Problems*, PhD thesis, Kyoto University, Kyoto, Japan, June 2004.

[12] G. A. P. Kindervater and M. W. P. Savelsbergh, *Vehicle routing: handling edge exchanges*, UK: Wiley, 1998, ch. Local search in combinatorial optimization, pp. 337–360.

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

[14] A. N. Letchford, J. Lysgaard, and R. W. Eglese, *A branch-and-cut algorithm for the capacitated open vehicle routing problem*, Journal of the Operational Research Society, 58 (2007), pp. 1642–1651.

[15] F. Li, B. Golden, and E. Wasil, *The open vehicle routing problem: algorithms, large-scale test problems, and computational results*, Computers & Operations Research, 34 (2007), pp. 2918–2930.

[16] Y. Marinakis and M. Marinaki, *A hybrid genetic-particle swarm optimization algorithm for the vehicle routing problem*, Expert Systems with Applications, 37 (2010), pp. 1146–1455.

[17] N. J. Nilsson, *Principles of Artificial Intelligence*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1980.

[18] I. Osman, *Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem*, Operations Research, 41 (1993), pp. 421–451.

[19] D. Pisinger and S. Ropke, *A general heuristic for vehicle routing problems*, Computers & Operations Research, 38 (2007), pp. 2403–2435.

[20] P. P. Repoussis, D. C. Paraskevopoulos, G. Zobolas, C. D. Tarantilis, and G. Ioannou, *A web-based decision support system for waste lube oils collection and recycling*, European Journal of Operational Research, 195 (2009), pp. 676–700.

[21] P. P. Repoussis, C. D. Tarantilis, and G. Ioannou, *The open vehicle routing problem with time windows*, Journal of the Operational Research Society, 58 (2007), pp. 355–367.

[22] R. Russell, W. C. Chiang, and D. Zepeda, *Integrating multi-product production and distribution in newspaper logistics*, European Journal of Operational Research, 35 (2008), pp. 1576–1588.

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

[24] C. D. Tarantilis, D. Diakoulaki, and C. T. Kiranoudis, *Combination of geographical information system and effective routing algorithms for real life distribution operations*, European Journal of Operational Research, 152 (2004), pp. 437–453.

[25] C. D. Tarantilis, C. Kiranoudis, G. Ioannou, and G. Prastacos, *Solving the open vehicle routing problem via a single parameter metaheuristic algorithm*, Computers & Operations Research, 37 (2005), pp. 443–455.

[26] M. Yousefikhoshbakht, F. Didehvar, and F. Rahmati, *An efficient solution for the vrp by using a hybrid elite ant system*, International Journal of Computers Communications & Control, 9 (2014), pp. 340–347.

[27] X. Zhang and L. Tang, *A new hybrid ant colony optimization algorithm for the vehicle routing problem*, Pattern Recognition Letters, 30 (2009), pp. 848–855.