African Journal of Business Management Vol. 7(11), pp. 801-810, 21 March, 2013
Available online at http://www.academicjournals.org/AJBM
DOI: 10.5897/AJBM2013.1607
ISSN 1993-8233 ©2013 Academic Journals

Full Length Research Paper

A hybrid tabu search algorithm for the vehicle routing problem with simultaneous pickup and delivery and maximum tour time length

Milad Keshvari Fard* and M. Reza Akbari

Department of Industrial Engineering, Sharif University of Technology, Tehran, Iran.

Accepted 4 February, 2013

The vehicle routing problem with simultaneous pick-up and delivery (VRPSPD) and maximum time limit for traversing of each tour is a variant of the classical vehicle routing problem (VRP) where customers require simultaneous delivery and pick-up. Delivery loads are taken from a single depot at the beginning of the vehicle's service, while pick-up loads are delivered to the same depot at the end of the service. Also traversing time of each route should not encroach the specified limit. In this research, the aforesaid problem was introduced and a mixed integer programming model was developed for it. Because of being NP-Hard and the impossibility of solving it in the large instances, a hybrid tabu search algorithm was developed to handle the problem. For producing the initial solution for this algorithm, two methods were built. Furthermore, five procedures for improving the solution were developed, which three of them are being used for inter-route and the other two for intra-route improvement. Computational results were reported for 26 produced test problems of the size between 5 to 200 customers.

Key words: Vehicle routing problem, simultaneous pickup and delivery, maximum tour time length, heuristic, hybrid tabu search.

INTRODUCTION

The vehicle routing problem (VRP) is a general name for a large group of problems for determining vehicle routes, in which each vehicle departs from a specified depot, serves some customers and returns back to the depot at the end of its service. There is a variety of services in the real world, but physical delivery of goods is the most common type of it.

VRP in its ordinary mode consists of a depot, a fleet of homogenous vehicles placed in the depot and a set of customers who need delivery of goods from the depot. The goal of the problem in its simplest status is to minimize the total routing cost subject to the capacity of vehicles. Beside the basic VRP, there is a vast variety of related problems. Toth and Vigo (2002) provided comprehensive details of these problems.

One extension of the basic VRP is the vehicle routing problem with simultaneous pickup and delivery (VRPSPD). In this problem that first was introduced by Min (1989), not only customers require delivery of goods, but also they need a simultaneous pickup of goods that should be sent to the depot.

From the practical point of view, many situations need simultaneously pickup and delivery; for example in soft drink industry not only full bottles should be delivered to the customers, but also empty ones must be picked up and sent back to the depot. Reverse logistics is another area in which planning for vehicle routes is a kind of VRPSPD. This is a very important issue, especially in countries that companies are obliged for taking the responsibility of their products during the life cycle. Managing the returned goods can also take the form of VRPSPD in some problems. Furthermore VRPSPD may be seen in many other real world problems; for example Galvao and Guimaraes formulated the problem of transporting the individuals between the continent and oil exploration and

*Corresponding author. E-mail: milad.keshvarifard@essec.edu. Tel: +33666089896.

production platforms located in the Campos Basin in the state of Rio de Janeiro, in which transportation was carried out by helicopters based on the continent (Montané and Galvao, 2006).

The main contribution of this research is that we have added the maximum tour time length constraint to the original VRPSPD. This constraint can be found in many other vehicle routing problems, but according to our best knowledge it has not been considered in the VRPSPDs.

The main application of this constraint is in problems that the loads are spoilable and the travel time should not exceed a specified amount. As a good example we can mention transporting the comestibles (meat, fruits, vegetables, etc.) through ships. Another application is when there is a necessity for fleets' planning in different times without transforming the problem to a dynamic problem. In this case we can be assured about the upper limit of the time each customer are being served and also availability of a vehicle after a specific time; therefore we will be able to plan in a more efficient way for our fleet.

It should be said that maximum tour time length constraint resembles to the maximum distance travelled restriction. As the costs are presented by either distance or time between the customers, these two problems are very similar, however in our problem, unlike the problems with maximum distance traveled, service time in the customers' place has been considered too. Since this time is not a tiny amount in many cases such as loading and unloading the ships, considering it makes the problem more realistic. Furthermore these two problems have some differences in their essences.

In the current problem, all vehicles are similar and there is just one depot. These assumptions have been considered in all of the VRPSPD researches in the literature. Also to make the problem more realistic, the travel's time matrix between customers is assumed to be asymmetric. The aim of the problem originally is to minimize the travel time of all vehicles, but there can be one more goal of minimizing the vehicles' number in addition to the previous one in the proposed Hybrid algorithm. In this method that first was proposed by Casco et al. (1988), a penalty factor is added to every arc connected to the depot. Moreover, by removing maximum tour time constraint or equating pickup or delivery demands of customers to zero, the problem can be changed to eight more different vehicle routing problems.

The reminder of this paper is organized as follows. A brief literature review of VRPSPD is presented. Also, we present mathematical formulation of the problem, and therefore explain our proposed algorithm for solving the problem. This is then followed by computational results and conclusion.

RELATED WORKS

The VRPSPD was first introduced by Min (1989) which was a real life problem concerning the distribution of books among libraries by two vehicles. To solve this problem, he proposed a two-step procedure in which customers first were clustered into groups according to their positions, and then in each cluster, a TSP is solved by using branch-and-price technique. The resulting infeasible arcs were penalized by setting their lengths to infinity, and the TSPs were solved again. After Min's (1989) research, there was a gap of more than 10 years without any published work on this problem. Salhi and Nagy (1999) introduced a comprehensive classification of different types of vehicle routing problem with pickup and delivery (VRPPD) and separated VRPSPDs from vehicle routing problem with mixed backhaul (VRPMB). They developed four insertion-based heuristics to solve VRPSPD which first generated partial routes for the customers and then inserted the remaining customers into these routes. Dethloff (2001) introduced a cheapest insertion algorithm according to the remaining vehicle's capacity and the least possible increasing of the total length. Nagy and Salhi (2005) added three new methods to their previous work and also presented some methods to refine solutions as well as more complex functions on nodes. Crispim and Brandao (2005) solved VRPSPD by a hybrid algorithm of tabu search and variable neighborhood descent. Chen and Wu (2006) developed a solution that in its first step, the initial solution was made by a cheapest insertion algorithm and then was improved by a hybrid algorithm of tabu search and record-to-record travel. Dell'Amico et al. (2006) presented the first published work on exact method for this problem by a branch-and-price algorithm. Montané and Galvao (2006) considered the problem with maximum tour length assumption. Initial solutions were found using a number of methodologies (sweep, tour partitioning, and extensions of the TSPPD heuristics) and then acted as inputs for the next step which was a tabu search algorithm. Bianchessi and Righini (2007) proposed some constructive algorithms, local search, and tabu search methods for this problem and concluded that using tabu search with complex neighborhoods produces better solutions. Wassan et al. (2008) used a reactive tabu search which utilized modified sweep algorithm for generating initial solutions. Gajpal and Abad (2009a) proposed a saving based algorithm for the problem. In the same year they also proposed a hybrid algorithm of Ant colony system and two local search schemes for it (Gajpal and Abad, 2009a). Subramanian et al. (2010) suggested a parallel metaheuristic algorithm consist of random neighborhood ordering and iterated local search using a constructive algorithm to generate initial solution for this problem. Zachariadis et al. (2009) proposed a hybrid algorithm of tabu search and guided local search that used a construction heuristic based on cost saving to get an initial point. Karlaftis et al. (2009) used a genetic algorithm to solve the problem of ship routing between some ports in Greece. This problem also included the time windows assumption. Ai and Kachitvichyanukul (2009) proposed a particle swarm optimization for this problem. Zachariadis et al. (2010) developed a metaheuristic algorithm based on adaptive memory which used different aspects of the good solutions to make better ones. Catay (2010) proposed a hybrid algorithm of ant colony and local search which made initial solution by the nearest neighborhood heuristic. Mingyong and Erbao (2010) added time windows and maximum distance traveled constraints to the original VRPSPD and solved it by a differential evolution algorithm. Gutiérrez-Jarpa et al. (2010) developed another branch and price algorithm for the problem considering the assumption of time windows. Their algorithm was capable of solving the sample sizes up to 50 customers. Fan (2011) considered VRPSPD problem with time windows and defined customer satisfaction according to the time which customer is being visited and its requested time windows. He used a Tabu search algorithm and solved problem up to the size of 50 customers. Subramanian et al. (2011) proposed a branch and cut algorithm which was able to find better lower and upper bounds for the VRPSPD problems for problems with size up to 200 customers. Zachariadis and Kiranoudis (2011) proposed a local search heuristic which used aspiration criteria of Tabu search for diversification. They tested their algorithm on data with maximum size of 400 customers. Tasan and Gen (2012) presented a genetic algorithm for VRPSPD. Wang and Chen (2012) also considered the problem with time windows constraints and proposed a genetic algorithm for it. Their algorithm was able to solve problems with up to 100 customers. Goksal et al. (2012) used a hybrid algorithm of particle swarm and simulated annealing to solve VRPSPD. Cruz et al. (2012) proposed a hybrid heuristic of variable neighborhood descent, tabu search, and path relinking for VRPSPD. Zhang et al. (2012) were the first who consider VRPSPD with stochastic travel time and solved it using scatter search algorithm. They solved problems with at most 400 customers.

MATHEMATICAL FORMULATION FOR VRPSPD

Suppose G = (V, A) is a directed graph where $V=\{v_0, v_1, ..., v_n\}$ is the vertex set and $A=\{(v_i, v_j): v_i, v_j \in V, i \neq j\}$ is the arc set. Also vertex v_0 represents the depot. The other parameters and variables are as the following.

n: total number of customers; n=|V|-1

t_{ij} : time to traverse distance from customer i to j

p_j : pickup demand of customer j; j=1,…,n

d_j : delivery demand of customer j; j=1,…,n

s_j : service time of customer j; j=1,…,n

Q: vehicle's capacity

T: maximum allowed time for each route

$\bar{k}$: maximum number of available vehicles

Decision variables

x_{ij}^k : is 1 if arc (i, j) belongs to the route operated by vehicle k, otherwise is 0;

y_{ij} : demand picked-up in customers routed up to node i (including node i) and transported in arc (i, j);

z_{ij} : demand to be delivered to customers routed after node i and transported in arc (i, j);

The corresponding mathematical formulation is given by

$$Min \sum_{k=1}^{\bar{k}} \sum_{i=0}^{n} \sum_{j=0}^{n} t_{ij}\, x_{ijk} \quad (1)$$

St:

$$\sum_{i=0}^{n} \sum_{k=1}^{\bar{k}} x_{ijk} = 1 \qquad j=1,...,n \quad (2)$$

$$\sum_{i=0}^{n} x_{ijk} - \sum_{l=0}^{n} x_{jlk} = 0 \qquad j=0,...,n \quad k=1,...,\bar{k} \quad (3)$$

$$\sum_{j=1}^{n} x_{0jk} \leq 1 \qquad k=1,...,\bar{k} \quad (4)$$

$$\sum_{i=0}^{n} y_{ji} - \sum_{l=0}^{n} y_{lj} = p_j \qquad j=1,...,n \quad (5)$$

$$\sum_{i=0}^{n} z_{ji} - \sum_{l=0}^{n} z_{lj} = -d_j \qquad j=1,...,n \quad (6)$$

$$y_{ij} + z_{ij} \leq \sum_{k=1}^{\bar{k}} Q\ x_{ijk} \qquad i=j=0,...,n \quad (7)$$

$$\sum_{i=0}^{n} \sum_{j=0}^{n} x_{ijk}\,(s_i + t_{ij}) \leq T \qquad k=1,...,\bar{k} \quad (8)$$

$$x_{ijk} \in \{0,1\} \quad (9)$$

$$y_{ij}, z_{ij} \geq 0 \quad (10)$$

In this model that is an extension of the proposed model by Montané and Galvao (2006), the objective function seeks to minimize total distance traveled. Constraints (2) ensure that each customer is visited by exactly one vehicle while constraints (3) guarantee that the same vehicle arrives and departs from each customer it serves.

Constraints (4) show the maximum number of vehicles that can be used. Restrictions (5) and (6) are flow equations for pickup and delivery demands, respectively; they guarantee that both demands are satisfied for each customer. Constraints (7) establish that pickup and delivery demands will only be transported using arcs included in the solution; they further impose an upper limit on the total load transported by a vehicle in any given section of the route. Restrictions (8) clarify that traversing time of each route should not exceed the specified limit. Finally constraints (9) and (10) define the nature of the decision variables.

In short the set of aforementioned constraints ensures that each vehicle departs the depot with a load equal to the sum of delivery demand of customers in its route and comes back to the depot with a load equal to the sum of pickup demand of customers in the same route; while there is no violation on vehicle's capacity and maximum tour time. Toth and Vigo (2002) showed that capacitated vehicle routing problem (CVRP) is NP-Hard, and hence the current problem is a generalization of CVRP (by equating traverse time to infinity and equating pickup demands of customers to zero), it is a NP-Hard problem too. Because of this, we should seek heuristic and metaheuristic methods for solving it.

SOLUTION ALGORITHM

Tabu search that was first introduced by Glover (1976), is one of the most famous improving metaheuristic algorithms which is widely used in the literature. It uses an initial solution as a starting basis for seeking improved solutions. We start by describing two constructive heuristics that were used to generate this initial solution, proceed with discussing different neighborhoods, then talking about our tabu search algorithm and its characteristics (short-term memory, tabu tenure, long-term memory and etc.), and finally describe the general framework of the proposed tabu search algorithm for the current problem.

Initial solution

In this research we used two methods to generate initial solution as a starting point to the hybrid algorithm that are as follows.

Initial algorithm 1

In this algorithm, first an initial complete tour starting from the depot which consists of all of the customers is built by solving a TSP problem through a greedy heuristic algorithm. Then the customers, according to their position in this initial tour, are added to the end of routes -just before the depot- which starts from depot and ends to it. Afterwards, it is examined whether there is any violation of vehicle's capacity and maximum tour time length or not. In case of trespassing from each of these constraints, the current route would close and the customer is added to the next route; this will continue until all of customers have been assigned to routes. Then this work is repeated for different starting points in the primary TSP tour. After that, the primary TSP tour is reversed (It is traversed in the opposite direction) and all of the operations above will be repeated for it. Therefore this algorithm recurs for 2n times and at last the best solution is reported as the initial solution.

Initial algorithm 2

This method is actually a constructive heuristic that a TSP sub-problem is solved for producing a route. In this method we choose among customers that still have not been assigned to any route; the one that is the nearest to the last placed customer in the current route, is chosen for the last position in the route (in case that current route is still null, the nearest customer to the depot), and by supposing of existence the depot at the beginning and the end of this route, we check the feasibility of maximum tour time and vehicle's capacity for that route. In case of violating, this customer is removed from current route, the route is closed and the customer is added to a new route from depot to depot. To get an initial solution, both methods are executed and the better answer is caught as the initial solution; thus initial solution is chosen among 2n+1 solutions.

Improvement procedures

Five common improving functions are used in order to enhance solution's quality. As we know, due to the fluctuation of vehicle's load in a route, by adding or substituting a new customer into a route, all of the customers must be reassessed to ensure that vehicle's capacity is not violated. Since this study is based on a complete neighborhood's search, by increasing the problem size, computing time increases exponentially; so some algorithms are developed for these operators to decrease computational operations. Since these algorithms deal with vehicle's capacity, they can be used in other VRPSPDs. Figure 1 shows the function of these improving operators.

Shift

This operator searches to find the best place for each customer in the other routes. Also it may assign a new (empty) route to a customer or removes a route (if it contains just one customer). It should be said that the mechanism for feasibility test is according to the proposed algorithm by Wassan et al. (2008). Also because it is faster to check tour time constraint, it is checked first. To reduce the calculations in each iteration, only the best feasible place is saved for each customer. Then the best replacement is chosen among at most, n, found neighborhoods.

Interchange

This function switches two customers in different routes. It checks all of the possible attitudes for each two different routes and each two customers in these routes, and chooses the best feasible move according to the solution's value and tabu degree of that move. To reduce the huge amount of calculations, at the beginning of each implementation of this procedure, a function calculates the vehicle's load after visiting each of the customers in the route for all of the routes, and identifies customer(s) in each route that the vehicle's load becomes maximum after visiting them.

Also to hasten implementation of this operator, a procedure for reducing the calculations related to vehicle's capacity was developed. For each of the two routes, suppose that the customer which is going to be removed, is X and the customer that is planned to be inserted, is Y. Moreover delivery and pickup demands of X are p_x and d_x. Vehicle's capacity is Cap and maximum load of the vehicle during its current path is maxload. There are two possibilities for X:

1) The position of customer X on its current route is equal or greater than the position of the last maxload on that route. This path would be feasible after switching the two customers if the two conditions below are valid.

- $d_Y - d_X + \max load \leq Cap$
- For all the customers in the route whose positions are between X's position (consist of X) to the last customer in the route:

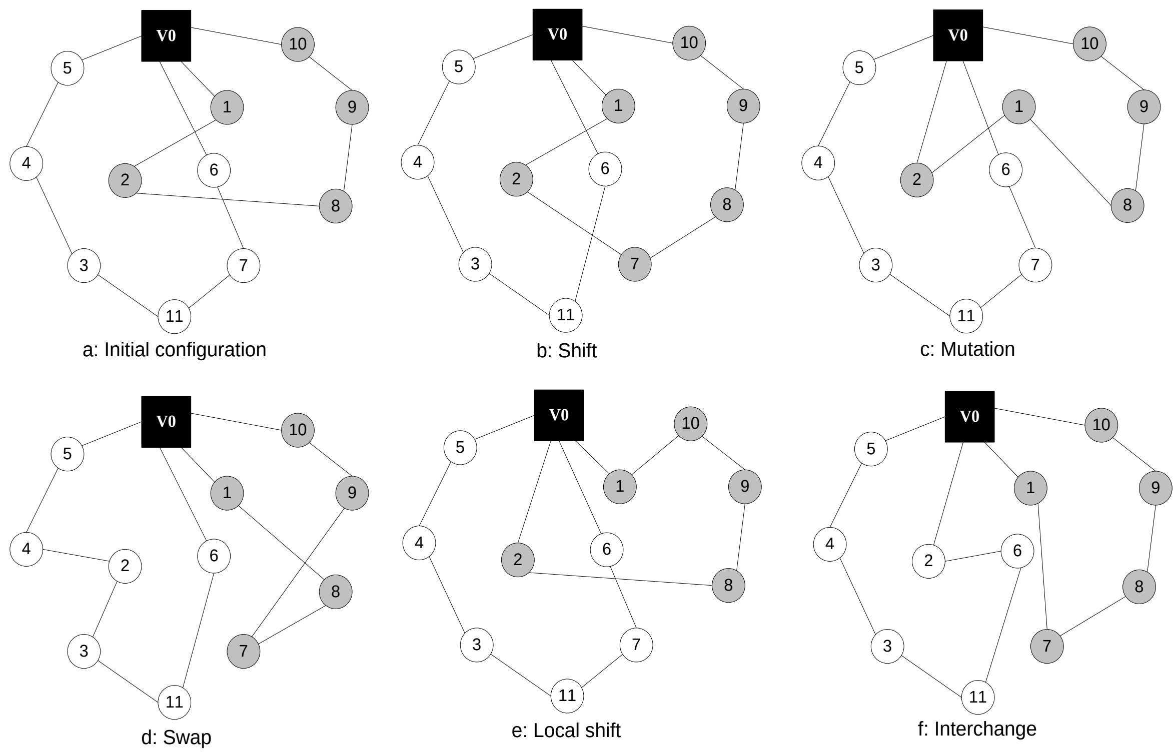

Figure 1. Improving operators.

Vehicle's load after visiting the customer + $p_Y - p_X \leq Cap$

If both conditions are true, then this substitution is feasible.

2) The position of customer X on its current route is less than the position of the first maxload on that route. This path would be feasible after substitution if two conditions below are valid.

- $p_Y - p_X + \max load \leq Cap$
- For all the customers in the route from the beginning of the route until X's position: Vehicle's load after visiting the customer + $d_Y - d_X \leq Cap$

If both conditions be true, then this substitution is feasible.

Therefore we will have four special situations (according to two customers' positions in their current routes) and in the case of none of those (the position of X is between two maxloads), the algorithm would check the whole route for feasibility of vehicle's capacity. This algorithm is caught according to this fact that by adding (removing) customer X, vehicle's load would increase by p_x for the customers after X, and decrease by d_x for those before X. Using this method, checking the vehicle's capacity feasibility is done for almost half of the route and therefore computing time decreases a huge amount. This procedure would give better results in cases that vehicle's capacity is considerable in comparison with each customer's demand (more than 8 times).

Mutation

This function switches the places of two customers within a route. The algorithm that is used in this function to deal with vehicle's capacity constraint for reducing the amount of calculations, is that if the position of the two customers are simultaneously less than the position of the first maxload or simultaneously equal or greater than the position of the last maxload in that route, this switching is feasible for vehicle's capacity; otherwise feasibility of capacity must be examined in a normal way. Also at the beginning of this algorithm, another function should identify customer(s) in each route, which maxload occurs after visiting them.

The logic in this algorithm is that if both customers be at one side of the maxload's position, none of their pickup or delivery loads would affect on maxload and hence, no violation of vehicle's capacity would occur. Using this method caused almost halving the computational operations related to capacity feasibility test of this function. Like Interchange, the performance of this function will increase by having more spacious vehicles.

Swap

This function acts like interchange; it takes different pairs of routes; in each pair chooses pairs of customers and moves each of the two customers simultaneously to the best possible place in the other's route.

Before each iteration of this operator, by a separate function, algorithm finds vehicle's load after visiting each of the customers in a route and the maxload of that route in the case of removing each of the customers of that route, and does it for all of the routes. Also by splitting each part of the algorithm and changing some orders during implementation, computation time decreases more than fifty percent. Because of the numerous steps of the algorithm, we skip it here.

Local shift

This function catches a customer and moves it to the best place on its route. This work is done for all routes and all customers in each route. In this function, the method introduced in Mutation is used in order to decrease the number of needed computations for vehicle's capacity constraint. Also to restrain storing of bad neighborhoods, this function uses the foresaid method in Shift.

The Tabu Search strategy in neighborhood's searching

Each of the five mentioned improving operators act as below in the framework of tabu search.

1. Start;
2. For current solution survey all of the possible neighborhoods and identify their value and tabu degree and save them;
3. Let $x_{n+1} = x_n$ in cases of no feasible solution and go to step 6, otherwise go to step 4;
4. Find the solution with best value (x'); if it is better than x^* (The best solution found) let $x^* = x_{n+1} = x'$ and go to step 6, otherwise go to 5;
5. Choose the solution that first has the least tabu degree and second has the least value (x'') and let $x_{n+1} = x''$;
6. End.

The aforementioned procedure is used has more authorization to move in tabu areas in comparison with the customary procedure of tabu search. The reason for this work is the little number of feasible neighborhoods in each iteration, even by doing a complete search in neighborhood; it is caused by the nature of the problem and constraints. It can be leaded to entrap the search into a local optimum and to prohibit moving to other promising areas especially by progressing the problem and tightening the routes. Hence the mentioned procedure was used to deal with this issue.

Tabu list: In vehicle routing problems, this list is usually about the arcs or nodes that is added or removed from a route. In this paper after each movement, all of the inserted and removed arcs are added to the tabu list.

Tabu tenure: The number of iterations that one arc remains tabu should be higher for deleted arcs than the inserted ones; it is because that inserted arcs are promising and is better to have more freedom. To find the best values for tabu tenure, several values were tested. Unlike many papers that believed tabu tenure should be a percent of the number of customers, best values of this factor were founded in relation to the root of n. In fact, large values for tabu tenure make the search random. Also choosing tabu tenure from an interval showed better results than using constant values.

Tabu tenure for deleted arcs: $$\begin{cases} [2,3] & n \le 6 \\ [3,4] & 7 \le n \le 12 \\ \left[[0.9\sqrt{n}],[1.3\sqrt{n}]\right] & n \ge 13 \end{cases}$$

Tabu tenure for inserted arcs: $$\begin{cases} [2,2] & n \le 6 \\ [2,3] & 7 \le n \le 12 \\ \left[[0.6\sqrt{n}],[0.9\sqrt{n}]\right] & n \ge 13 \end{cases}$$

Long term memory: Two diversification and intensification phases were used in order to improve the solution quality. The method is exactly the one that was used by Montané and Galvao (2006).

Hybrid tabu search algorithm framework

Step 0: Generate an initial solution by two mentioned procedures.
Step 1: Perform normal tabu search by starting from the initial solution for I1 iterations.
Step 2:
1- Equate the tabu list values to zero.
Halve intervals of tabu tenure.
Perform intensification phase by starting from the best found solution in the previous step for I2 iterations.
Step 3: Performing diversification phase for I3 iterations.
Step 4:
Return tabu tenure values to its original state.
Perform normal tabu search for I4 iterations.
Step 5: Perform local search phase by starting from the best solution found in previous steps for I5 iterations.

Although the stopping condition is finishing of all the above steps, but steps 1 and 4 will finish if there is no change in the best found solution for I6 iterations. The values of I1 to I6 are follows. They decrease by increasing the problem's size in such way to catch the best balance between solution's quality and time.

I1= [4000/ $\sqrt{n}$];
I2= [I1/6];
I3= [I1/8];
I4= [$\frac{2}{5}$ I1];
I5=10;
I6= [7000/ $\sqrt{n}$];

In steps 1, 4 and 5, in each iteration, three improving operators of Shift, Swap and Local shift are done successively. Also five improving operators of Shift, Swap, Local shift, Interchange and Mutation act in a sequential manner in each iteration of intensification and diversification phases.

COMPUTATIONAL RESULTS

For implementing the proposed algorithm, instances with number of customers equal to 5,6,7,8,9,10,11,12,20,50, 100,150 and 200 are created. 12, was the problem with the maximum number of customers which could be solved by exact method in less than 30 min.

Three instances for each of the scales with 50 customers or more for better assessing were used. As the problem was introduced for the first time, no match benchmark data were available and hence our data is generated by distributions (as follow) that are adopted from CMT problems of Salhi and Nagy (1999).

- X coordinate of customers and depot: UNI [0-71]
- Y coordinate of customers and depot: UNI [0-71]
- Pickup demand of each customer: UNI [0-41]
- Delivery demand of each customer: UNI [0-41]

Also service time to each customer was caught from UNI [1-12], vehicle's capacity considered to be 100 and the total tour time was equated to 330.

The proposed hybrid tabu search algorithm was implemented in MATLAB and the exact method was

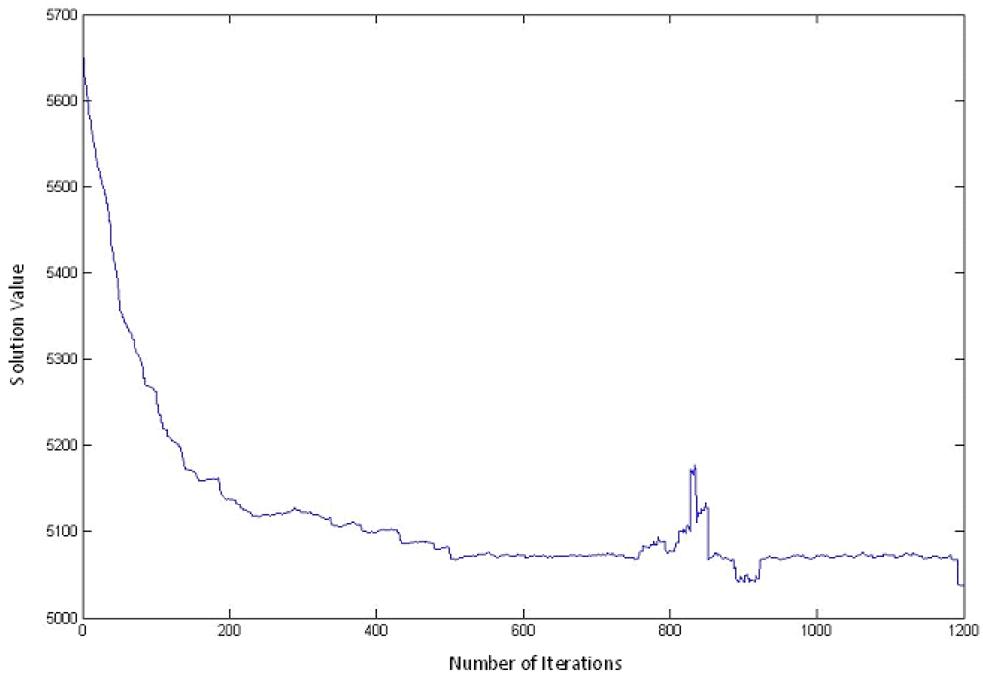

Figure 2. Convergence of hybrid tabu search algorithm in problem 200-3.

executed by CPLEX 12. Both codes ran on an INTEL Core 2 Quad (using just a single core) with 2.4 GHz, 4 GB RAM, and operating system of Windows 7.

Comparison between initial solution algorithms

As earlier mentioned, two methods were used to generate initial solutions. In Table 1, required time for each method to be performed, number of routes generated by that method and cost of the proposed solution are presented.

From observation, solutions quality in initial algorithm 1 is better or equal to initial algorithm 2 and hence it is identified as the starting point of search by the tabu search algorithm. However initial algorithm 2 in most cases produced acceptable solutions (3.27% difference with initial algorithm 1) and the two problems generated the best initial solution; but the strength point of this algorithm is in its very small computing times as well as a linear increase in them by enhancing the problem’s size unlike the other method.

An evidence for the power of these two algorithms is the small difference of the accepted initial solution with the global optimum found by exact method (4.64%) in 8 small problems (problems 5 to 12). They also directly produced the global optimum for problems 5 and 9.

Computational results of the hybrid tabu search algorithm

In Table 2, information about solving time, the number of produced routes and objective functions are presented for both of the hybrid algorithm and exact method.

As it can be seen, for all of small problems except problem 8, the solution found is equal to the solution of exact algorithm. This reveals the power of the hybrid algorithm. Figure 2 demonstrates the convergence of this algorithm during the implementation. In this figure the effect of each step is obvious.

Both in this problem or other problems solved by this algorithm, the Tabu search algorithm starts with a steep decrease in the objective function and after a while it reaches to a steady state. Then, intensification phase is started and we observed some decrease (between 0.1% to 1.2%) in the best solutions found in previous phase; these solutions had very little deviance from those of previous phase by the general position of the customers in the routes. After that, diversification phase makes some turmoil in the solutions, and in near half of the cases, it could generate improvements up to 0.9% in the solution. The fourth phase gave improvement just in 4 problems; none of them were higher than 1%. Finally, the local search phase which was started from the best solution found in previous phases, almost in all cases provided improvements at most equal to 0.7%.

The proposed Hybrid algorithm was able to create a total decrease of 10.5% to more than 17% (12.5% in average) to the initial solution in medium and large size problems. The algorithm also provided an average of 17.3% decrease in the number of required routes from the best initial solution. The latter improvement has a high importance, since it determines the number of required vehicles and has a high weight in determining the total

Table 1. Comparison between methods of generating initial solution.

Problem	Initial algorithm 1			Initial algorithm 2		
	Time(s)	No. of routes	Value	Time (s)	No. of routes	Value
5	0.00376	2	272.1	0.00083	2	277.6
6	0.00505	2	355	0.00099	3	367.8
7	0.00841	2	307.8	0.00099	2	307.8
8	0.00801	3	378.3	0.00103	3	468.4
9	0.0091	2	380.2	0.00108	2	386
10	0.02013	4	413.5	0.00189	4	436.7
11	0.02105	3	501.9	0.00121	3	513.6
12	0.01708	4	457.3	0.00132	4	486.2
20	0.04229	7	734.2	0.00181	7	734.2
50-1	0.17155	12	1661.9	0.00224	13	1697.2
50-2	0.18598	14	1509.1	0.00225	15	1524.3
50-3	0.16843	15	1631.4	0.00225	15	1702.6
100-1	0.58481	28	3377.1	0.00425	29	3474.9
100-2	0.61825	29	3111	0.00434	29	3132.2
100-3	0.57716	26	2759.9	0.00422	26	2818.2
150-1	1.33166	42	4285.8	0.00636	42	4311.7
150-2	1.33994	44	4959.2	0.00641	47	5257.3
150-3	1.36214	45	5047.3	0.00641	45	5076.8
200-1	2.39265	61	6679.2	0.00833	61	6810.1
200-2	2.3817	61	6270.2	0.0086	62	6303.2
200-3	2.33211	54	5657.7	0.00847	53	5699.9

Table 2. Results for the hybrid tabu search and the exact method.

Problem	Initial solution		Hybrid tabu search			Exact method		
	No. of routes	Value	Time (s)	No. of routes	Value	Time (s)	No. of routes	Value
5	2	272.1	0.6	2	272.1	0.01	2	272.1
6	2	355.0	0.7	2	331.0	0.19	2	331.0
7	2	307.8	1.5	2	302.1	1.31	2	302.1
8	3	378.3	3.7	2	357.5	2.06	3	344.5
9	2	380.2	1.6	2	380.2	7.05	2	380.2
10	4	413.5	1.2	4	376.1	19.34	4	376.1
11	4	501.9	1.5	3	468.3	24.14	3	468.3
12	4	457.3	1.7	4	439.9	210.60	4	439.9
20	7	734.2	30.7	6	637.0	-	-	-
50-1	12	1661.9	35.5	11	1476.8	-	-	-
50-2	14	1509.1	45.3	12	1308.9	-	-	-
50-3	15	1631.4	46.3	13	1460.3	-	-	-
100-1	28	3377.1	76.0	22	2802.7	-	-	-
100-2	29	3111.0	94.5	23	2730.8	-	-	-
100-3	26	2759.9	86.9	22	2491.6	-	-	-
150-1	42	4285.8	169.1	35	3766.4	-	-	-
150-2	44	4959.2	147.3	36	4413.2	-	-	-
150-3	45	5047.3	180.3	37	4403.2	-	-	-
200-1	61	6679.2	311.1	47	5640.2	-	-	-
200-2	61	6270.2	376.8	48	5416.5	-	-	-
200-3	54	5657.7	364.3	45	5037.0	-	-	-

cost.

Conclusion

VRPSPD has been receiving growing attention due to the increasing importance of the reserve logistic activities in the past few years. In this paper the vehicle routing problem with simultaneous pickup and delivery and maximum tour time length constraint was introduced for the first time to cope with the subset of such problems in which the total time of each trip (consist of travel time as well as loading and unloading time) should not exceed a specified amount. Then a mathematical formulation was developed for the problem. Since this problem is NP-Hard, exact methods cannot deal with it in large scale instances, and hence using heuristic methods to solve it is inevitable.

For this purpose, first, two procedures for generating the initial solution were developed. Although solution's values which were produced by initial algorithm 1 were slightly better than those of the initial algorithm 2, but the computational time of the latter algorithm was significantly better than the time needed by initial algorithm 1.

Then a number of the most famous improving functions were developed to be compatible with current problem. Also by doing some changes in their algorithms, needed operations and therefore their computational time were decreased considerably. Hence these modifies were about vehicle's capacity, those algorithms with some little changes would be applicable for the other VRPSPDs. So in addition to developing a new problem in the field of VRPSPD, the second contribution of this paper is defined as developing new procedures for some famous improving methods in VRPSPDs for reducing calculation-time, which are practical in other VRPSPDs.

Then, a hybrid algorithm of tabu search and local search was developed to improve the initial solution through the improving procedures. It was implemented on test problems from 5 to 200 customers. This algorithm could find global optimum in 88% of small problems and showed to be a time effective algorithm since the maximum time it spent was 376 s.

In terms of future research directions, the proposed algorithm can be tested on VRPSPD without tour time constraint to show its strength in comparison with the other algorithms developed for it. Moreover this algorithm can be combined with other heuristics to reach better results. Other algorithms which have been verified to be powerful in dealing with VRPSPDs like ACS can be developed for this problem. Finally adding multiple depots assumption makes the problem more realistic.

REFERENCES

Ai T, Kachitvichyanukul V (2009). "A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery" Comput. Oper. Res. 36:1693-1702.

Bianchessi N, Righini G (2007). "Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery", Comput. Oper. Res. 34:578-594.

Casco D, Golden B, Wasil E (1988). "Vehicle routing with backhauls: models algorithms and case studies", In: Golden B, Assad A (Eds.), Vehicle Routing: Methods and Studies. North-Holland, Amsterdam pp.127-147.

Catay B (2010). "A new saving-based ant algorithm for the Vehicle Routing Problem with Simultaneous Pickup and Delivery", Expert Syst. Appl. 37(10):6809-6817.

Chen J, Wu T (2006). "Vehicle routing problem with simultaneous deliveries and pickups", J. Oper. Res. Soc. 57(5):579-587.

Crispim J, Brandao J (2005). "Metaheuristics applied to mixed and simultaneous extensions of vehicle routing problems with backhauls", J. Oper. Res. Soc. 56(11):1296-1302.

Cruz R, Silva T, Souza M, Coelho V, Mine M, Martins A (2012). "GENVNS-TS-CL-PR: A heuristic approach for solving the vehicle routing problem with simultaneous pickup and delivery", Electron. Notes Discrete Math. 39:217-224 .

Dell'Amico M, Righini G, Salani M (2006). "A branch-and-price approach to the vehicle routing problem with simultaneous distribution and collection", Transp. Sci. 40(2):235-247.

Dethloff J (2001). "Vehicle routing and reverse logistics: The vehicle routing problem with simultaneous delivery and pick-up", OR Spektrum 23:79-96.

Fan J (2011). "The Vehicle Routing Problem with Simultaneous Pickup and Delivery Based on Customer Satisfaction", Procedia Eng. 15:5284-5289

Gajpal Y, Abad P (2009a). "An ant colony system (ACS) for vehicle routing problem with simultaneous delivery and pickup", Comput. Oper. Res. 36(12):3215-3223.

Gajpal Y, Abad P (2009b). "Saving-based algorithms for vehicle routing problem with simultaneous pickup and delivery", J. Oper. Res. Soc. 61:1498-1509.

Goksal F, Karaoglanb I, Altiparmak F (2012). "A hybrid discrete particle swarm optimization for vehicle routing problem with simultaneous pickup and delivery", Comput. Ind. Eng. (In Press).

Gutiérrez-Jarpa G, Desaulniers G, Laporte G, Marianov V (2010). "A branch-and-price algorithm for the Vehicle Routing Problem with Deliveries, Selective Pickups and Time Windows", Eur. J. Oper. Res. 206:341-349.

Karlaftis M, Kepaptsoglou K, Sambracos E (2009). "Containership routing with time deadlines and simultaneous deliveries and pick-ups", Transp. Res. 45(1):210-221.

Min H (1989). "The multiple vehicle routing problem with simultaneous delivery and pickup points", Transp. Res. A 23(5):377-386

Mingyong L, Erbao C (2010). "An improved differential evolution algorithm for vehicle routing problem with simultaneous pickups and deliveries and time windows", Eng. Appl. Artif. Intell. 23(2):188-195.

Montané F, Galvao R (2006). "A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service", Comput. Operat. Res. 33(3):595-619.

Nagy G, Salhi S (2005). "Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries", Eur. J. Oper. Res. 162(1):126-141.

Salhi S, Nagy G (1999). "A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling", J. Oper. Res. Soc. 50(10):1034-1042.

Subramanian A, Drummonda L, Bentes C, Ochi L, Farias R (2010). "A parallel heuristic for the Vehicle Routing Problem with Simultaneous Pickup and Delivery", Comput. Oper. Res. 37(11):1899-1911.

Subramanian A, Uchoac E, Pessoa A, Ochi L (2011). "Branch-and-cut with lazy separation for the vehicle routing problem with simultaneous pickup and delivery", Oper. Res. Lett. (In press).

Tasan AS, Gen M (2012). "A genetic algorithm based approach to vehicle routing problem with simultaneous pick-up and deliveries", Comput. Ind. Eng. 62(3):755-761.

Toth P, Vigo D (2002). "The vehicle routing problem", Philadelphia, PA: SIAM Philadelphia.

Wang H, Chen Y (2012). "A genetic algorithm for the simultaneous delivery and pickup problems with time window", Comput. Ind. Eng. pp.84-95.

Wassan N, Wassan A, Nagy G (2008). "A reactive tabu search algorithm for the vehicle routing problem with simultaneous pickups and deliveries", J. Comb. Optim. 15(4):368-386.

Zachariadis E, Kiranoudis C (2011). “A local search metaheuristic algorithm for the vehicle routing problem with simultaneous pick-ups and deliveries”, Expert. Syst. Appl. 38(3):2717-2726.

Zachariadis E, Tarantilis C, Kiranoudis C (2009). "A hybrid metaheuristic algorithm for the vehicle routing problem with simultaneous delivery and pick-up service", Expert Syst. Appl. 36(2):1070-1081.

Zachariadis E, Tarantilis C, Kiranoudis C (2010). "An adaptive memory methodology for the vehicle routing problem with simultaneous pick-ups and deliveries", Eur. J. Oper. Res. 202(2):401-411.

Zhang T, Chaovalitwongse W, Zhang Y (2012). “Scatter search for the stochastic travel-time vehicle routing problem with simultaneous pick-ups and deliveries”, Comput. Oper. Res. 39(10):2277-2290.