A new hybrid metaheuristics for the vehicle routing problem with simultaneous pick-up and delivery

Lijun Meng, Xiaochai Guo
Department of Management,
Zhejiang University
Hangzhou, China
menglijun1982.student@sina.com, ggxxcc06@zju.edu.cn

Abstract —**The vehicle routing problem with simultaneous pick-up and delivery is the problem of optimally integrating goods distribution and waste collection. We analyze the vehicle routing problem in which, a single depot can offer service to many customers that may both receive and send good by multiple vehicles, customers strictly require that their delivery and pickup is to be served in a single visit by a vehicle simultaneously. The objective of the problem is to minimize the total distance to service all customers, each with a single stop, without violating the capacity of each vehicle. Firstly, the revised tour-partitioning algorithm is used to get the initial solution of the vehicle routing problem with simultaneous pick-up and delivery In the reactive tabu search phase, variable neighborhood search modules are also proposed to improve the results. Finally, computational experiments show that the proposed hybrid heuristic is able to obtaining optimal solutions very efficiently.**

Keywords-vehichel routing problem with simultaneously delivery and pickup; reactive tabu search; variable neighbourhood search

I. INTRODUCTION

The vehicle routing problem with simultaneous pick-up and delivery (VRPSPD) is to optimizing the delivery of goods from warehoused to customers and the simultaneous collection of wasted or used products from customers to warehouses or some specialized recycling sites. Such an example is the distribution system grocery store chains.

VRPSPD is an extension to the VRP. Given a number of customers with a known delivery demand and pickup demand and located on a transportation network, there is a single depot that can supply the customers and receive their good. Customers strictly require that their delivery and pickup is to be serviced in a single visit by a vehicle. The service is provided by a set of vehicles of limited capacity that leaves the depot carrying a mount of goods and returns to the depot carrying a mount of wastes. In each point along its tour each vehicle cannot carry a total load greater than its capacity. The goal is to minimize the total distance of the tours, subject to maximum capacity constraints on the vehicle.

The problem of finding a feasible solution to the VRPSPD with a prescribed number of vehicles is NP-hard. So many researches focus on the metaheuristic for the VRPSPD problem. Also, in this paper, a revised-tour partitioning procedure to generate initial solutions and a heuristic based on the variable neighborhoods search and reactive tabu search mechanism are proposed to resolve the VRPSPD.

The rest of this paper is organized into five sections. In section Ⅱ, we presented a review of the literature about VRPSPD. Section Ⅲ presented a mathematical model of the VRPSPD. In section Ⅳ, we briefly discussed the variable neighborhoods search and reactive tabu search mechanism, and presented out algorithm. In section Ⅴ,we present our experimental results obtained on problem instance and its comparison with Nicola et al 's study. Finally, conclusions and suggestions for future research are discussed in the last section.

II. LITERATURE REVIEW

Many authors recognize the necessity of a simultaneous delivery and pickup at the same customer location and the feasibility of a mixed load of deliveries and pickups in the same vehicle.

Reference [1] was the first study about VRPSPD. This study solved a practical problem faced by a public library, with one depot, two vehicles and 22 customers. The customers were first clustered into groups and then in each group the traveling salesman problems were solved. The infeasible arcs were penalized and the problems are resolved. Reference [2] proposed a load-based insertion procedure that extends the idea of 1-insertion to cluster insertion and tested four insertion methods. Their computation results showed that cluster insertion method offered positive improvement.

Reference [3] used an insertion-based algorithm more typically adopted for mixed problem. Customers were inserted into emerging routes according to three criteria: travel distance, capacity remaining and distance from the depot. Reference [4] developed an insertion heuristic called RESCAP-RS, based on the residual capacity concept to avoid the short-sightedness of a simple greedy insertion.

Reference [5] used the tabu search but found initial feasible solutions by an insertion method, which relies on both distance-based criteria and load-based criteria. The neighborhoods for the improvement phase was built on the moves 2-exchange, swap, shift, 2-opt and or-opt. An effective hybrid heuristic based on the record-to-record travel, tabu lists,

978-1-4244-2013-1/08/$25.00 ©2008 IEEE

and the route improvement procedures was proposed to improve the initial solution.

Reference [6] presented a hybrid of tabu search and variable neighborhood search. Initial solutions are generated using a sweep method. If any route within this solution is infeasible due to intermediate arcs being overloaded, the order of the customers on the route is exchanged until was established. The improvement phase is built on the moves "insert" and "swap."

Reference [7] developed a tabu search algorithm relied on both short-term and long-term memory to solve VRPSPD with maximum travel distance. Initial solutions are found using a number of methodologies, including sweep, tour partitioning. This algorithm uses three types of movements to obtain inter-route adjacent solution. Two different search strategies were implemented for selecting the next movement: first admissible movement and best admissible movement.

Reference [8] presented several heuristic algorithms for the VRPSPD, and addressed the issue of applying the tabu search paradigm to algorithms based on complex and variable neighborhoods. Reference [9] designed a reactive tabu search metaheuristic that can check feasibility of proposed moves quickly and reacts to repetitions to guide the search. They modified the classical sweep method to generate the initial solution. The RTS guided the search towards better solutions by dynamically maintaining the tabu list size. This study pointed that combing reactive tabu search with variable neighborhood search was likely to yield a powerful algorithm.

III. MATHEMATIC MODELS

Notations

V, the set of customer

V0, the set of customers and depot

n, customer number

c_{oj}, the distance between the depot and customer j, $j = 1,2,3,\ldots,n$

c_{ij}, the distance between customer i and customer j, $i, j = 1,2,3,\ldots,n$

q_i, the demand of customer i $i = 1,2,3,\ldots,n$

p_i, The supply of customer i, $i = 1,2,3,\ldots,n$

MC, the maximum capacity which the vehicles.

$x_{ij}^k = 1$, if arc j is part of route k which is served by vehicle k, otherwise, it equals to 0

y_{ij}, demand pick-up in clients routed up to node i (including node i) and transported in arc (i,j)

z_{ij}, demand to be delivered to client after node i and transported in arc (i,j)

Models

$$\text{Minimize} \sum_{k=1}^{\bar{k}} \sum_{i=0}^{n} \sum_{j=0}^{n} c_{ij} x_{ij}^k . \quad (1)$$

Subject to:

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

$$\sum_{i=0}^{n} x_{ij}^k - \sum_{i-0}^{n} x_{ji}^k = 0, j = 1,\ldots,n, \quad k = 0,\ldots,\bar{k} . \quad (3)$$

$$\sum_{i=0}^{n} y_{ji} - \sum_{i=0}^{n} y_{ij} = p_j, \forall j \neq 0 . \quad (4)$$

$$\sum_{i=0}^{n} z_{ij} - \sum_{i=0}^{n} z_{ji} = d_j, \forall j \neq 0 . \quad (5)$$

$$y_{ij} + z_{ij} \leq MC \sum_{k=1}^{\bar{k}} x_{ij}^k, i, j = 0,\ldots,n . \quad (6)$$

$$x_{ij}^k \in \{0,1\}, i, j = 0,\ldots,n . \quad (7)$$

$$z_{ij} \geq 0, i, j = 0,\ldots,n . \quad (8)$$

$$y_{ij} \geq 0, i, j = 0,\ldots,n . \quad (9)$$

Constraints (2) force every customer to be visited once; constraint (3) (4) and (5) are flow conservation constraints on the number of vehicles and on the amounts of pick-up and delivery load; constraint (6) ensure that the vehicle capacity is not exceeded. Constraints (7), (8) and (9) define the nature of the decision variables.

IV. SOLUTION ALGORITHM

Our algorithm consists of two phases: constructing an initial solution and then improving on it.

A. Initial Solution

We modified the tour partitioning proposed in [10] to generate the initial solution. The original tour partitioning procedure requires solving the TSP firstly. To avoiding the complexity of solving TSP, we skip this step. By ordering the customers, basic groups are constructed. Let the basic group replace the corresponding TSP solution as the initial input to the improvement phase.

Step 1: the customers were ordered depending on the net demand from maximum value to minimum value. Every customer had its own order number. Equation (10) was used to determine the net demand of each customer

$$\text{The net demand} = q_i - p_i . \quad (10)$$

Step2: the minimum number of routes is calculated using (11):

$$N = \max\left(\sum_{i=1}^{n} p_i / MC, \sum_{i=1}^{n} q_i / MC\right) \quad (11)$$

Step 3: according to the customer's net demand order, classify the customers into N groups, called basic group.

3.1The first customer of the basic group $i\ (i \le N)$ is the customer whose order number is i.

3.2The second customer of the basic group $i\ (i \le N)$ is the customer whose order number is $i+N$

3.3Obviously, the mth customer of basic group $i\ (i \le N)$ is the customer whose order number is $i+m*N$

3.4Stop the process until all the customers have been classified into basic groups.

Step 4: The tour partitioning procedure is used to get an initial feasible route for every basic group.

4.1 Initialize $i=1$

4.2 A route served by a single vehicle is started from the first customer of the basic group i. Every vehicle travels along the basic group as far as possible that is until it reached a capacity violation. The vehicle skips the customer that create a capacity violation, travel along the basic group until arrives at the end of basic group.

4.3The customers that create a capacity violation will be left to be incorporated into a residual group.

4.4 Set $i=i+1$, perform step 4.2 and step 4.3,until $i=N+1$

Step 5: if there are still some customer left in the residual group, set the residual group as the input of the procedure, go to step1; else stop the initial solution procedure.

B. *Improvement Procedures*

Routine REVERSE: this procedure entails simply reversing the direction of a route.

Routine RELOCATION: this neighborhood includes all strongly feasible solutions obtained by deleting customer i from t_1 and inserting it into tour t_2.

Routine 2-EXCHANGE: two links are removed form two different routes and two new links are introduced by connecting the first customer on the first link to the last customer on the second link and connecting the first customer on the second link to the last customer on the first link.

Routine 2-OPT: in this procedure, two links not adjacent to each other are removed from the same route and the segments are connected in all possible ways.

Routine PERTURB: a customer i is removed from a route t_1 and inserted into a route t_2 ,at the same time a customer j is removed from t_2 and inserted into a third route t_3.

Routine SELF-GROUP EXCHANGE: this routine is an extension to relocation. Two customers i and j belong to the same tour t_1, customer i is removed from its location and inserted into customer j 's location. Simultaneously, customer j is removed from its location and inserted into customer i 's location.

Routine DIFFERENT-GROUP EXCHANGE: this routine is an extension to relocation. Customer i belongs to the tour t_1, and customer j belongs to the different tour t_2. Customer i is removed from its location in tour t_1 and inserted into customer j 's location in tour t_2. Simultaneously, customer j is removed from its location in t_2 and inserted into customer i 's location in tour t_1.

C. *The Variable Neighborhood Search*

The variable neighborhood search of Hansen and Mladenovic begins with an initial solution that is improved iteratively through movements in different kinds of neighborhoods that are sorted a priori according a certain criterion. Then at each local search step a neighborhood is completely explored. If an improving solution is found in it, then the local search step is terminated and the current solution is updated; otherwise the next neighborhood is explored. The search terminated when no improvement solution is found in any neighborhood. The order in which the neighborhoods are considered is fixed, and at each iteration, the neighborhood list is scanned from the beginning.

According to two criterions: the average time per iteration and the average cost of the solutions computed by the corresponding simple neighborhood local search, the order in which the neighborhoods are considered in our algorithm is given in table Ⅰ.

Steps of variable neithbourhood search are as follow:

Step 1. Set $k=1$

Step 2. Exploration of neighborhood using the kth neighborhood routines. Find the best neighborhood S' of S

Step 3. If the solution thus obtained S' is better than S, , stop the variable neighborhood search algorithm;

Else, set $k=k+1$, go to step 2.until $k=7$

In our test, we use the same technique described in reference [8]: customer number is 100, D=10, the results are the average values computed on 50 instances.

D. *The Reactive Tabu Search*

Reactive tabu search (RTS) first introduced by [11] focus on tabu list size (tls). In RTS scheme, the *tls* depends on the repetition of solutions and consequently *tls* is determined dynamically. RTS employs two mechanisms that both react to the repetitions. The first mechanism is used to product a balanced tabu list size and consists of two reactions. The fast reaction increased the list size when solutions are repeated; the slow reaction reduces the list size for those regions of the search space that do not need large list lengths. The second mechanism provides a systematic way to diversity the search when it is only confined to one portion of the solution space.

TABLE I. ORDER OF THE SEVEN IMPROVEMENT ROUTINES

Routine	Results	Time	Rank
2-exchange	7914.58	0.149	6
Relocation	7897.64	0.566	5
Different-group interchange	7895.64	0.015	4
Self-group interchange	7144.52	0.004	1
Reverse	8051.8	0.0004	7
2-opt	7189.3	0.0004	2
Perturb	7796.46	35.369	3

The constant values for the RTS parameters are the same as in [11]. Table II shows the parameter in RTS mechanism and its value.

RTS mechanism (see in [9]):

(1) Search of repetition of S, if found than

GapRept=CurTimeRept-LastTimeRept,and go to (2);

Else go to (4)

(2) If repetition of S>REP, t

hen set Chaotic=Chaotic+1;Else go to (3).

If Chaotic>Chaos, then set Chaotic=0, clear and rebuild tabu search data structures and go to Step 3 of the RTS algorithm;

Else go to (3).

(3) If GapRept<GapMax,

than tls=tls*Increase, set LastChange=0, MovAvg=0.1*GapRept+0.9*MovAvg; and go to (5);

Else go to (4).

(4) If LastChange>MovAvg, thentls=tls*Decrease, and set LastChange=0;

Else set LastChange=LastChange+1, store the solution S and go to (5).

(5) Stop the RTS mechanism.

TABLE II. THE PARAMETER IN RTS MECHANISM

Parameter		Value
tls	Initial tabu list size value	1
Chaotics	Counter for the often-repeated sets of solution	0
MovAvg	Moving average for the detected repetitions	0
GapRept	Gap between two consecutive repetitions	0
LastChange	Number of iterations since last change in tls value	0
LastTimeRept	Iteration number when last time an identical solution was noticed.	0
CurTimeRept	Iteration number of the most recent repetition	0
REP	Maximum limit for the often-repeated solutions	3
Chaos	Maximum limit for the sets of often-repeated solutions	3
Increase	Percentage increase for the tabu tenure value	1.1
Decrease	Percentage decrease for the tabu tenure value	0.9
GapMax	Constant used to compare with GapRept to get the moving average	50

E. Hybrid heutistic

The hybrid heuristic is now outlined as follows:

Step 1: Initial Solution: Generate an initial solution S based on the revised tour partitioning procedure. Set $S_{best} = S$

Step 2: Initialization: Initialize all the parameters in the reactive tabu search mechanism. Set number of iterations 200.

Step 3: Search Framework:

Using the variable neighbourhood search algorithm to get the neighbourhood of S. Using the RTS mechanism, update tabu list size tls and tabu list.

If distance(S)<distance(S_{best}), then set $S_{best} = S$

Step 4: Termination: Perform step 3 until iter is reached.

V. COMPUTATIONAL RESULTS

A. data generation

In our test ,we use the same technique described in [8]: 6 sets of 50 instance, with a number of customers equal to 50 and 100.Customer's coordinates are integer numbers uniformly distributed between 0 and 100,and the depot is located at (0,0). Customers' demands are integer numbers uniformly distributed in the range [1, D] with D=10, 20 and 40. The demand of each customer is considered as a pick-up or a delivery demand with probability 50%.We considered capacity values equal to the maximum value of customers' demand, that is D, and to twice as much. The results are the average values computed on 50 instances.

All solution algorithms and procedures described in this research were coded in C language, and all computational experiments were conducted on a Pentium IV1.3GHz PC.

B. Computational Results and Comparison

Table III,IV, V ,and VI reported the computational results of our hybrid algorithm and its comparison with the reference [8]'s results.In table III,IV, V ,and VI, the column "compare" is obtained using (12):

The column "compare" = (vehicle (or distance) of our algorithm's result-vehicle (or distance) of reference [8] results)/ vehicle (or distance) of reference [8], (12)

The computational results show that:

(1) The revised tour partitioning procedure we propose could obtain higher initial solutions than reference [8]' results, both from the distance and the number of vehicles.

(2) The hybrid algorithm we propose could obtain better final solutions, both from the distance and the number of vehicles, than reference [8]' results, in the parameter set that maximum capacity of vehicle is equal to D.

(3) The hybrid algorithm we propose could obtain final solutions whose number of vehicles is fewer than reference [8]' results, and distance is longer than reference [8]' results ,in the parameter set that maximum capacity of vehicle is equal to 2*D.

TABLE III. COMPUTATION RESULTS FOR N=50, MC=D

D	Reference [8] 's results				our algorithm's results								
	Initial		*Final*		*Initial*				*Final*				
	Distance	Veh	Distance	Veh	Veh	Compare	Distance	Compare	Veh	Compare	Distance	Compare	Time
10	3574	17.34	3080	17.02	19.9	+14.8%	4671	+30.7%	15.86	-6.8%	2944	-4.4%	82.
20	3539	16.88	2971	16.50	19	+12.6%	4574	+29.3%	15.46	-6.3%	2887	-2.8%	84
40	3476	16.16	2945	15.86	18.1	+7%	4422	+21.4%	15	-5.4%	2782	-5.5%	82

TABLE IV. COMPUTATION RESULTS FOR N=50, MC=2*D

D	Reference [8] 's results				our algorithm's results								
	Initial		*Final*		*Initial*				*Final*				
	Distance	Veh	Distance	Veh	Veh	Compare	Distance	Compare	Veh	Compare	Distance	Compare	Time
10	2138	8.58	1654	8.44	9.02	+5%	3558	+66.5%	8.12	-3.8%	1733	+4.8%	79
20	2106	8.18	1588	8.08	8.68	+6%	3541	+68.1%	7.78	-3.7%	1667	+5%	80
40	2090	7.96	1567	7.9	8.3	+4%	3415	+63.4%	7.6	-3.8%	1620	+3.4%	85

TABLE V. COMPUTATION RESULTS FOR N=100, MC=D

D	Reference [8] 's results				our algorithm's results								
	Initial		*Final*		*Initial*				*Final*				
	Distance	Veh	Distance	Veh	Veh	Compare	Distance	Compare	Veh	Compare	Distance	Compare	Time
10	6816	32.22	5410	31.78	39.24	+21.8%	9164	+34.5%	30	-5.6%	5392	-0.3%	460
20	6777	31.54	5353	31.14	36.7	+16.4%	8968	+32.3%	29.6	-5%	5293	-1.1%	450
40	6790	31.70	5362	31.26	35.36	+11.5%	8837	+30.1%	29.12	-6.8%	5280	-1.5%	456

TABLE VI. COMPUTATIONAL RESULTS FOR N=100, MC=2*D

D	Reference [8] 's results				our algorithm's results								
	Initial		*Final*		*Initial*				*Final*				
	Distance	Veh	Distance	Veh	Veh	Compare	Distance	Compare	Veh	Compare	Distance	Compare	Time
10	3863	15.62	2890	15.5	17.65	+13%	7012	+81.5%	15.39	-0.7%	3193	+10%	290
20	3899	15.2	2852	15.02	16.4	+7.9%	6868	+76.2%	14.74	-1.8%	3056	+7%	307
40	3868	14.98	2816	14.84	15.88	+6%	6816	+76.2%	14.47	-2.5%	2996	+6.4%	303

Obviously, although our revised tour partitioning procedure obtained an initial solution worsen than reference [8], but our hybrid algorithm can obtain a better solution that reference [8].

Therefore, it is proved that our hybrid algorithm is capable of solving the VRPSPD effectively.

VI. CONCLUSION AND SUGGESTIONS

This research has dealt with the vehicle routing problem with simultaneous deliveries and pickups. A revised tour partitioning procedure has been proposed to obtain an initial solution. A hybrid heuristic based on the variable neighborhood search and reactive tabu search mechanism and some improvement procedures has been presented to improve the initial solutions. Computational results have shown that our algorithm is capable of obtaining solutions in good quality.

As for the future research, taking into account that the customers impose strict earliest and latest time deadline, that is, the hard time windows, is an important issues. The reactive tabu search mechanism could be incorporated by the technique of Adaptive Memory Programming.

REFERENCES

[1] H. Min, "The multiple vehicle routing problem with simultaneous delivery and pick-up points," Transportation Res, Oxford, vol. 23A, pp.377-386, September 1989.

[2] S. Salhi, and G. Nagy, "A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling," J Opl Res Soc, Basingstoke, vol. 50, pp.1034-1042, June 1999.

[3] J. Dethloff, "Vehicle routing and reverse logistics: the vehicle routing problem with simultaneous delivery and pickup," OR Spektrum. New York, vol. 23, pp. 79-96. February 2001.

[4] J. Dethloff, "Relation between vehicle routing problems: an insertion heuristic for the vehicle routing problem with simultaneous delivery and pick-up applied to the vehicle routing problem with backhauls," J Opl Res Soc. Basingstoke, vol. 53, pp.115-119, January 2002.

[5] J.F Chen, and T.H Wu, "Vehicle routing problem with simultaneous deliveries and pickups," J Opl Res Soc. Basingstoke, vol.57, pp. 579-587, May 2006.

[6] J. Crispim, and J. Brandao, "Metaheuristics applied to mixed and simultaneous extensions of vehicle routing problems with backhauls," J Opl Res Soc. Basingstoke, vol. 56, pp.1296-1302, November 2005

[7] F.A.T Montane, and R.D Galvao, "A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service", Computers &Operations Research. Oxford, vol. 33, pp. 595-619. March 2006.

[8] B. Nicola, and R. Giovanni, "Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery", Computers & Operations Research. Oxford, vol. 34, pp.578-594, February 2007.

[9] N.A Wassan, A.H Wassan, and G.Nagy, "A reactive tabu search algorithm for the vehicle routing problem with simultaneous pickups and deliveries," Journal of Combinational optimization. Dordrecht , vol 15, pp. 368-386 ,April 2008

[10] G. Mosheiov, "Vehicle routing with pick-up and delivery: tour-partitioning heuristics," Computer and Industrial Engineering. Oxford, vol. 34, pp.669-84, March 1998.

[11] R. Battiti, and G. Tecchioli, "Reactive tabu search," ORSA J Computing. Oxford, vol. 6, pp.126-140, June 1994.