Journal of Advances in Computer Research
Quarterly ISSN: 2008-6148
Sari Branch, Islamic Azad University, Sari, I.R.Iran
(Vol. 3, No. 4, November 2012), Pages: 55-66
www.jacr.iausari.ac.ir

An Improved Modified Tabu Search Algorithm to Solve the Vehicle Routing Problem with Simultaneous Pickup and Delivery

Majid Yousefikhoshbakht[1]✉, Farzad Didehvar[2], Farhad Rahmati[2]
(1) Mathematics and Computer Science, Young Researchers club, Hamedan Branch, Islamic Azad University, Hamedan, Iran
(2) Department of Mathematics and Computer Science, Amirkabir University of Technology, Tehran, Iran
khoshbakht@iauh.ac.ir; didehvar@iut.ac.ir; farhmati@iut.ac.ir

Received: 2012/11/14; Accepted: 2012/12/31

Abstract

The vehicle routing problem with simultaneous pickup and delivery (VRPSPD) is a well-known combinatorial optimization problem which addresses provided service to a set of customers using a homogeneous fleet of capacitated vehicles. The objective is to minimize the distance traveled. The VRPSPD is an NP-hard combinatorial optimization problem. Therefore, practical large-scale instances of VRPSPD cannot be solved by exact solution methodologies within acceptable computational time. Our interest was therefore focused on meta-heuristic solution approaches. For this reason, a modified tabu search (PA) is proposed for solving the VRPSPD in this paper. tComputational results on several standard instances of VRPSPD show the efficiency of the PA compared with other meta-heuristic algorithms.

Keywords: *Vehicle Routing Problem;* Tabu Search; NP-hard Problems*; Simultaneous Pickup and Delivery*

1. Introduction

The vehicle routing problem (VRP) has played an important role in supply chains, where such problems often arise in the first transportation step (to collect agricultural products, for instance) or in the final distribution phase toward customers. A typical VRP aims to find a set of tours taken by several vehicles in order to transport loads from a depot to a lot of customers and to return to the depot without exceeding the capacity constraints of each vehicle at minimum cost. Since the customer combination is not restricted to the selection of vehicle routes, VRP is considered as a combinatorial optimization problem where the number of feasible solutions for the problem increases exponentially with the increase in the number of customers. delivery and pickup

The vehicle routing problem with simultaneous pickup and delivery (VRPSPD) is a well-known problem in the area of network and operation research. In this problem, the vehicles are not only required to deliver goods to customers but also to pick some goods up at customer locations. It should be noted that the VRPSPD can be seen as a pickup and delivery problem (PDP) in the recent classification on static PDP. The VRPSPD is also called the multi-vehicle Hamiltonian one-to-many-to-one PDP with combined demands. By this definition, the deliveries are from a depot and the pickups will be returned to the depot. The customer demand is combined; which means that there is at least one customer with non zero pickup and delivery demand. The VRPSDP is characterized by the following: a fleet of identical vehicles located at the depot are used to serve customers distributed geographically in the area. The capacity of each vehicle is called Q. A customer requires a given shipment to be delivered and another load to be picked up during a single visit by a vehicle. The objective is to design a set of minimum cost routes to serve all customers so that the load on a vehicle is below vehicle capacity Q at each point on the route. An example of a single solution consisting of a set of routes constructed for a VRPSPD is presented in Figure 1.

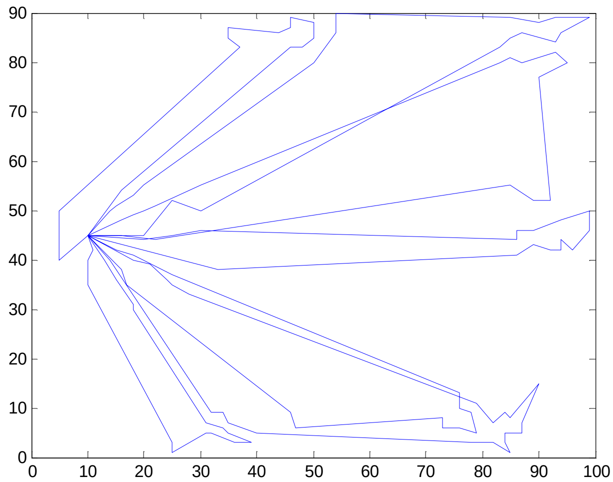

Figure 1. A solution of VRPSPD

It is noted that the VRPSDP is different from the other two related problems including VRP with Backhauls (VRPB) and CVRP. One obvious difference among CVRP, VRPB and VRPSDP is the variation of the vehicle load during the whole route. In the CVRP, the load decreases (increases) monotonously while in the VRPB, first, it decreases to zero and then starts to increase. Neither of the two cases is true in the VRPSDP. In this problem, the load of a vehicle varies irregularly and the maximum load may appear in the middle of the route. Furthermore, if the total demand of all the customers assigned to the same vehicle does not exceed the capacity limit in the CVRP, the feasibility of the route is always maintained, no matter what the visiting sequence is. There is a similar case in VRPB. However, it is quite different in the VRPSDP.

VRPSDP was introduced by Min [1] to solve a real life problem of transporting books between libraries. He used the cluster first and route second approach to solve the problem. A TSP problem is solved during the routing phase and if the TSP route violates the vehicle capacity constraint, the TSP problem is resolved by penalizing the arc violating the vehicle capacity constraint. The techniques used for solving the VRPSPD can be categorized into exact, heuristic and meta-heuristic methods. Exact approaches such as Lagrangean relaxation and branch and bound for solving the VRPSPD are successfully used only for relatively small problem sizes but they can guarantee optimality based on different techniques. These techniques use algorithms that generate both a lower and an upper bound on the true minimum value of the problem instance. If the upper and lower bound coincide, a proof of optimality is achieved. Dell'Amico et al. First proposed an exact method based on column generation, dynamic programming, and branch and price method for this problem [2]. However, the computational complexity of VRPSPD is evident from the computational result, in which an hour of computational time sometimes is not enough for solving a small size problem consisting of 40 customers.

VRPSPD is an NP-hard problem in the strong sense because when either all pickup demands or all delivery demands are set to zero, the problem reduces to the capacitated vehicle routing problem (CVRP) which is a known NP-hard problem. In other words, when the problem size is increased, the exact methods cannot solve it. Consequently, heuristic methods are used for solving these problems and they settle for the suboptimal solutions in a reasonable amount of time for instances with large sizes. Salhi and Nagy [3] solved VRPSDP by using the insertion-based heuristic method which they designed for vehicle routing problem with backhaul and mixed load (VRPBM). They also extended the VRPBM heuristic to multi-depot version of the problem. Dethloff [4] described the application of VRPSDP in reverse logistics. He proposed an insertion-based heuristic method which uses the concept of the residual load.

A new kind of heuristic algorithm which basically tries to combine basic heuristic methods in higher level frameworks which are aimed at efficiently and effectively exploring a search space has emerged in the last 30 years. Nowadays these methods are commonly called meta-heuristics. These algorithms have higher performance than exact and heuristic ones. As a result, nowadays they have received much attention from researchers and scientists in order to solve combinatorial optimization problems. Because of using the randomization concept in search for finding better solutions, this group of algorithms is more effective in escaping from local optimum and can obtain more quality solutions. That is why the recent publications are all based on meta-heuristic approaches. Tang and Galvao developed a tabu search (TS) algorithm to solve VRPSPD. In their formulation, the VRPSPD is formulated to minimize the total traveled distance of the route considering maximum distance and maximum capacity constraints on the vehicles [5]. Emmanouil et al proposed a hybrid solution approach for VRPSPD through incorporating the rationale of two well-known metaheuristics which have proven to be effective for routing problem variants, namely, TS and guided local search [6]. The performance of their metaheuristic algorithm was tested on benchmark instances involving 50 to 400 customers. It produced high quality results, improving several best solutions previously reported. Zhang et al developed a new scatter search approach for the one of the most important extensions of VRPSPD called stochastic travel time VRPSPD by incorporating a new chance constrained programming method [7]. They also proposed a genetic algorithm approach to this problem. In this paper, the Dethloff data will be used to evaluate the performance characteristics of both approaches. The computational results suggest that the scatter search solutions are superior to the genetic algorithm solutions.4-opt

VRPSPD In the following parts of this paper, the basic and the proposed idea are explained In Section 2. In Section 3, the PA is compared with some of the other algorithms on standard problems belonging to VRPSPD library. In Section 4, the conclusions are presented.

2. Solution method

Tabu search

Most engineering optimization algorithms are based on numerical linear and nonlinear programming methods which require substantial gradient information and usually seek to improve the solution in the neighborhood of a starting point. These algorithms, however, reveal a limited approach to complicated real-world optimization problems. If there is more than one local optimum in the problem, the result may depend on the selection of an initial point and the obtained optimal solution may not necessarily be the global optimum. Meta-heuristic algorithms have been one of the most important groups for solving combinatorial optimization problems. These algorithms designate a computational method which optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. Furthermore, the gradient search may become difficult and unstable when the objective function and constraints have multiple or sharp peaks. Meta-heuristics make few or no assumptions about the problem being optimized and can search very large spaces of candidate solutions. However, meta-heuristics such as memetic algorithm, simulated annealing, particle swarm optimization and ant colony optimization do not guarantee that an optimal solution will be found. These algorithms have been successfully applied to many difficult optimization problems including TSP, vehicle routing problem, quadratic assignment problem and job-shop scheduling problem, etc.

The TS is one of the most powerful meta-heuristic algorithms which has recently received much attention from researchers and scientists. This algorithm which was first proposed by Glover [8] is inspired by the principles of Artificial Intelligence, especially the use of memory. As a local search technique, TS moves from a current solution to the best solution in its neighborhood by exploring the solution space at each iteration. The main principle of the TS method is accepting neighboring solutions which deteriorate the current objective function value in order to escape from premature local optimum (Figure 2). The diagram pictured in Figure 3, is the coordinate plane. The horizontal line labeled "X" is the x-axis and the vertical line labeled "Y" is the y-axis.

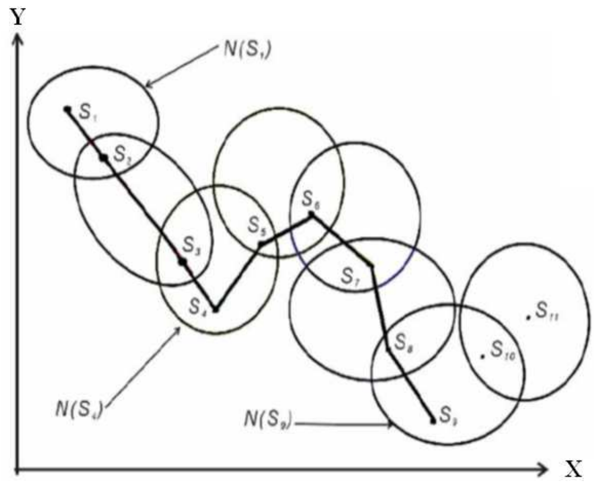

Figure 2. Escaping from local optimum

There are many important concepts in TS: tabu list, tabu tenure, aspiration criteria, neighborhood and neighborhood size, intensification, diversification, frequency-based memory, strategic oscillation, moves and evaluation of the moves which are all described as follows:

The tabu list is one of the ways of using memory by storing in a list the solutions explored throughout the search or, more commonly, some relevant attributes of these solutions. This tabu list has two main purposes: to prevent the return to the most recently visited solutions in order to avoid cycling and to drive the search towards the regions of the solution space which have not been explored yet but have great potential for containing good solutions. When a single attribute is marked as tabu, this typically results in more than one solution being tabu. Some of these solutions, which must now be avoided, could be of excellent quality and might not have been visited. To mitigate this problem, "aspiration criteria" are introduced. These criteria override a solution's tabu state, thereby including the otherwise-excluded solution in the allowed set. A commonly used aspiration criterion is to allow solutions which are better than the currently-known best solution. In operational terms, the TS consists of moving successively, in each iteration, from one solution S to the best solution in its neighborhood called N(S) which is not either in the tabu list or if it is in the list it satisfies some aspiration criteria.

The deterministic nature of the movements may cause cycling phenomena to occur. To avoid cycling, TS has declared Tabu the recently visited solutions or recently applied moves for a specific number of iterations (Tabu Tenure). When the search attempts to move towards a tabu declared state, this transition is forbidden, unless it improves the best solution ever encountered during the search (aspiration criterion).

A successful application of TS needs a powerful technique for search intensification and diversification. The intensification is a detailed exploration of some regions of the solution space which are usually in the vicinity of a good solution. The diversification is leading the search to the promising regions of the solution space which has been not explored yet. The successful application of TS requires a good balance between intensification and diversification.

A pseudo-code of TS for the TSP is presented on the Figure 3.

```
Tabu search
Input: s^0                                   // initial solution;
s ← s^0 ; s* ← s^0;
Given neighborhood function N(s), tabu list T(s) and aspiration condition A(s).
Repeat   //main cycle
Find the best feasible solution s′ ∈ {N(s) − T(s) + A(s)} ;
s ← s′;                          //replace the current solution by the new one.
If f(s) < f(s*) then s* ← s ;    //save the best so far solution.
Update tabu list;
        Update neighborhood function;
        Update aspiration condition;
Until termination criterion is satisfied.
```

Figure 3. Pseudo-code of the TS

The proposed algorithm

Now a modified tabu search (PA) is presented which uses a several local search algorithm as neghibourhood procedure. In our approach, three main modification respects to the TS have been made:

1. In order to be applied, the TS requires an initial solution. The Nearest Neighbor algorithm (NNA) is used to obtain initial solutions as one of the famous heuristic algorithms.
2. The PA comprises three types of neighborhood moves including 2-Opt, 0-1 and 1-1 exchanges. These moves are distinguished in terms of exchanges performed to convert one tour into another. As it will be explained later these moves are not equally performed in each iteration.
3. To improve the TS further, the size of tabu list is considered variable for intensification and diversification policies. The size of tabu list is considered a minimum value for the diversification policy and a maximum value for the intensification policy.

In order to be applied, the TS requires an initial solution. The Nearest Neighbor algorithm (NNA) is used to obtain initial solutions. This method produces a feasible solution that can be used as an initial solution for the TS. NNA is designed to be fast to compute and to provide a starting solution so that the TS may improve. First in this algorithm, the vehicle type is selected, beginning with type 1. If none of the unrouted customers is admissible in terms of the load carrying capacity of the vehicle, that type of vehicle is increased by a unit and this is repeated until at least one of the unrouted customers can be served by this type of vehicle. Afterwards, the route which is assigned to the chosen vehicle starts with the unrouted customer farthest from the depot. The next customer to be inserted in the route will be the one who has not been served yet and who is nearest to the customers of the route. That customer also has to be admissible in terms of the vehicle's load carrying capacity. The selected customer is inserted in the route before or after its NNA by taking into account the place which increases the travelling distance by the least amount. This process is repeated until no customer is admissible in the current route. When this happens a new vehicle is selected and the whole process is repeated until all the customers are routed. The GENIUS algorithm of [9] is composed of two procedures: GENI and US. The GENI is used to construct a TSP tour and the US to improve this tour. In this method, after constructing all the routes, the US is applied to each of the routes in order to try to reduce the travel distance. In this method, after constructing all the routes, the US is applied to each of the routes in order to try to reduce the travel distance.

The neighborhood structure is an important key feature in the performance of any TS because it determines the extent and the quality of the solution space explored. The PA comprises three types of neighborhood moves including 2-Opt, 0-1 and 1-1 exchanges. These moves are distinguished in terms of exchanges performed to convert one tour into another and they can be described as follows:

2-Opt move. The most commonly encountered move is the 2-Opt [10]. Suppose a single route consists of the following set of nodes n_1 in the given order $n_1=(0,1,\ldots,k,0)$, and let $A_1=\{(i,i+1); (j,j+1)\}$ is a set of two edges in n_1 which form a crisscross. 2-Opt move eliminated the crisscross and reversed a section of the route by deleting the branches (i,i+1), (j,j+1) and replacing them with its supplement (i,j), (i+1,j+1) to reconstruct the route. In multiple routes, edges (i,i+1), and (j,j+1) belong to different routes but they form a crisscross again. The 2-Opt move is applied exactly in the same way as in the case of multiple routes. The move is granted when it is considered favorable for the performance of the entire algorithm in terms of objectives and constraints. This move is demonstrated in Figure 4 [12].

0-1 Exchange move. This move transfers a node from its position in one route to another position in either the same or a different route. Consequently, while the initial tour is (0,…,i,i+1,…,j-2, j-1,j,j+1,…,0), the improved one is constructed as (0,…,i,j,i+1,…,j-2, j-1,j+1,…,0). The move is granted when it is considered favorable for the performance of the entire algorithm in terms of objective and constraints. This move is demonstrated in Figure 5 [12].

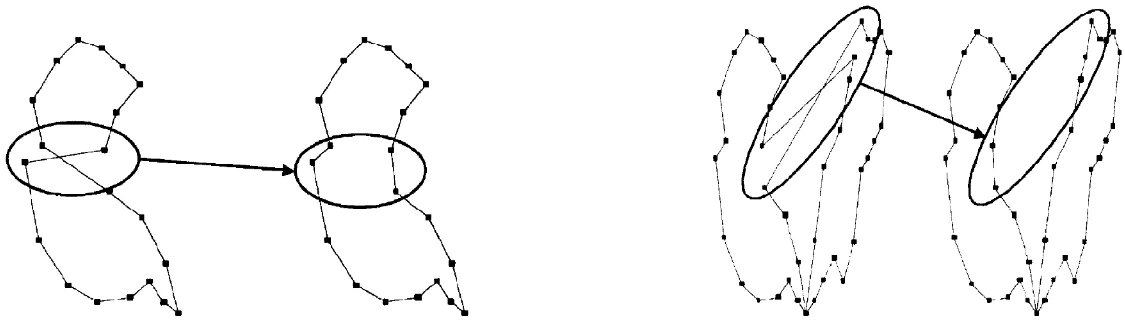

Figure 4. 2-Opt mover for single route (left) and for multiple routes (right)

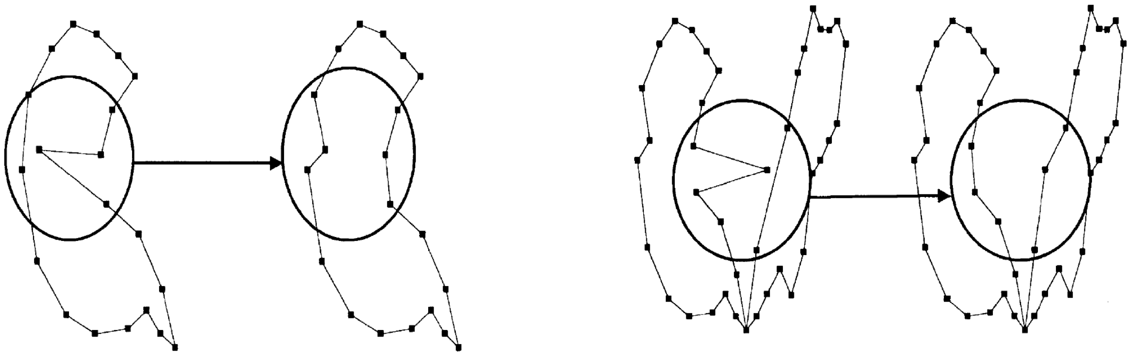

Figure 5. 0-1 Exchange mover for single route (left) and for multiple routes (right)

1-1 Exchange move. This move swaps two nodes from either the same or different routes. Consequently, if it is supposed that the initial tour consists of the set of nodes (0,…,i-1,i,i+1,…,j-1,j,j+1,…,0), the improved one is constructed as (0,…,i-1,j,i+1,…,j-1,i,j+1,…,0). The same procedure is conducted in the case of multiple routes. The move is granted when it is considered favorable for the performance of the entire algorithm in terms of objective and constraints. This move is demonstrated in Figure 6 [12].

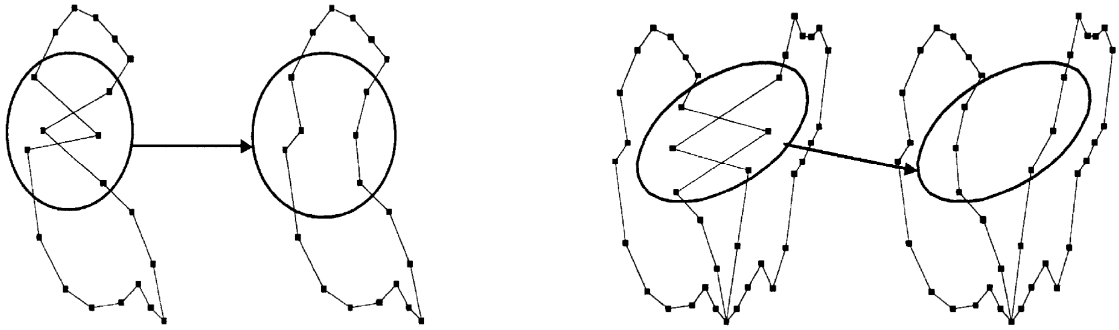

Figure 6. 1-1 Exchange mover for single route (left) and for multiple routes (right)

These moves are not equally performed in each iteration for two reasons: to diversify the search and to keep the computing time at reasonable levels. In each iteration all the customers are candidates to be moved. Therefore, *n* numbers of neighborhoods are produced by the mentioned algorithm in which 30, 35 and 35 percent of them belong to 2-Opt, 0-1 and 1-1 exchanges respectively (*n* is the number of nodes in each problem).

Changing the size of Tabu list may serve as a strong diversification technique. It works as follows: If the PA meta-heuristic has not updated a best solution for a pre-specified number of consecutive iterations, it must be driven the search towards a part of solution space which has not been explored yet (diversification policy). Thus, the length of Tabu list is incresed. After the diversification policy, the search process is intensified by decreasing the value of the Tabu list for a number of consecutive iterations. A pseudo-code of PA for the VRPSPD is presented on the Figure 4.

```
Build s^0 as initial solution by Nearest Neighbor algorithm (NNA);
s ← s^0 ;
s* ← s^0;             // the best know solution found;
Iter=0;               // number of iterations that the best solution of algorithm has not changed.
Consider 2-opt, 0-1 and 1-1 exchanges as neighborhood function N(s)
T list T(s)= ϕ;
Max length of tabu list=5;
Iter=0;
D=4;                  // if the best know solution of the algorithm has not changed for D iterations, the max length
tabu list is incresed.
Aspiration condition A(s) is satisfied if a better solution will be gained in compared to previous solutions.
Repeat   //main cycle
         Produce the n new solutions by using the 2-opt move (30 percent), 0-1 (35 percent) and 1-1
         exchnages (35 percent).
Find the best feasible solution s′ ∈ {N(s) − T(s) + A(s)} ;
Replace the current solution s by the new one s′;
If f(s) < f(s*) then
         s* ← s ;
         Max length of tabu list=5;
         Else
              Iter=iter+1
              If iter>D
                   Max length of tabu list=10;
                    Iter=0;
              End
         End
Update  list;
Until termination criterion is satisfied.
Output: s*
```

Figure 4. Pseudo-code of the PA

Parameter settings

Like any meta-heuristic algorithm, the solutions produced by the PA were dependent on the seed used to generate the sequence of pseudo-random numbers and on the different values of the search parameters of the algorithm. The parameter setting procedure is necessary to reach the best balance between the quality of the solutions obtained and the required computational attempt. It should be mentioned that there is no way of defining the most effective values of the parameters. Therefore, they were established based on perception and on experiments. The results confirm that our parameter setting works well. Also In addition, better solutions may exist. The most influential parameters of the PA and their values are listed in Table 1.

Table 1: Parameter setting for meta-heuristic method

Description	Value
The size of the tabu list	5-15
The size of the tabu Tenure	20
The number of initial solutions	1
The maximum number of consecutive iterations allowed when best solution has not been updated	10
The number of consecutive iterations when the diversification policy is implemented	6
The number of consecutive iterations when the intensification policy is implemented	6
The number of iterations after which the meta-heuristic algorithm terminate if fails to reach a new best solution	20
The maximum allowed running time	100

3. Computational experiments

The PA was coded in Matlab 7. All the experiments were implemented on a PC with Pentium 4 at 2.4GHZ and 2GB RAM and Windows XP Home Basic Operating system. Because the PA is a meta-heuristic algorithm, the results are reported for ten independent runs. In this section, the algorithm was tested on a set of VRPSPD benchmark problems with sizes ranging from 50 to 199 nodes. Some numerical results of comparison between the PA and the algorithms which obtained the best results in Salhi and Nagy's instances [13], namely those proposed by Chen and Wu (CW) [14], Wassan et al. (W) [15], Zachariadis et al. (Z) [16], Subramanian et al. 1 (S) [17] and Subramanian et al. 2 (PVND) [18] are presented.

Table 2 shows the results of the PA for the VRPSPD benchmark problem instances. In this table, Columns 2-6 show the problem size n, the number of vehicles v, the best known solutions (BKS), the best value result of the PA and the CPU time of the PA for the best value result over the ten runs for each problem. The left column indicates the percentage of PA improvement compared to the BKS (Gap).

$$\text{Gap}=100\times\ \frac{\left(\text{value of Algorithm- value of BKS}\right)}{\text{value of BKS}}$$

This table shows that the PA can be used to solve the VRPSPD effectively. From Table 2 it can be seen that among the 14 test problems, the maximum relative error is 2.72% and the average relative error is 0.34%.

Table 2. Results of the PA for 14 VRPSPD instances

Instance	n	v	BKS	Best	Time	Gap
CMT1X	50	3	466.77	**466.77**	3.12	0
CMT1Y	50	3	466.77	**466.77**	3.45	0
CMT2X	75	6	668.77	672.35	6.67	0.54
CMT2Y	75	6	663.25	672.35	6.87	1.37
CMT3X	100	5	721.27	**721.27**	15.90	0
CMT3Y	100	5	721.27	**721.27**	17.31	0
CMT12X	100	5	644.70	662.22	16.92	2.72
CMT12Y	100	5a	659.52	**659.52**	19.51	0
CMT11X	120	4	835.26	835.26	25.90	0
CMT11Y	120	4	830.39	**830.39**	27.73	0
CMT4X	150	7	852.46	**852.46**	43.61	0
CMT4Y	150	7	852.35	852.46	39.12	0.01
CMT5X	199	10	1029.25	**1029.25**	70.91	0
CMT5Y	199	10	1029.25	1030.55	73.22	0.13

Table 3 presents the comparison of the best results of our algorithm with other published Bio inspired research studies in terms of the optimal solution found. It is important to point out that Wassan et al. [15] may have used another approach to generate the instance CMT1Y. The optimum solution of this instance (466.77) was found by means of the mathematical formulation presented in [2]. This value is greater than the one obtained by Wassan et al. [15] (458.96). It should be noted that the optimum solution coincides with the solution found in [18] and by the PA.

Table 3. Comparison between PA and other metaheuristic algorithms

	CW [14]		W [15]		Z [16]		S [17]		PVND [18]		PA		BKS
	BS	V	BS	V	BS	V	BS	V	BS	V	BS	V	
CMT1X	478.52	3	468.30	3	469.80	3	466.77	3	466.77	3	466.77	3	466.77
CMT1Y	480.78	3	458.96a	3	469.80	3	466.77	3	466.77	3	466.77	3	466.77
CMT2X	688.51	6	668.77	6	684.21	6	684.21	6	684.21	6	672.35	6	668.77
CMT2Y	679.44	6	663.25	6	684.21	6	684.21	6	684.21	6	672.35	6	663.25
CMT3X	744.77	5	729.63	5	721.27	5	721.40	5	721.27	5	721.27	5	721.27
CMT3Y	723.88	5	745.46	5	721.27	5	721.40	5	721.27	5	721.27	5	721.27
CMT12X	678.46	6	644.70	5	662.22	5	662.22	5	662.22	5	662.22	5	644.70
CMT12Y	676.23	6	659.52	6	662.22	5	662.22	5	662.22	5	659.52	5	659.52
CMT11X	858.57	4	861.97	4	838.66	4	839.39	4	833.92	4	835.26	4	835.26
CMT11Y	859.77	5	830.39	4	837.08	4	841.88	4	833.92	4	830.39	4	830.39
CMT4X	887.00	7	876.50	7	852.46	7	852.83	7	852.46	7	852.46	7	852.46
CMT4Y	852.35	7	870.44	7	852.46	7	852.46	7	852.46	7	852.46	7	852.35
CMT5X	1089.22	10	1044.51	9	1030.55	10	1030.55	10	1029.25	10	1029.25	10	1029.25
CMT5Y	1084.27	10	1054.46	9	1030.55	10	1031.17	10	1029.25	10	1030.55	10	1029.25

The results of this comparison show that the PA gains worse solutions than the CW in CMT4Y and it gains better solutions than the CW in other problems. Furthermore, the results indicate that although the W obtains a better solution than the PA for CMT2X, CMT2Y and CMT12X, this algorithm cannot maintain this advantage in the rest of the examples and the PA yields better or equal solutions than this algorithm for other instances. The S in not a powerful algorithm for solving instances of VRPSPD and gains worse or equal solutions compared to the PA for all instances. Moreover, the computational experiments also show that in general the PA produces better results compared to PVND algorithms in terms of the quality of the solution and could find the best solutions for 8 of the 14 instances.

4. conclusion

In this paper, a new algorithm based on TS for solving the VRPSPD is discussed. The main idea is to give extra emphasis on the global-best and iteration-best solutions. Experiments are implemented to evaluate the algorithm's performance on some test instances from the literature. Computational results demonstrate that our algorithm is effective in solving VRPSDP. It seems that the combination of the PA and ant colony system or using strong local algorithms like Lin-kernigan can yield better results for the PA. This approach can be extended for further research not only to other types of routing problems including BVRP and school bus routing problem but also to more complex cases like assignment and scheduling problems.

5. acknowledgement

The authors would like to acknowledge the Young Researchers Club, Hamedan Branch, Islamic Azad University for the financial support of this work.

6. Reference

[1] Min, H., The multiple vehicle routing problem with simultaneous delivery and pick-up points. Transportation Research A 23(5), 1989, 377-386.

[2] Dell'Amico, M., Righini, G. and Salani, M., A branch-and-price approach to the vehicle routing problem with simultaneous distribution and collection, Transportation Science 40(2), 2006, 235-247.

[3] Salhi, S., Nagy, G., A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling. The Journal of the Operational Research Society 50(10), 1999, 1034.

[4] Dethloff, J., Vehicle routing and reverse logistics: the vehicle routing problem with simultaneous delivery and pick-up. OR Spektrum 23(1), 2001, 79.

[5] Tang, F. A., Galvao, R. D., A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service. Computers and Operations Research 33(3), 2006, 595-619.

[6] Emmanouil, E. Z., Christos, D. T., Chris. T. K., A hybrid metaheuristic algorithm for the vehicle routing problem with simultaneous delivery and pick-up service. Expert Systems with Applications 36(2), 2009, 1070-1081.

[7] Zhang, T., Chaovalitwongse, W. A., Zhang, Y., Scatter search for the stochastic travel time vehicle routing problem with simultaneous pickups and deliveries. Computers & Operations Research 39, 2011, 2277–2290.

[8] Glover, F., Future Paths for Integer Programming and Links to Artificial Intelligence. Computers and Operations Research 13, 1986, 533-549.

[9] Gendreau, M., Hertz, A., Laporte, G., New insertion and post optimization procedures for the traveling salesman problem. Operations Research 40 (6), 1992, 1086–1094.

[10] Croes, G., A method for solving traveling salesman problems. Operations Reaserch 6, 1958, 791-812.

[11] Waters, C. D. J., A solution procedure for the vehicle scheduling problem based on iterative route improvement. Journal of Operational Research Society 38, 1987, 833-839.

[12] Tarantilis, C. D., Kiranoudis, C. T., A meta-heuristic algorithm for the efficient distribution of perishable foods. Journal of Food Engineering 50, 2001, 1-9.

[13] Salhi, S., Nagy, G., A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling. Journal of the Operational Research Society 50(10), 1999, 1034–1042.

[14] Chen, J. F., Wu, T. H., Vehicle routing problem with simultaneous deliveries and pickups. Journal of the Operational Research Society 57(5), 2006, 579–587.

[15] Wassan, N. A., Wassan, A. H., Nagy, G., A reactive tabu search algorithm for the vehicle routing problem with simultaneous pickups and deliveries. Journal of Combinatorial Optimization15(4), 2008, 368–386.

[16] Zachariadis, E. E., Tarantilis, C. D., Kiranoudis, C. T., A hybrid metaheuristic algorithm for the vehicle routing problem with simultaneous delivery and pick-up service. Expert Systems with Applications 36(2), 2009, 1070–1081.

[17] Subramanian, A., CabralL, A. F., Ochi, L. S., An efficient ILS heuristic for the vehicle routing problem with simultaneous pickup and delivery. Technical Report, Universidade Federal Fluminense, 2008, available at http://www.ic.uff.br/ PosGraduacao/RelTecnicos/401.pdfS.

[18] Subramanian, A., Drummonda, L. M. A., Bentes, C., Ochi, L. S., Farias, R., A parallel heuristic for the Vehicle Routing Problem with Simultaneous Pickup and Delivery. Computers & Operations Research 37, 2010, 1899–1911.