Global Journal of Pure and Applied Mathematics.
ISSN 0973-1768 Volume 13, Number 6 (2017), pp. 1553-1567

http://www.ripublication.com

Solving CVRP by Using Two-stage (DPSOTS) Algorithm

Abdul-Gabbar Khaddar Bakhayt[1], Hassan A. AlSattar[2] and Iraq T. Abbas[3,4]

[1] Department of Statistics, College of Economic and Administration, University of Baghdad, Baghdad, Iraq.
[2] Departments of Artificial Intelligence, Faculty of Science, Universiti Pendelikon Sultan Idris, Malaysia.
[3] Departments of Mathematics, Faculty of Science, Universiti Putra Malaysia, Serdang, Selangor, Malaysia.
[4] Departments of Mathematics, Faculty of Science, University of Baghdad, Baghdad, Iraq.
Corresponding Author

Abstract

Capacitated vehicle routing problem (CVRP) is an NP-hard problem. For large-scale problems, it is quite difficult to achieve an optimal solution with traditional optimization methods due to the high computational complexity. A new hybrid approximation algorithm is developed in this work to solve the problem. In the hybrid algorithm, discrete particle swarm optimization Tabu Search (DPSOTS) using to solve multi-objective for two-stage for the optimal. The computational study showed that the proposed algorithm is a feasible and effective approach for capacitated vehicle routing problem, especially for large scale problems (i.e Our algorithm has a 70 ٪ and the other like DPSOGA 22.14٪, TS 34.14 ٪, DPSO 19.64٪ and GA 21.78 ٪).

Keyword: Capacitated Vehicle Routing Problem (CVRP), Discrete Particle Swarm Optimization (DPSO), Genetic Algorithm (GA), Tabu Search (TS).

INTRODUCTION

The vehicle routing problem (VRP), which was first introduced by Dantzig and Ramser (1959)**[8]**, is a well-known combinatorial optimization problem in the field of service operations management and logistics. The capacitated vehicle routing problem (CVRP) is an NP-hard problem for simultaneously determining the routes for several vehicles from a central depot to a set of customers, and then return to the depot without exceeding the capacity constraints of each vehicle. In practice, the problem is aimed at minimizing the total cost of the combined routes for a fleet of vehicles **[0]**. Since cost is closely associated with distance, in general, the goal is to minimize the distance travelled by a fleet of vehicles with various constraints. Many different approaches have been developed to solve the CVRP. In general, the approaches are divided into two classes: exact algorithms (Christofides et al., 1981**[10];** Toth and Vigo, 1998**[11])** and heuristic algorithms. Since the capacitated vehicle routing problem is an NP-hard problem (Laporte, 1992**[12]**), as Toth and Vigo (2002)**[13]** reported, no exact algorithm can consistently solve CVRP-instances with more than 50 customers; thus, the heuristic approaches are considered as reasonable choice in finding solutions for large-scale instances.

In [22], the purpose of this paper is to describe TABUROUTE, a new tabu search heuristic for the vehicle routing problem with capacity and route length restrictions.

In the paper [23] we consider, in a unified framework, both the symmetric and the asymmetric versions of the vehicle routing problem with backhauls, for which we present a new integer linear programming model and a Lagrangian lower bound which is strengthened in a cutting plane fashion.

In this paper, we introduce a very fast and easily implemented hybrid algorithm based on discrete particle swarm optimization (DPSO) and genetic (TS) algorithm. The proposed method uses DPSO to assign the customers on routes and TS algorithm to avoid becoming trapped in local optimum. And this work develops a Practical Swarm Intelligence-Genetic Algorithm (PSOTS) algorithm to solve a vehicle routing problem.

The problem of finding optimal routes for groups of vehicles, the Vehicle Routing Problem (VRP), belongs to the class of NP-hard combinatorial problems. The fundamental objectives are to find the minimal number of vehicles, the minimal travel time or the minimal costs of the travelled routes. In practice the basic formulation of the VRP problem is augmented by constraints such as e.g. vehicle capacity or time interval in which each customer has to be served, revealing the Capacitated Vehicle Routing

Problem (CVRP) and the Vehicle Routing Problem with Time Windows (VRPTW) respectively.

Lenstra and Rinnooy Kan [1] have analyzed the complexity of the vehicle routing problem and have concluded that practically all the vehicle routing problems are NP-hard (among them the classical vehicle routing problem), since they are not solved in polynomial time.

According to Solomon and Desrosiers [2], the vehicle routing problem with time windows (VRPTW) is also NP-hard because it is an extension of the VRP.VRP, it is still NP-hard (Dror and Trudeau, [3], Archetti et al., [4]). Therefore, the VRPTWSD is NP-hard, since it is a combination of the vehicle routing problem with time windows (VRPTW) and the vehicle routing problem with split delivery (VRPSD), and that makes a strong point for applying heuristics and meta-heuristic in order to solve the problem.

The sequence of the work is described next. Section 2 describes the literature review for VRPSD and its extensions. Section 3 presents the problem definition, including the Mathematical formulation. Section 4 describes the scatter search overview. Section 5 describes the heuristic and the scatter search approach proposed in order to solve the model. Section 6 presents the computational results. Finally, some conclusions are drawn in the last section.

2. MATHEMATICAL MODEL AND PROBLEM DESCRIPTION

The capacitated vehicle routing problem (CVRP) is a difficult combinatorial optimization problem, and generally can be described as follows: Goods are to be delivered to a set of customers by a fleet of vehicles from a central depot. The locations of the depot and the customers are given. The objective is to determine a viable route schedule which minimizes the distance or the total cost with the following constraints:

(1) Each customer is served exactly once by exactly one vehicle;

(2) Each vehicle starts and ends its route at the depot;

(3) The total length of each route must not exceed the constraint;

(4) The total demand of any route must not exceed the capacity of the vehicle. We can see the figure (2) below:

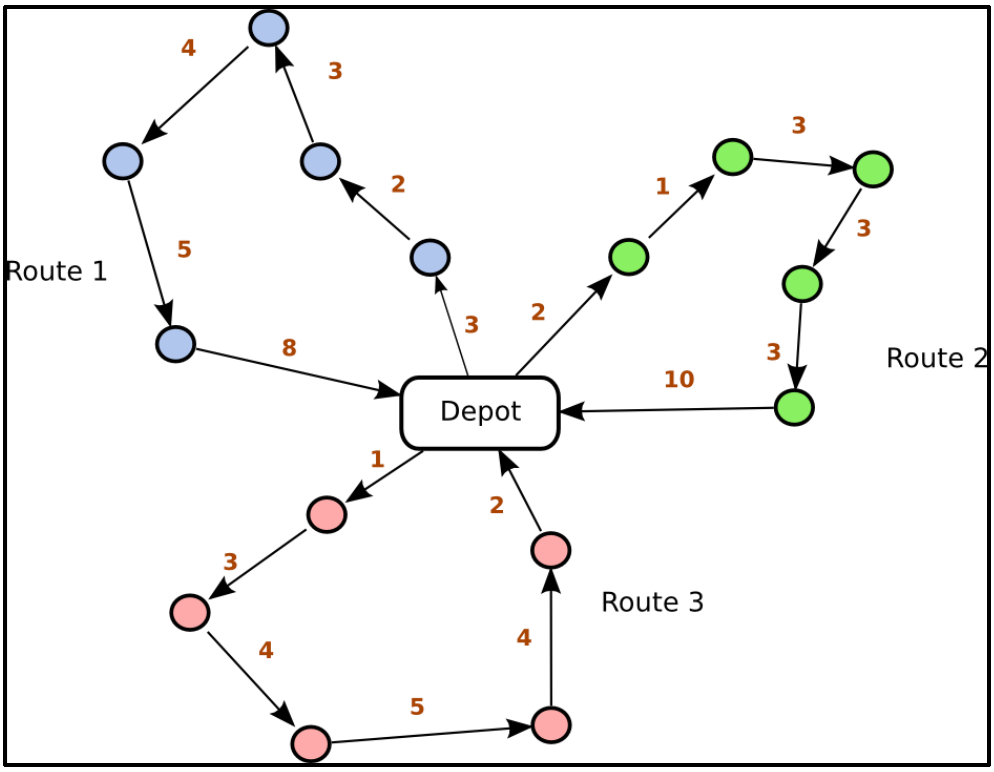

Figure (2): Capacitated vehicle routing problem for 15 costumers and 5 vehicles

Assume that the depot is node 0, and N customers are to be served by K vehicles. The demand of customer i is q_i, the capacity of vehicle k is Q_k, and the maximum allowed travel distance by vehicle k is Dk. Then the mathematical model of the CVRP based on the formulation given by Bodin et al.(1983)**[7]** is described

as follows:

$$\text{Minimize} \sum_{k=1}^{K}\sum_{i-0}^{N}\sum_{j=0}^{N} C_{ij}^{\ k} X_{ij}^{\ k} \qquad \ldots\ldots..(1)$$

subject to:

$$X_{ij}^{k} = \begin{cases} 1 \textit{if} \text{ vehicle travels from customer i to j}, \\ 0 \textit{otherwise} \end{cases} \qquad \ldots..(2)$$

$$\sum_{k=1}^{K}\sum_{i=0}^{N} X_{ij}^{\ k} = 1, \quad j=1,2,3,\ldots N \qquad \ldots..(3)$$

$$\sum_{k=1}^{K}\sum_{j=0}^{N} X_{ij}^{k} = 1, \quad i=1,2,3,\ldots N \qquad \ldots..(4)$$

$$\sum_{i=0}^{N} X_{it} - \sum_{j=0}^{N} X_{tj} = 0 \text{ , } k=1,2,3,\ldots,K;\ t=1,2,3,\ldots,N \qquad \ldots\ldots(5)$$

$$\sum_{i=0}^{N}\sum_{j=0}^{N} d_{ij}^{k} X_{ij}^{k} \le D_k,\ k=1,2,3,\ldots,K \qquad \ldots..(6)$$

$$\sum_{j=0}^{N} q_j \left(\sum_{j=0}^{N} X_{ij}^{k} \right) \le Q_k \text{ , } k=1,2,3,\ldots,K \qquad ....(7)$$

$$\sum_{j=0}^{N} X_{0j}^{k} \le 1 \text{ , } k=1,2,3,\ldots,K \qquad \ldots..(8)$$

$$\sum_{i=0}^{N} X_{i0}^{k} \le 1 \text{ , } k=1,2,3,\ldots,K \qquad \ldots..(9)$$

$$X_{ij}^{k} \in \{0,1\} \text{ , } I,j=0,1,2,\ldots,N \text{ ; } k=1,2,3,\ldots,K \qquad \ldots.(10)$$

Where N represents the number of customers, and K is the number of vehicles, and C_{ij}^{k} is the cost of travelling from customer i to customer j by vehicle k and D_{ij}^{k} is the travel distance from customer i to customer j by vehicle k.

2.1 Fitness function

Fitness is used to evaluate the performance of particles in the swarm. Generally, choosing a proper objective function as fitness function to represent the corresponding superiority of each particle is one of the key factors for successful resolution of the relevant .Therefore, according to the description in Section 2, we choose the following equation as fitness function: problem using DPSO algorithm. In the CVRP, the objective is to minimize the total cost or distance.

$$\text{Fit}=\sum_{k=1}^{K}\sum_{i=0}^{N}\sum_{j=0}^{N} C_{ij} X_{ij}^{\ k} \qquad \text{…..(14)}$$

The fitness function constraints are described in Section 2. The objective of the scheduling is to minimize the total cost, i.e. Fit, so the particle with the minimal fitness will outperform others and should be reserved during the optimization process.

The objective function Eq.(1) is to minimize the total cost by all vehicles. Constraints Eq.(3) and (4) ensure that each customer is served exactly once.

Constraint Eq.(5) ensures the route continuity. Constraint Eq.(6) shows that the total length of each route has a limit. Constraint Eq.(7) shows that the total demand of any route must not exceed the capacity of the vehicle. Constraints Eq.(8) and (9) ensure that each vehicle is used no more than once. Constraint Eq.(10) ensures that the variable only takes the integer 0 or 1.

3. DISCRETE PARTICLE SWARM OPTIMIZATION

3.1 Standard particle swarm optimization

Particle swarm optimization (PSO) is a parallel population-based computation technique proposed by Kennedy and Eberhart,1995**[18]**; which was motivated by the organisms behavior such as schooling of fish and flocking of birds. PSO can solve a variety of difficult optimization problems (Salman et al., 2002**[19]**; Shigenori et al., 2003**[20]**). PSO's major difference from genetic algorithm (GA) is that PSO uses the physical movements of the individuals in the swarm and has a flexible and well-balanced mechanism to enhance and adapt to the global and local exploration abilities, whereas GA uses genetic operators. Another advantage of PSO is its simplicity in coding and consistency in performance. The global optimizing model proposed by Shi and Eberhart (1999)**[21]** is as follows:

$$V_{id} = W\times V_{id} + C_1\times \text{Rand}\times(P_{best}-X_{id})+C_2\times \text{Rand}\times(G_{best}-X_{id}) \qquad \text{….(11a)}$$

$$X_{id}=X_{id}+V_{id}, \qquad \text{…..(11b)}$$

Where V_{id} is the velocity of particle i, X_{id} is the particle position, W is the inertial weight. C_1 and C_2 are the positive constant parameters, Rand and rand are the random functions in the range [0, 1], P_{best} is the best position of the ith particle and G_{best} is the best position among all particles in the swarm.

3.2 Discrete particle swarm optimization (DPSO)

Due to its global and local exploration abilities, simplicity in coding and consistency in performance, PSO algorithm has been widely applied in many fields although PSO algorithm was originally proposed for continuous optimization problems. In our research, we mainly use discrete data to process problems. Therefore, developing a mechanism to realize discrete optimization problem is attractive. We adopt the quantum discrete PSO algorithm proposed by Yang et al.(2004) **[9]**to solve the CVRP.

In quantum theory, a bit that is the minimum unit carrying information is always in a state of the range [0, 1]. A quantum particle vector is defined as follows:

$$V = [V_1, V_2, ..., V_M], ([1, 2, ..., N], V_i = v_i^1, v_i^2, \ldots, v_i^N),$$

where $0 \leq v_i^j \leq 1$,(i=1, 2, …, M; j=1, 2, …, N); N is the particle's length and M is the swarm size. V_i^j denotes the probability of the jth bit of the ith particle being 0. The following description is the rule from a quantum particle vector to a discrete particle vector.

Assume that $X=[X_1,X_2,\ldots,X_M]$ (=[$X_i=x_i^1,x_i^2,\ldots x_i^N$]) is the particle denotation for the practical problems. Where $x_i^j \in \{0,1\}$ (i=1, 2, …, M; j=1, 2, …, N) represents the corresponding discrete particle position of the quantum particle v_i^j, N is the particle's length and M is the swarm size. For each v_i^j (i=1, 2, …, M; j=1, 2, …, N), generate a random number in the range [0, 1]. If the random number is greater than v_i^j then $x_i^j=1$ otherwise $x_i^j=0$. DPSO algorithm can be described as follows:

$$Vlocalbest = \alpha * X_{localbest} + \beta * (1 - X_{localbest}) \quad \ldots(13a)$$

$$Vglobal = \alpha * X_{globalbest} + \beta * (1 - X_{globalbest}) \quad \ldots(13b)$$

$$V = w*V + c_1 + Vlocalbest + c_2\, Vglobal \quad \ldots(13c)$$

Where $\alpha+\beta=1$, $0<\alpha, \beta<1$ are control parameters which indicate the control degree of V. $w+c_1+c_2=1$, $0< w, c_1, c_2<1$. In Eq.(13c), the first part represents the inertia of previous probability; the second part is the "cognition" part, which represents the local exploration probability; the third part is the "social" part, which represents the cooperation among all quantum particles. So w, c_1 and c_2 represent the degree of the belief in oneself, local exploration and global exploration, respectively. The process of implementing the discrete PSO (DPSO) is described as follows:

Step 1: Initialize the quantum particles V and the

Discrete particles X.

Step 2: For discrete particles X, calculate the fitness.

Step 3: Calculate $V_{localbest}$ according to Eq.(13a).

Step 4: Calculate $V_{globalbest}$ according to Eq.(13b).

Step 5: Compute quantum probability V according to Eq.(13c).

Step 6: Calculate discrete particles X, If rand > v_i^j then xij=1;else x_i^j =0.

Step 7: Loop to Step 2 until one of the stopping criteria (generally, a sufficiently good fitness or the specified number of generations) is satisfied.

4. DPSO FOR CAPACITATED VEHICLE ROUTING PROBLEM

In this section, we describe the formulation of DPSO algorithm for the capacitated vehicle routing problem. How to encode a schedule is one of the key issues in successfully applying DPSO to the CVRP, namely, finding a suitable mapping from problem solution to DPSO particle. Consider the problem in which N customers are to be served by K vehicles, and then we can set up a search space of N×K dimensions every particle is composed of K sections and every section has N discrete points. The value of each discrete point is 0 or 1. If the value is 1, it represents that the corresponding customer is served by the relevant vehicle. The position of each particle indicates the relevant sequence of the customers served by each vehicle. We can give an example of the problem in which 8 customers are served by 2 vehicles. Fig.1 shows a stochastic particle position of this example.

A capacitated vehicle routing problem instance

(Customer, Vehicle)

(1,2), (2,1), (3,1), (4,2), (5,1), (6,2), (7,2), (8,1)

Mapping

	The first position	The second position
Dimension:	1 2 3 4 5 6 7 8	9 10 11 12 13 14 15 16
Position:	0 1 1 0 1 0 0 1	1 0 0 1 0 1 1 0

Figure (1) A DPSO particle mapping for the example

5. TABU SEARCH (TS)

Tabu Search (TS) is among the most cited and used met heuristics for CO problems. TS basic ideas were first introduced in Glover [1986], based on earlier ideas formulated in Glover [1977].7 Adscription of the method and its concepts can be found in Glover and Laguna [1997]. TS explicitly use the history of the search, both to escape from local minima and to implement an explorative strategy. We will first describe a simple version of TS, to introduce the basic concepts. Then, we will explain a more applicable algorithm and finally we will discuss some improvements. Figure (3) below have the details.

Procedure basic TS
$s \leftarrow GenerateInitialSolution()$
$TabuList \leftarrow \phi$
$while\ termination\ conditions\ not\ met\ do$
$s \leftarrow ChooseBestOf(N(s) \setminus TabuList)$
$Update(TabuList)$
$endwhile$

Figure 3. Algorithm: Simple Tabu Search (TS).

The simple TS algorithm applies a best improvement local search as basic ingredient and uses a short term memory to escape from local minima and to avoid cycles. The short term memory is implemented as a tabu list that keeps track of the most recently visited solutions and forbids moves toward them. The neighborhood of the current solution is thus restricted to the solutions that do not belong to the tabu list. In the following we will refer to this set as allowed set. At each iteration the best solution from the allowed set is chosen as the new current solution. Additionally, this solution is added to the tabu list and one of the solutions that were already in the tabu list is removed (usually in a FIFO order). Due to this dynamic restriction of allowed solutions in a neighborhood, TS can be considered as a dynamic neighborhood search technique

[St ¨ utzle 1999b]. The algorithm stops when a termination condition is met. It might also terminate if the allowed set is empty, that is, if all the solutions in N(s) are forbidden by the tabu list. The use of a tabu list prevents from returning to recently visited solutions; therefore it prevents from endless cycling and forces the search to accept even uphill moves. The length l of the tabu list (i.e., the tabu tenure) controls the memory of the search process. With small tabu tenures the search will concentrate on small areas of the search space. On the opposite, large tabu tenure forces the search process to explore larger regions, because it forbids revisiting a higher number of solutions. The tabu tenure can be varied during the search, leading to more robust algorithms. An example can be found in Taillard [1991], where the tabu tenure is periodically reinitialized at random from the interval [lmin, lmax]. A more advanced use of a dynamic tabu tenure is presented in Battiti and Tecchiolli [1994] and Battiti and Protasi [1997], where the tabu tenure is increased if there is evidence for repetitions of solutions (thus a higher diversification is needed), 8Strategies for avoiding to stop the search when the allowed set is empty include the choice of the least recently visited solution, even if it is tabu. 9Cycles of higher period are possible, since the tabu list has a finite length l which is smaller than the cardinality of the search space. While it is decreased if there are no improvements (thus intensification should be boosted).

6. DPSO-TS OPTIMIZATION ALGORITHM

By combining DPSO with TS algorithm, we can get a new hybrid optimization approach DPSO-TS. DPSO has strong global search ability but, as a stochastic search algorithm, cannot guarantee to converge to the global optimal solution at the end. TS algorithm uses a certain probability to avoid becoming trapped in a local optimum. This hybrid approach makes full use of the strong global search ability of DPSO and the strong local search ability of TS and offsets the weaknesses of each other. The algorithm is shown in Fig.4.

Begin

Step1: Initialize parameters: swarm size, maximum of generation,

α, β, w, c1, c2; Set t0, tf, λ, R by experiment; Generation:=1;

Step2: Initialize quantum particles V and discrete particles X;

Step3: Evaluate each particle’s fitness according to Eq.(14);

Step4: Obtain $X_{\text{globalbest}}$ and $X_{\text{localbest}}$;

Repeat

Step5: Compute $V_{\text{localbest}}$ according to Eq.(13a) and compute $V_{\text{globalbest}}$ according to Eq.(13b);

Step6: Compute quantum probability $\boldsymbol{V}$ according to Eq.(13c);

Step7: Obtain $\boldsymbol{X}$ according to $\boldsymbol{V}$;

Step8: If ($> j$) j =1;

Step9: **If (** $rand > V_i^j$)$x_i^j = 1$;

Step10: Else x_i^j =0;

Step11: Compute each particle's fitness according to Eq.(14);

Step12: Find new $X_{globalbest}$ and $X_{localbest}$ and update $X_{globalbest}$ and $X_{localbest}$;

Step13: Carry out SA subprogram on each route of each particle;

Step14: Compute the particle's fitness according to Eq.(14);

Step15: Update $X_{globalbest}$ and $X_{localbest}$;

Step16: Generation=Generation+1 .Until (one of termination conditions is satisfied)

Step17: Output the optimization results;

End

Tabu Search algorithm subprogram (for each route of each particle)

```
s ← s0
 s Best ← s
 Tabu List ← []
While (not stopping Condition ())
        Candidate List ← []
        Best Candidate ← null
        For (s Candidate in s Neighborhood)
If (  (not tabuList.contains(s Candidate)) and (fitness(s Candidate) > fitness(best Candidate)) )
                        best Candidate ← s Candidate
                end
        End
        s ← best Candidate
        If (fitness (best Candidate) > fitness(s Best))
                s Best ← best Candidate
        End
        tabuList.push (best Candidate);
        if (tabuList.size > maxTabuSize)
```

tabuList.removeFirst ()

End

End

Return s Best

Figure 4. DPSO-TS hybrid optimization algorithm

Table 1

Instance		GA				DPSO				TS				DPSO-TS				DPSO-GA			
Name	Optima	Min	Max	Mean	No.c	Min	Max	Mean	No.c	Min	Max	Mean	No.c	Min	Max	Mean	No.c	Min	Max	Mean	No.c
A-n32-k5	784	809	855	833	0	966	1059	998.5	0	801	845	825.1	0	**796**	**808**	**798.6**	0	801	871	830.3	0
A-n33-k5	661	668	713	686.2	0	733	887	819.7	0	**661**	685	671	4	**661**	**671**	**663**	**8**	**661**	702	684.6	1
A-n45-k7	1146	1252	1310	1276.8	0	1434	1547	1496.5	0	1164	1226	1185.3	0	**1158**	**1196**	**1170.2**	0	1213	1299	1257.6	0
A-n46-k7	914	1020	1109	1071.5	0	1030	1111	1063.4	0	**914**	1001	953.9	2	**914**	**959**	**920.6**	**4**	944	1067	1009.7	0
B-n31-k5	672	692	710	698.4	0	708	745	733.9	0	**672**	694	685.5	2	**672**	**686**	**676.5**	**3**	677	697	687.4	0
B-n35-k5	955	970	1002	990.5	0	1050	1177	1109.8	0	**955**	985	964.5	**2**	**955**	**979**	**962**	1	971	1011	992.1	0
B-n45-k5	751	776	842	796.8	0	992	1091	1050.6	0	**751**	798	765.1	2	**751**	**752**	**751.4**	**6**	772	835	791.3	0
B-n45-k6	678	742	776	762.1	0	887	963	936.1	0	690	727	702.9	0	**678**	**693**	**686.2**	**1**	717	792	753.3	0
E-n22-k4	375	**375**	382	378.4	3	392	462	437.6	0	**375**	382	377.7	5	**375**	**375**	**375**	**10**	**375**	387	380.3	4
E-n30-k3	506	**506**	544	520.5	1	556	586	569	0	**506**	537	521.6	4	**506**	**506**	**506**	**10**	**506**	544	529.6	3
E-n33-k4	835	841	864	853.6	0	921	966	938.8	0	**835**	873	846.4	3	**835**	**837**	**835.2**	**9**	839	868	853.6	0
E-n51-k5	521	564	604	580.3	0	685	759	723.2	0	**521**	547	534.2	2	**521**	**529**	**521.8**	**9**	528	579	558.6	0
E-n76-k7	683	785	901	847.1	0	769	839	811.4	0	703	723	711.5	0	**690**	**703**	**696**	0	774	812	795.7	0
E-n76-k8	735	871	963	918.9	0	864	974	929.5	0	757	785	768.3	0	**739**	**753**	**747.9**	0	821	902	862.7	0
E-n101-k8	817	1047	1109	1071.7	0	1033	1093	1061.1	0	834	867	849.6	0	**833**	**855**	**846**	0	960	1060	1022.1	0
F-n45-k4	724	737	791	757.3	0	732	761	738.9	0	**724**	745	732.1	2	**724**	**728**	**724.4**	9	735	775	753.2	0
F-n72-k4	237	252	275	263.3	0	242	267	257.9	0	**237**	244	240.1	3	**237**	**240**	**237.6**	6	247	276	261.9	0
P-n16-k8	**450**	**450**	**450**	**450**	**10**	**450**	**450**	**450**	**10**	**450**	**450**	**450**	**10**	**450**	**450**	**450**	**10**	**450**	**450**	**450**	**10**
P-n19-k2	**212**	**212**	**212**	**212**	**10**	**212**	**212**	**212**	**10**	**212**	**212**	**212**	**10**	**212**	**212**	**212**	**10**	**212**	**212**	**212**	**10**
P-n20-k2	**216**	**216**	**216**	**216**	**10**	**216**	**216**	**216**	**10**	**216**	**216**	**216**	**10**	**216**	**216**	**216**	**10**	**216**	**216**	**216**	**10**
P-n21-k2	**211**	**211**	**211**	**211**	**10**	**211**	**211**	**211**	**10**	**211**	**211**	**211**	**10**	**211**	**211**	**211**	**10**	**211**	**211**	**211**	**10**
P-n22-k2	**216**	**216**	**216**	**216**	**10**	**216**	**216**	**216**	**10**	**216**	**216**	**216**	**10**	**216**	**216**	**216**	**10**	**216**	**216**	**216**	**10**

P-n22-k8	590	590	632	597.6	6	590	603	598.1	4	590	623	593.3	9	**590**	**590**	**590**	**10**	590	646	614.7	2
P-n23-k8	529	529	549	537.6	1	529	569	541.9	1	529	537	531.2	6	**529**	**529**	**529**	**10**	529	570	539.1	2
P-n40-k5	458	469	500	483.2	0	466	489	477.8	0	459	486	469	0	**458**	**459**	**458.3**	**7**	475	519	491.6	0
P-n45-k5	510	529	564	547	0	513	554	536.9	0	515	525	519.5	0	**510**	**513**	**510.6**	**8**	522	567	536.9	0
P-n50-k7	554	588	667	637.5	0	608	645	630.9	0	561	588	573.7	0	554	558	556.4	1	583	627	605	0
P-n50-k8	631	699	778	732.2	0	722	780	753.5	0	630	655	642.7	1	629	640	632.1	2	676	725	698	0

7. COMPUTATIONAL RESULTS AND FUTURE RESEARCH

To illustrate the effectiveness and good performance of the proposed algorithm, various kinds of benchmark instances with different sizes have been selected for the computation. We programmed the algorithms in Matlab 6.5 and ran them on Mobile Intel Pentium IV CPU 1.80 GHz with 256 M RAM.

Here we can see from the table (1) that the new hyper algorithm (DPSOTS) has been a good result compares with the other algorithms. Also we can see it’s have a better optimal solution for almost the test problems. As a result the new hyper algorithm (DPSOTS) could be depend to solve CVRP for two-stage. Experimental results demonstrate that the proposed algorithm produced **DPSOTS 54٪, DPSOGA 22.14 ٪, TS 34.14 ٪, DPSO 19.64 ٪ and GA 21.78 ٪,** that mean our algorithm is faster than the others to reach optimal solution.

Application of the proposed DPSOTS in a larger problem and testing its validity to an industry will be our upcoming work.

REFERENCES

[1] Gen-ke, CHEN Ai-ling YANG, and W. U. Zhi-ming. "Hybrid discrete particle swarm optimization algorithm for capacitated vehicle routing problem."

[2] Lenstra, J K and A H G Rinnooy Kan (1981), Complexity of Vehicle and Scheduling Problems, Networks 11, 221-227.

[3] Solomon, M M and J Desrosiers (1988), Time Window Constrained Routing and Scheduling Problem, Transportation Science 22, 1-13.

[4] Dror, M and P Trudeau (1990), Split Delivery Routing, Naval Research Logistics 37, 383- 402.

[5] Archetti, C, M Mansini and M G Speranza (2005), Complexity and Reducibility of the Skip Delivery Problem, Transportation Science 39, 182-187.

[6] Bodin, L., Golden, B.L., Assad, A., Ball, M.O., 1983. The state of the art in the routing and scheduling of vehicles and crews. Computers & Operations Research, 10:69-221.

[7] Dantzig, G.B., Ramser, R.H., 1959. The truck dispatching problem. Management Science,6:80-91.

[8] Yang, S.Y., Wang, M., Jiao, L.C., 2004. A Quantum Particle Swarm Optimization. Proceeding of the 2004 IEEE Congress on Evolutionary Computation, 1:320-324.

[9] Christofides, N., Mignozzi, A., Toth, P., 1981. Exact algo tree and shortest path relaxations. Mathematical Programming, 20(1):255-282. [doi:10.1007/BF01589353]

[10] Toth, P., Vigo, D., 1998. Exact Solution of the Vehicle Routing Problem. In: Crainic, T.G., Laporte. (Eds.), Fleet Management and Logistics. Kluwer Academic Publishers, Norwell, MA, p.1-31.

[11] Laporte, G., 1992. The vehicle routing problem: an overview of exact and approximate algorithms. European Journal of Operational Research, 59(3):345-358. [doi:10.1016/ 0377-2217(92)90192-C]

[12] Toth, P., Vigo, D., 2002. The Vehicle Routing Problem. SIAM Monographs on Discrete Mathematics and Applications, Vol. 9. SIAM, Philadelphia, PA.

[13] Kennedy, J., Eberhart, R., 1995. Particle Swarm Optimization. Proceeding of the 1995 IEEE International Conference on Neural Network, p.1942-1948.

[14] Salman, A., Ahmad, I., Madani, S.A., 2002. Particle swarm Optimization for task assignment problem. Microprocessors and Microsystems,26(8):363-371.[doi:10.1016/S0141-9331(02)00053-4]

[15] Shigenori, N., Takamu, G.J., Toshiki, Y., Yoshikazu, F., 2003. A hybrid particle swarm optimization for distribution state estimation. IEEE Trans. on Power Systems,18(1):60-68. [doi:10.1109/TPWRS. 2002.807051]

[16] Shi, Y., Eberhart, R., 1999. Empirical Study of Particle Swarm Optimization. Proceedings of Congress on Evolutionary Computation, p.1945-1950.

[17] Gendreau, Michel, Alain Hertz, and Gilbert Laporte. (1994):"A tabu search heuristic for the vehicle routing problem." Management science 40.10 :1276-1290.

[18] Toth, Paolo, and Daniele Vigo. (1997) "An exact algorithm for the vehicle routing problem with backhauls." Transportation science 31.4: 372-385.

[19] Mingozzi, Aristide, Roberto Roberti, and Paolo Toth. "An exact algorithm for the multitrip vehicle routing Problem." INFORMS Journal on Computing 25.2 (2013): 193-207.

[20] Pan, Tinglei, Haipeng Pan, and Jingfei Gao. "An improved ant colony algorithm based on vehicle routing Problem." Control Conference (CCC), 2015 34th Chinese. IEEE, 2015.

[21] Toth, Paolo, and Daniele Vigo. "A heuristic algorithm for the symmetric and asymmetric vehicle routing Problems with backhauls." European Journal of Operational Research 113.3 (1999): 528-543.

[22] Baldacci, Roberto, Paolo Toth, and Daniele Vigo. "Exact algorithms for routing problems under vehicle capacity constraints." Annals of Operations Research 175.1 (2010): 213-245.

[23] Zhou, Ying, and Jiahai Wang. "A local search-based multiobjective optimization algorithm for multiobjective vehicle routing problem with time windows." IEEE Systems Journal 9.3 (2015): 1100-1113.

[24] Showkat, Faiz Faiq. "A Modified ANT System Optimization Algorithm for the Capacitated Vehicle Routing Problem." Journal of Al-Nahrain University Science 14.1 (2011): 161-170.

[25] Toth, Paolo, and Daniele Vigo, eds. Vehicle routing: problems, methods, and applications. Vol. 18. Siam, 2014.