International Journal of Industrial Engineering & Production Research

International Journal of Industrial Engineering & Production Research

December 2014, Volume 25, Number 4
pp. 287-295

http://IJIEPR.iust.ac.ir/

pISSN: 2008-4889

A Quantum Evolutionary Algorithm for the Vehicle Routing Problem with Delivery Time Cost

*A.K. Beheshti & S.R. Hejazi**

Ali Kourank Beheshti, *Ph.D student, Department of Industrial and Systems Engineering, Isfahan University of Technology, Isfahan,84156-83111, Iran*
Seyed Reza Hejazi, *Associate professor, Department of Industrial and Systems Engineering, Isfahan University of Technology, Isfahan,84156-83111, Iran*

KEYWORDS

The vehicle routing problem with delivery time cost, Column generation, Quantum evolutionary algorithm

ABSTRACT

We consider the Vehicle Routing Problem with Delivery Time Cost (VRPDTC) in which aims to find a set of routes of minimal total costs including the travelling cost and delivery time cost, starting and ending at the depot, in such a way that each customer is visited by one vehicle given the capacity of the vehicle to satisfy a specific demand. In this paper, first a mathematical model of this problem is presented. Then a column generation approach is employed to obtain the lower bounds of problems. In order to solve this problem, a quantum evolutionary algorithm based on quantum computations is proposed. Finally, the computational results of some instances are reported and the results of these approaches are compared. The results demonstrate the effectiveness of the quantum algorithm in solving this problem.

1. Introduction

The Vehicle Routing Problem (VRP) is a well-known combinatorial optimization problem (COP) with a wide range of applications. It deals with the determination of minimum cost routes from a central depot to a set of geographically dispersed customers. The Vehicle Routing Problem with Time Window (VRPTW) is an extension of the VRP where another constraint is added, requiring the start of service at each customer within a time window.

A Time Window (TW) is defined as the time interval within which a vehicle has to arrive at a node, and it is usually characterized by an early arrival time (EAT) and a late arrival time (LAT)[1]. The time window can be specified in terms of its being either a single-sided or a double-sided window. In a single-sided time window, the pickup points usually specify the deadlines by which they must be serviced. In double-sided time window, however, both the earliest and the latest service times are imposed by the nodes[2]. In fact, in single-sided time window, EAT is equal to LAT.

As mentioned earlier, customer service level affects both customer satisfaction and distribution costs. Consequently, time windows must be set by taking into account both customer satisfaction and distribution costs. Distribution costs include such costs as fuel cost, vehicle cost, driver salary, etc. These costs are usually considered in vehicle routing problem. VRP is one of the most applicable problems in industries and plays a pivotal role in logistics. In VRPTW, delivery locations

* **Corresponding author: Seyed Reza Hejazi**
Email: **rehejazi@cc.iut.ac.ir**
Paper first received Apr. 28, 2013, and in accepted form Jan, 09, 2014.

have predefined time windows, within which the deliveries (or visits) must be made.

In classic VRPTW models, however, it was assumed that time windows have been determined by distributer or customers, and at the operational level, they are considered as an input parameter for the VRPTW model. However, in order to determine optimal time windows, an integrated and comprehensive model is required in which single-sided time window setting (due date assignment) decisions are integrated with routing decisions.

The problems with due date determination in the field of scheduling problems have received considerable attention in the last years due to the introduction of new methods of inventory management such as just-in-time (JIT) concepts.

In JIT systems, jobs are to be completed neither too early nor too late which leads to the scheduling problems with both earliness and tardiness costs and assigning due dates. The due date assignment problems make practical sense when a firm offers a due date to its customers during sale negotiations and has to offer a price reduction when the due date is far away from the expected one[3].

In traditional machine scheduling models, due dates are considered as given by exogenous decisions. In an integrated supply chain, in order to avoid tardiness penalties, due dates may be determined by taking into account the system’s ability to meet the assigned due dates as part of the scheduling process. The ability to control due dates can be a major factor in improving system performance. Of course, extending the due date for an order will also have costs associated with it, what is usually called the due date assignment cost. This may come, for example, from price discounts the supplier may have to offer in order to retain the customer[4].

In the present paper, an integrated mathematical model, a column generation algorithm and a quantum algorithm are proposed for single-sided time window setting. In the second section the related literature is briefly reviewed. This is followed by a presentation of the model in section three and a column generation approach in section four. Section five presents the quantum evolutionary algorithm.

The computational results of the study are discussed in section six and the paper ends with the concluding remarks in the seventh section.

2. Literature Review

As mentioned earlier, vehicle routing problem with time window (VRPTW) is an extension of the VRP where another constraint is added requiring the start of service at each customer within a time window. If the time window constraints must be satisfied strictly, such problem is called the Vehicle routing problem with hard time window (abbreviated as VRPHTW). The Vehicle routing problem with soft time window (abbreviated as VRPSTW) is a relaxation of the VRPHTW; in the former, time windows can be violated if a penalty is paid and this penalty is often assumed to be linear with the degree of violation([5-10]); in the latter violations are infeasible[11, 12]. In some cases, violation of time windows does not directly incur any penalty cost, although the satisfaction levels of customers (the service level of suppliers) may drop and lead to profit loss in the long term. In these cases, researchers usually apply fuzzy theory to the routing problem([13-15]).

The delivery time window setting problem was noticed for dispatching repairmen problem by Madesn *et al*[16]. In this problem, requests for service that would arrive during a single week were scheduled to be serviced during the forthcoming week. The challenge in this scheduling is to determine and commit to a particular delivery time window that will lead to efficient routing solutions. They proposed a heuristic method called BARTOC (Booking Algorithm for Routing and Timing of Customers) that was based on cluster-first, route-second approach[16]. This problem is placed in attended home delivery class. The delivery time window setting has been also noticed in attended home delivery and E-Fulfillment. In these cases, a 'Time Slot' is usually utilized instead of a 'Time Window'. Attended delivery could be essential for security reasons (e.g. electronics), or when goods are perishable (e.g. groceries) or physically large (e.g. furniture), or when a service is performed (e.g. repairing or product installation)[17]. Campbell and Save lsbergh [18] proposed the Home Delivery Problem (HDP) and solved it by a heuristic algorithm. In this problem, the vendor has to decide whether to accept it or not, and, if he accepts it, he has to decide which time slot should be used to guarantee delivery. They studied dynamic routing and demonstrated the significance of considering the opportunity cost associated with accepting a delivery request in a certain time slot.

Jabali et al. [19] proposed Self-Imposed Time Windows (SITW) when a logistics service provider quotes a delivery time window to his customer. They described how a VRP with SITW and stochastic travel times can benefit from time buffers and developed a hybrid LP/tabu search algorithm for producing high-quality solutions. Ibaraki et al. [20] proposed local search algorithms for the vehicle routing problem with soft time window constraints. The time window constraint for each customer was treated as a penalty function, which was very general in the sense that it could be non-convex and discontinuous as long as it was piecewise linear. They used local search to assign customers to vehicles and to find orders of customers for vehicles to visit. After fixing the order of customers for a vehicle to visit, they proposed a dynamic programming algorithm to efficiently compute the optimal start times of services for customers in a given route. Ibaraki *et al. [21]* treated the time window constraint for each customer as a penalty function, and

assumed that it was convex and piecewise linear. Given the order of customers each vehicle has to visit, dynamic programming was used to determine the optimal start time to serve the customers so that the total time penalty was minimized. Tas et al.[22] considered the vehicle routing problem with flexible time window (VRPFlexTW) which enables to serve customers outside their original time boundaries with respect to a given tolerance. Compared to the VRPTW, the VRPFlexTW permits fixed deviations from customer time windows at a cost. Their solution procedure comprised three main components: initialization, routing and scheduling.
However, in this paper due times is set with regard to routing and delivery time window costs. This approach can be used when information of all customers such as demand are known. Consequently, it cannot be utilized in some e-fulfillment services where time window must be immediately determined in customer call time. However, in other services, it can be used and it can determine optimal delivery time window.
In the following section, an integrated model and a column generation algorithm are presented and due to the problem complexity, a quantum evolutionary algorithm is proposed.

3. Mathematical Model

In this section a mathematical model is presented for the vehicle routing problem with delivery time cost. The customer and distributer behaviors are described below.
Distributer must provide service to some customers. Each customer calls the distributer and states his demand. After all the customers call, the distributer starts routing and determining delivery due date while considering fuel cost and delivery time cost. After the distributer determines the due dates, the customers accept the delivery due date. The assumptions of this model are summarized below:

1. A single depot and a homogeneous fleet are considered.
2. A single product for distribution is considered.
3. Planning is worked out for one day.
4. For each customer a delivery time cost is defined which can be equal for all customers.
5. Each customer is visited by exactly one vehicle.
6. Customers accept the delivery due date that the distributer has set.
7. No disruptions during travel might occur due to weather, human, or other unexpected factors.

This problem is called 'Vehicle Routing Problem with Delivery Time Cost' (VRPDTC) and is defined as: “given a set of customers, a set of vehicles and a depot, the VRPDTC is to find a set of routes of minimal total cost, starting and ending at the depot, in such a way that each customer is visited by one vehicle with regard to the capacity of the vehicle to satisfy a specific demand”. Total cost includes fuel costs and delivery time costs.
More formally, the VRPDTC is defined on a directed graph G = (V,A), where $\mathrm{V} = \{v_0, \dots, v_n\}$ is the set of nodes and A is the set of arcs. Vertex v_0 is a depot node; other vertexes are customer nodes.
Non-negative travel time t_{ij} is associated with each arc(v_i, v_j), which satisfies the triangle inequality. Each customer $v_i \in V \backslash \{v_o\}$ has a demand d_i. Furthermore, let γ and δ_i be the per time unit cost for fuel and due date, respectively. A homogeneous fleet of vehicle with capacity q is available to serve the customers.
Parameters and decision variables of the mathematical model are displayed in Table 1.

Table 1. The parameters and decision variables of the proposed model

Parameter/ Decision Variables	Definition
K	Maximum number of vehicles
q	Capacity of each vehicle
t_{ij}	Traveling time between two vertices $(i, j) \in A$
d_i	The demand of customer *i*
γ	Fuel cost of a vehicle per time unit
δ_i	Delivery time cost per time unit of customer i
x_{ijk}	This variable is equal to 1 if arc *(i,j)* is used by vehicle *k* and 0 otherwise
s_{ik}	The start time of service for customer *i* when serviced by vehicle *k*
a_i	The delivery time of customer

The VRPDTC can then be described using the following model:

$$Min \sum_{k\epsilon K}\sum_{i\epsilon V}\sum_{j\epsilon V}\gamma t_{ij}x_{ijk}+\sum_{i\epsilon V}f(a_i) \quad (1)$$

$$\sum_{k\epsilon K}\sum_{j\epsilon V}x_{ijk}=1 \quad \forall i\in V \quad (2)$$

$$\sum_{i\epsilon V}d_i\sum_{j\epsilon V}x_{ijk}\le q \quad \forall k\epsilon K \quad (3)$$

$$\sum_{j\epsilon V}x_{0jk}=1 \quad \forall k\epsilon K \quad (4)$$

$$\sum_{i\epsilon V}x_{ihk}-\sum_{j\epsilon V}x_{hjk}=0 \quad \forall h\epsilon V,\forall k\epsilon K \quad (5)$$

$$s_{ik}+t_{ij}-M(1-x_{ijk})\le s_{jk} \quad \forall i,j\epsilon V,\forall k\epsilon K \quad (6)$$

$$a_i=\sum_{k\epsilon K}s_{\ ik} \quad \forall i\in V \quad (7)$$

$$x_{ijk}\epsilon\{0,1\} \quad \forall i,j\epsilon V,\forall k\epsilon K \quad (8)$$

$$s_{ik}\ge 0 \quad \forall i\in V,\forall k\epsilon K \quad (9)$$

$$a_i\ge 0 \quad \forall i\in V \quad (10)$$

The objective function (1) of this model is to minimize the total cost including fuel and delivery time costs. $f(a_i)$ is a due date assignment function. We assume that $f(a_i)=\delta_i a_i$ in this study. With this function, VRPTW is a general model for VRPDTC. Constraint (2) ensures that each customer exactly is met by one vehicle. Constraint (3) is guarantee capacity constraint. Next, constraints (4)-(5), characterize the flow on the path to be followed by vehicle k. Constraint (6) states that vehicle k cannot arrive at j before $s_{ik}+t_{ij}$, if it is traveling from i to j. Here M is a large scalar. Note that this constraint also forbids subtours in the solution. Constraint (7) calculates the start time of the service at node i. We can remove it and use $\sum_k s_{\ ik}$ instead of a_i in the objective function. Constraints (8)-(10) are the definition constraints of the decision variables and they refer to whether the vehicle k moves from vertex i to j or no, the start time of the service at customer i when serviced by vehicle k, and the delivery time of customer i that the distributer determines, respectively.

We should take into account the special state of the problem in order to examine the complexity of the current model. If we place the restrictions on the instances of the current model, where the $\mathrm{f}(\mathrm{a_i})=0$ for each customer *i*, it can be easily reduced to VRP. Since VRP are NP-Hard, the current model is at least NP-Hard. It can use the exact algorithm to solve the small and relatively medium size problems; however, to solve the large size problems, given the complexity of the model, it should employ heuristic and met heuristic algorithms. Hence, in the following sections, a column generation approach is employed to obtain the lower bounds of problems and to solve the problems, a quantum algorithm is proposed.

4. Column Generation Approach

Column generation (Danzig-Wolfe decomposition) is one of the most applicable algorithms in the exact optimization field for the VRP models. In this paper a column generation algorithm is designed for VRPDTC. Danzig-Wolfe decomposition of VRPDTC results in the set partitioning master problem and an Elementary Shortest Path Problem with Resource Constraints and Delivery Time Cost (ESPPRCDTC) as its sub problem.

Using Danzig-Wolfe decomposition, the master problem (M P), which consists of selecting a set of feasible paths of minimum cost, is described in equations (11)-(13), where P is the set of all feasible paths:

$$min\sum_{p\epsilon P}c_p\theta_p \quad (11)$$

Subject to

$$\sum_{p\epsilon P}\alpha_{ip}\theta_p=1,\ \forall\, v_i\in V\backslash\{v_o\} \quad (12)$$

$$\theta_p=\{0,1\},\forall\, p\epsilon P \quad (13)$$

θ_p indicates whether path $p\epsilon P$ is selected ($\theta_p=1$) or not ($\theta_p=0$); c_p represents the cost of path p and the number of times path p meets customer v_i is determined by α_{ip}. As the size of the set P grows exponentially with the number of customers, we consider $M\,P(P_1)$ as the restriction of the M P to a subset of variables $P_1\subset P$ and this problem is called the Restricted Master Problem. The column generation algorithm based on[23], is described in Algorithm 1:

Algorithm 1- Pseudo code algorithm for Column generation

Generate an initial set of columns$\boldsymbol{P_1}$
Do
  Solve $\boldsymbol{M\,P(P_1)}$
  $\boldsymbol{\Gamma}\leftarrow$ column (s) provided by the sub problem
  $\boldsymbol{P_1}\leftarrow\boldsymbol{P_1}\cup\boldsymbol{\Gamma}$
While $\boldsymbol{\Gamma}\neq\emptyset$

As mentioned before, the sub problem of this problem is ESPPRCDTC. In each iteration of column generation algorithm, a new column is obtained from solving ESPPRCDTC. In this paper, we have adapted Fillet *et al*'s approach for solving ESPPRCDTC. Fillet et al. [24]proposed an exact algorithm to solve the ESPPRC. A full description of the algorithm can be found in[24]. In the Fillet *et al*'s algorithm, each path, from the source to a node in the network, is assigned a label, which is a vector of (1) the cost of that (partial) path, (2) the resources consumed so far (vehicle load and elapsed time), (3) a resource corresponding to each customer node indicating whether the customer has already been visited or not, and (4) the total number of unreachable customers. Customers are considered *unreachable* if they have already been visited in the path or if they cannot be served by the vehicle without violating a resource constraint. We considered fuel and delivery time costs as the cost of (partial) path. Dominance rules are used to compare partial paths arriving at the same location and to discard some of

them. To obtain the optimal solution of the shortest path problem, one just needs to consider non-dominated labels, i.e., non-dominated paths[24].
A drawback of column generation approach is that in some cases, the solution is fractional and consequently it needs to embed a column generation algorithm in to a branch–and–bound algorithm and this approach is known as branch-and-price. However this study does not make use of branch-and-price to solve the problem, and column generation approach has been used for calculation of lower bounds in problems with fractional solutions.

5. Met Heuristic Algorithm

Quantum evolutionary algorithm (QEA) is a met heuristic algorithm in optimization field which is based on evolutionary computations especially genetic algorithm (GA). The last twenty years have seen the application of various properties from quantum physics to building a new kind of computers, quantum computers. In contrast to classical computers that deal with binary digits (bits), quantum computers work by manipulating quantum bits (cubits); these are the smallest units of information that can be stored in a two-state quantum computer[25]. A Q-bit $|\psi\rangle$ is not represented accurately but it is represented as a linear combination of vectors $|0\rangle$ and $|1\rangle$ .This is given in the following equations:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad (14)$$

So that:

$$|0\rangle = \begin{bmatrix}1\\0\end{bmatrix} \textit{ and } |1\rangle = \begin{bmatrix}0\\1\end{bmatrix}$$

And α and β are the numbers that satisfy (15):

$$|\alpha|^2 + |\beta|^2 = 1 \quad (15)$$

$|\alpha|^2$is the probability that Q-bit to be in zero state and $|\beta|^2$is the probability that the Q-bit is in 1 state.
So a string which has been formed from m Q-bit is shown as follow :

$$\begin{bmatrix}\alpha_1 & \alpha_2 & \cdots & \alpha_m \\ \beta_1 & \beta_2 & \cdots & \beta_m\end{bmatrix}$$

According to equation (14), vector $|\psi\rangle$ simultaneously takes into consideration 0 and 1 states. In quantum mechanism the ability of simultaneous existence in two or more states is called quantum states Superposition. In other words, the information stored in $|\psi\rangle$ is as a combination of all possible scenarios of $|0\rangle$ and $|1\rangle$. In order to access information in a Q-bit in a classic case it is necessary to create an observation, that is, a measurement. This measurement has as a probabilistic outcome a unique value contained in the superposition. Thus, when$|\psi\rangle$ is measured, it is possible to find the state $|0\rangle$ with a probability $|\alpha|^2$or the state$|1\rangle$ with a probability $|\beta|^2$[26]. This observation process is shown in Algorithm 2.

Algorithm 2- Pseudo code algorithm for observation process [25]

1. Begin
2. If $random[0,1) < |\alpha|^2$
3. Then $x = 0$
4. Else $x = 1$
5. End

In this paper, a quantum evolutionary algorithm has been proposed for the VRPDTC. In Algorithm 3, the designed algorithm has been described.

Algorithm 3- Pseudocode algorithm for Quantum Evolutionary Algorithm

1- $Initialize\ Q(t), t \leftarrow o$
2- While (not termination condition) do
3- $Generate\ p(t)\ observing\ the\ states\ of\ Q(t)$
4- $Modify\ p(t)$
5- $Evaluate\ p(t)$
6- $Store\ the\ best\ solution\ of\ P(t)\ in\ B(t)$7- $Update\ Q(t)$
8- $t \leftarrow t + 1$
9- End.

In this algorithm, first the initial population is generated, then until the termination criteria (e.g. a certain number of iterations) is not satisfy, the following operations is performed:
1 – The operation of $Q(t)$ population observation is done and saved in $P(t)$.
2 - Impossible chromosomes are modified.
3 - All the chromosomes are evaluated. (In terms of the objective function value)
4 - The best chromosome in $P(t)$ is stored in $B(t)$.
5 – $Q(t)$ is updated.
6 - $t \leftarrow t + 1$
Each of these steps is described more fully as follows.

5-1. Generating Initial Population

In this algorithm, $Q(t)$ represents the population which has PS[2] chromosome (solution). Each chromosome (i) in this population is represented with $q_i(t)$ and as the equation (16):

$$q_i(t) = \begin{bmatrix}\alpha_{i1}(t) & \alpha_{i2}(t) & \ldots & \alpha_{iN}(t) \ldots & \alpha_{iKN}(t) \\ \beta_{i1}(t) & \beta_{i2}(t) & \ldots & \beta_{iN}(t) \ldots & \beta_{iKN}(t)\end{bmatrix} \quad (16)$$

In this equation N,K represents the number of customers and vehicles respectively. In order to generate the initial population, PS arrays with sizes K×N are generated and the initial values are set as equation (17).

$$\alpha_{ij}(0) = \beta_{ij}(0) = \frac{\sqrt{2}}{2} \quad (17)$$

[2] Population size

In this paper, with the help of the initial experiments, PS set to 40. The generating initial population process is shown in algorithm 4.

Algorithm 4- Pseudo code algorithm for generating the initial population

For i=1 **to** PS **do**
  For j=1 **to** K×N **do**
  $\alpha_{ij}(0) = \beta_{ij}(0) = \frac{\sqrt{2}}{2}$
  End
End.

5-2. Observe Population $Q(t)$ and Storing in $P(t)$

In each iteration of the algorithm, the observation operation must be done for each $q_i(t)$ and $X_i(t)$ be generated for it. The structure of $X_i(t)$ is similar to $q_i(t)$ and is displayed as equation (18).

$$X_i(t) = [\underbrace{x_{i1}(t), \dots, x_{iN}(t)}_{\text{Vehicle1}}, \dots, \underbrace{x_{i(nk-n+1)}(t), \dots, x_{iKN}(t)}_{\text{Vehicle K}}] \quad (18)$$

$x_{iKN}(t)$ shows that in solution (i), whether customer N is met by vehicle K or not**.**

5-3. Modifying Chromosome

After observing the population, the solutions containing zero and one are formed. These solutions may be infeasible due to below constraints:

1. Each customer is visited by exactly one vehicle.
2. Capacity constraint.

At first, we check that whether each customer is visited by exactly one vehicle or not. If a customer has been met more than one, a vehicle randomly selected for meet it and this customer remove from customer list of other vehicles that met this customer. If a customer has been not met, a vehicle randomly selected for meet it.
Then we examine capacity constraint for each vehicle, the following operations will be done if it is not established:
Until the constraint is not established, one customer is randomly selected and if its value is one, it is changed to zero. Then this customer is added to an allowed vehicle and again this constraint is checked.

5-4. Assessing Chromosomes and Save the Best of Them

After modifying the chromosomes, the values of their corresponding objective function value (OFV) must be calculated based on Eq.1. With regard to coding procedure of solutions, chromosomes cannot show the meeting sequence of customers and they just show which customers have been assigned to each vehicle. In order to determining meeting sequence of customers in each vehicle, first a random sequence of those is selected and then a local search is used for improve it. In this local search two or three customers from one vehicle are selected and then if the exchanging of order of these customers improve OFV, this replacement will be occur. After OFV calculation of all chromosomes, the best of those is stored in B (t). Array B (t) is displayed as equation (19):

$$B(t) = [b_1(t)\ b_2(t)\ b_3(t) \dots\ b_{KN}(t)\ ] \quad (19)$$

5-5. Updating the Population

The gate operator is used to update the population of the solutions. This operator rotates each pair of α_{ij} and β_{ij} with $U(\Delta\theta_{ij})$ as the equation (20).

$$\begin{bmatrix}\alpha_{ij}(t+1)\\ \beta_{ij}(t+1)\end{bmatrix} = U(\Delta\theta_{ij})\begin{bmatrix}\alpha_{ij}(t)\\ \beta_{ij}(t)\end{bmatrix} \quad (20)$$

In which:

$$U(\Delta\theta_{ij}) = \begin{vmatrix} cos\left(\xi(\Delta\theta_{ij})\right) & -sin\left(\xi(\Delta\theta_{ij})\right) \\ sin\left(\xi(\Delta\theta_{ij})\right) & cos\left(\xi(\Delta\theta_{ij})\right)\end{vmatrix} \quad (21)$$

$$\xi(\Delta\theta_{ij}) = S(\alpha_{ij}, \beta_{ij}) \times \Delta\theta_{ij} \quad (22)$$

It these equations, $S(\alpha_{ij}, \beta_{ij})$ shows the direction of the rotation and $\Delta\theta_{ij}$ represents the rotation angle. Values of $S(\alpha_{ij}, \beta_{ij})$ and $\Delta\theta_{ij}$ are derived from Table 2. In this study, relying on the results of the pretest, the value of Δ is set to0.01π.

Tab.2. The values of parameters $S(\alpha_{ij}, \beta_{ij})$ and $\Delta\theta_{ij}$

x_i	b_i	$f(X) < f(B)$	$\Delta\theta_{ij}$	$S(\alpha_{ij}, \beta_{ij})$			
				$\alpha_{ij}\beta_{ij} > 0$	$\alpha_{ij}\beta_{ij} < 0$	$\alpha_{ij} = 0$	$\beta_{ij} = 0$
0	0	False	0	0	0	0	0
0	0	True	0	0	0	0	0
0	1	False	Δ	1	-1	0	±1
0	1	True	Δ	-1	1	±1	0
1	0	False	Δ	-1	1	±1	0
1	0	True	Δ	1	-1	0	±1
1	1	False	0	0	0	0	0
1	1	True	0	0	0	0	0

6. Computational Result

In this section, first a small-size instance of this problem is considered. Then, in order to verify the effectiveness of the proposed algorithm, a set of problems are randomly generated and results of QEA with CG[3] have been compared. The algorithms were coded using MATLAB language, executed on a computer with a 4.00 GB RAM and an Intel Core2 Duo,2.00 GHz CPU.

6-1. A Small-size Instance

In this section, a small-size instance with 7 customers is considered. The parameters of this instance are shown in Table 3. Vertex 1 is a depot node; other vertexes are customer nodes. The distance between each of two nodes is calculated based on Euclidean distance. The distance unit is converted into time unit by a coefficient $\alpha = 1$. Fuel cost of a vehicle per time unit is set to 1, the number of vehicle is set to 2 and the vehicle capacity is set to 50..

Tab.3.Parameters of random problems

Vertex	Parameters			
	x coordinate	y coordinate	demand	Delivery time cost per time
1	0	0	-	-
2	5	16	12	0.4
3	8	14	22	0.8
4	12	15	14	0.5
5	13	14	15	0.25
6	17	11	11	0.7
7	18	8	8	0.9
8	17	2	7	0.45

The objective function value of the final solution is 3.06. The routes characteristics in final solution are shown in Table 4.

Tab.4. Routes characteristics in final solution

Route No.	Route	Fuel cost	Delivery time cost
1	1-8-7-6-5-1	0.841	0.915
2	1-3-4-2-1	0.738	0.566

6-2 Performance evaluation of quantum evolutionary algorithm

In this section, in order to verify the effectiveness of the proposed algorithm, a set of problems are randomly generated. In these problems, the number of customers are defined in 13 classes and 10 instances are generated in each class. Hence, 130 instances are generated based on parameters that are shown in Table5. The demand of each customer is drawn from a lognormal distribution with parameters $\sigma = 1$ and $\mu = \ln(d') - \frac{\sigma^2}{2}$ based on [27] . This type of distribution is preferred over a uniform as it allows for a small number of customers with a high demand and a low average demand. When a demand greater than the vehicle capacity is drawn, this outcome is rejected and a new value is drawn. Therefore, d' has to be greater than $\bar{d}$ in order to get an average demand of $\bar{d}$ [27].

In random Instances, the distance between each two nodes is calculated based on Euclidean distance. The distance unit is converted into time unit by a coefficientα. The results of these problems are shown in Table 6.

Tab.5.Parameters of random problems

Parameter	Definition	Range of parameters
$\|V\| - 1$	The number of customers	n=5,10,15,16,17,18,19,20,25,30,35,40,45
$\bar{d}$	The average demand of a customer	200
q	Capacity of each vehicle	1000
x	x coordinate of customers place	[-100,100]
y	y coordinate of customers place	[-100,100]
α	Convert Euclidean distance to time multiple	(0,1]
γ	Fuel cost of a vehicle per time unit	(0,1]
δ_i	Delivery time cost per time unit of customer i	[0,1]

In Table 6, column 1 and 2 give the number of customers n and the number of instances for which an integral (optimal) solution was identified by CG. Column 3 compares objective function values of these two algorithms. Column 4 reports the average percentage gap between objective function values of these two algorithms, which is calculated as: $\left(\frac{\text{QEA}-\text{CG}}{\text{CG}}\right) 100\%$. The last column compares the average runtime (s) of these two algorithms.

As we can observe in Table 6, QEA and CG algorithm has obtained the optimum solution in the smaller instances with at most 20 customers. For the larger instances (25-45 customers) column generation has not obtained integral solutions except two instances, therefore OFV of CG is considered as lower bound. In these instances, average deviation of QEA from CG is very small and on average, the run time of the QEA is less than CG. These comparisons illustrate the efficiency of the proposed QEA algorithm.

7. Conclusion

This paper presented an integrated model for vehicle routing problem with delivery time cost. A column generation approach has been employed to obtain the lower bounds of problems. A quantum evolutionary algorithm based on quantum computations has been proposed to solve this problem. In the proposed algorithm, various operators such as observation process and gate operator were applied. For the assessment of the proposed algorithm, we compared it with the column generation in some random instances

[3] Column Generation

and it obtained the optimal and near-optimum solution in the majority of instances. However, this research considered a simple delivery time cost function. For future studies, it is suggested to work on other delivery time cost function such as general piece-wise linear function, step-wise function, applying hybrid algorithm of exact and heuristic methods (integrative or collaborative hybridization) and exact algorithm such as branch and price and branch and cut.

Tab.6. Comparison of QEA with CG results

n	Number of optimum solution (CG)	Number of solution in which OFV of		$\frac{QEA-CG}{CG}\%$	Average run time (s)	
		QEA=CG	QEA > CG		QEA	CG
5	10	10	0	0.00	1.3	3.7
10	10	10	0	0.00	5.1	7.6
15	10	10	0	0.00	8.2	14.6
16	10	10	0	0.00	9.7	20.3
17	10	10	0	0.00	14.2	22.4
18	10	10	0	0.00	14.8	18.8
19	10	10	0	0.00	15.4	24.1
20	10	10	0	0.00	18.1	23
25	2	2	8	0.54	34.1	31.1
30	0	0	10	0.92	43.7	34.4
35	0	0	10	1.83	49.2	41.9
40	0	0	10	2.13	57.6	101.5
45	0	0	10	4.06	68.1	241.5

References

[1] Deflorio, F.P., Perboli, G., and Tadei, R., "*Freight distribution performance indicators for service quality planning in large transportation networks*", Flexible services and manufacturing journal, Vol. 22, No. 1-2, 2010. pp. 36-60

[2] Tan, K.C., Chew, Y.H., and Lee, L.H., "*A Hybrid Multiobjective Evolutionary Algorithm for Solving Vehicle Routing Problem with Time Windows*", Computational Optimization and Applications, Vol. 34, No. 1, 2006. pp. 115–151.

[3] Gordon, V., Proth, J.-M., and Chu, C., "*A survey of the state-of-the-art of common due date assignment and scheduling research*", European Journal of Operational Research, Vol. 139, No. 1, 2002. pp. 1–25.

[4] Steiner, G., and Zhang, R., "*Minimizing the weighted number of tardy jobs with due date assignment and capacity-constrained deliveries*", Annals of Operations Research, Vol. 191, No. 1, 2011. pp. 171-181.

[5] Liberatore, F., Righini, G., and Salani, M., "*A column generation algorithm for the vehicle routing problem with soft time windows*", 4OR, Vol. 9, No. 1, 2011. pp. 49-82.

[6] Qureshi, A., Taniguchi, E., and Yamada, T., "*An exact solution approach for vehicle routing and scheduling problems with soft time windows*", Transportation Research Part E: Logistics and Transportation Review, Vol. 45, No. 6, 2009. pp. 960–977.

[7] Qureshi, A.G., Taniguchi, E., and Yamadaa, T., "*Exact solution for the vehicle routing problem with semi soft time windows and its application*", Procedia Social and Behavioral Sciences Vol. 2, No. 3, 2010. pp. 5931–5943.

[8] Calvete, H.I., Gale, C., Oliveros, M.a.-J., and Sa´nchez-Valverde, B.n., "*A goal programming approach to vehicle routing problems with soft time windows*", European Journal of Operational Research, Vol. 177, No. 3, 2007. pp. 1720–1733.

[9] Figliozzi, M.A., "*An iterative route construction and improvement algorithm for the vehicle routing problem with soft time windows*", Transportation Research Part C: Emerging Technologies, Vol. 18 No. 5, 2010. pp. 668–679.

[10] E. Taillard, P.B., M. Gendreau, F. Guertin, J.-Y. Potvin., " *A tabu search heuristic for the vehicle routing problem with soft time windows*", Transportation Science, Vol. 31, No. 2, 1997. pp. 170–186.

[11] Karimi, H., and Seifi, A., "*Acceleration of Lagrangian Method for the Vehicle Routing Problem with Time Windows*", International Journal of Industrial Engineering & Production Research, Vol. 23, No. 4, 2012. pp. 309-315.

[12] Bettinelli, A., Ceselli, A., and Righini, G., "*A branch-and-cut-and-price algorithm for the multi-depot heterogeneous vehicle routing problem with time windows*", Transportation Research Part C: Emerging Technologies, Vol. 19, No. 5, 2011. pp. 723-740.

[13] Tang, J., ZhendongPan, RichardY.K.Fung, and HenryLau, "*Vehicle routing problem with fuzzy time windows*", Fuzzy Sets and Systems Vol. 160, No. 5, 2009. pp. 683–695.

[14] Xu, J., Yan, F., and Li, S., "*Vehicle routing optimization with soft time windows in a fuzzy random environment*", Transportation Research Part E: Logistics and Transportation Review, Vol. 47, No. 6, 2011. pp. 1075–1091.

[15] Brito, J., Martinez, F.J., Moreno, J.A., and Verdegay, J.L., "*Fuzzy optimization for distribution of frozen food with imprecise times*", Fuzzy Optimization and Decision Making, Vol. 11, No. 3, 2012. pp. 337-349.

[16] Madsen, O.B., Tosti, K., and Vælds, J., "*A heuristic method for dispatching repair men*", Annals of Operations Research, Vol. 61, No. 1, 1995. pp. 213–226.

[17] Agatz, N., Campbell, A., and Fleischmann, M., "*Challenges and Opportunities in Attended Home Delivery*", In THE VEHICLE ROUTING PROBLEM: LATEST ADVANCES AND NEW CHALLENGES (Golden, B., Raghavan, s., Wasil, E), Vol., No., 2008.

[18] Campbell, A.M., and Savelsbergh, M., "*Decision Support for Consumer Direct Grocery Initiatives*", Transportation Science, Vol. 39, No. 3, 2005. pp. 313-327.

[19] Jabali, O., Leus, R., van Woensel, T., and de Kok, A.G., "*Self-impose time windows in vehicle routing*", Working Paper 1028, Katholieke Universiteit Leuven, Vol., No., 2010.

[20] Ibaraki, T., Imahori, S., Kubo, M., Masuda, T., Uno, T., and Yagiura, M., "*Effective local search algorithms for routing and scheduling problems with general time window constraint*", Transportation Science, Vol. 39, No. 2, 2005. pp. 206–232.

[21] Ibaraki, T., Imahori, S., Nonobe, K., Sobue, K., Uno, T., and Yagiura, M., "*An iterated local search algorithm for the vehicle routing problem with convex time penalty functions*", Discrete Applied Mathematics Vol. 156, No., 2008. pp. 2050 – 2069.

[22] Taş, D., Jabali, O., and Woensel, T.V., "*A Vehicle Routing Problem with Flexible Time Windows*", Computers & Operations Research, Vol., No., 2014.

[23] Feillet, D., "*A tutorial on column generation and branch and price for vehicle routing problem*", 4OR, Vol. 8, No. 4, 2010. pp. 407-424.

[24] Feillet , D., Dejax, P., Gendreau, M., and Gueguen, C., "*An exact algorithm for the elementary shortest path problem with resource constraints: application to some vehicle routing problems*.", Networks Vol. 44, No. 3, 2004. pp. 216–229.

[25] Zhang, G., "*Quantum-inspired evolutionary algorithms: a survey and empirical study*", Journal of Heuristics, Vol. 17, No. 3, 2011. pp. 303–351.

[26] Nicolau, A.d.S., Schirru, R., and Alvarenga de Moura Meneses, A., "*Quantum evolutionary algorithm applied to transient identification of a nuclear power plant*", Progress in Nuclear Energy, Vol. 53, No. 1, 2011. pp. 86-91.

[27] Jong, C.D., Kant, G., Vliet, A.V., "*On finding minimal route duration in the vehicle routing problem with multiple time windows*", Department of Computer Science, Utrecht University, Vol., No., 1998.