Hybrid Quantum Genetic Algorithm for Vehicle Routing Problem with Time Window

Ahmed Al-Thalej[1], Esam Taha Yassen[2] and Belal Al-Khateeb[3]

[1]*College of Computer Science and Information Technology, University of Anbar, Iraq.*

[2]*Information Technology Centre, University of Anbar, Iraq,*

[3]*College of Computer Science and Information Technology, University of Anbar, Iraq,*

ahmedmhnd@yahoo.cometalheety@gmail.com

belal@computer-college.org.

Abstract-Vehicle Routing Problem with Time Window (VRPTW)considered being the most popular and most widespread widely studied, because it includes the time windows constraint, which represents factual life situations**.** A large number of algorithms have been suggested to deal with VRPTW, which shows its importance, but none of them succeeded to be implemented efficiently on all situations of the problem. Therefore, the need to apply new or advanced algorithms to the VRPTW still exists. Recently, Quantum Genetic algorithm (QGA) has been presented as an powerful method to handle many real difficult problems and it has not been applied to solve the VRPTW. Therefore, in this work, the performance of QGA for solving the VRPTW is investigated.The obtained results show that the QGA capability of enhancing the solution quality decreases gradually during the search. That's mean the QGA stuck in local optima. In order to improve the quality of generated solution, a hybrid QGA (HQGA) is proposed. In this hybridization a Hill-Climbing algorithm (HC) is integrated with the QGA. This integration enables the QGA to explore the search space and the HC to exploit the search space. The experimental results show that the HQGA has attained competitive results in comparison to other compared approaches.

Keywords: Vehicle Routing Problem with Time Window, Genetic algorithm, Hill-Climbing, Quantum Genetic algorithm.

1. Introduction

One of the challenging optimization problems in transportation and distribution systems is known as the vehicle routing problem (VRP). Over the years, several extensions of the basic VRP have been introduced, such as the capacitated vehicle routing problem [1], vehicle routing problem with time windows [2], vehicle routing problem with pickup and delivery [3]. The VRPTW is one of the most important problems in many distribution systems problems. VRPTW is a problem to find the least distance for a range of ways to deliver goods using a combination

**Corresponding Author:*Ahmed Al-Thalej

Article History: Received: Sept 5, 2018, Revised: Oct 6, 2018, Accepted: Nov 20, 2018

of vehicles with a limited capacity and a specific service time window for each customer. Paths must be designed so that each point is visited once by one vehicle only within a certain time period , all routes from depot are starting and ending, and all customers' demands per particular route must be limited to the capacity of the vehicle. The customer service must start within the specified time windows [4,5].

Due to the importance of VRPTW, many algorithms have been proposed to address it [5].These algorithms can be classified into exact (exhaustive), heuristic and meta-heuristic algorithms .Many exact methods had handled with VRPTW such as [6,7].Exact algorithms are not suitable for large VRPTW problems regardless of the fact that it can obtain optimum solutions with guaranteed optimize for small VRPTW problems 25 to 50 instances, Due to the desired time which rises exponentially with the problem[8]. Accordingly, researchers promote to use meta-heuristic algorithms to deal with large-size VRPTW. There are two types of meta-heuristic; single-solution based meta-heuristics and population-based meta-heuristics [9]. Single-solution based meta-heuristics such as tabu search [10],hill-climbing algorithm (HC) [11], Greedy randomized adaptive search procedure (GRASP) [12], and simulated annealing (SA) [13]. Population-based meta-heuristics such as Ant Colony Optimization Algorithm [14],particle swarm optimization [8], genetic algorithm [15], scatter search [16]and harmony search algorithm [17,18].

Despite the fact that a large number of algorithms have been suggested to deal with VRPTW, none of them succeeded to be implemented efficiently on all problem instances. Therefore, the need to apply new or advanced algorithms to the VRPTW still exists. This stimulates the current study to select one of the recently proposed algorithms.Oneof the recent algorithms which could to solve several combinatorial problems is Quantum Genetic algorithm (QGA) [19], but it has not been applied to the VRPTW. Therefore, in this work, the Quantum Genetic algorithm is proposed for solving the VRPTW.

2. Quantum Genetic Algorithm

Based on the concept and principles of quantum computing the Quantum genetic algorithm is emerged. QGA combines some characteristics of quantum computation with the genetic algorithm. QGA is a probabilistic searching algorithm which exploits the power of quantum computation in order to accelerate genetic procedures [19].It is mainly based on *qubits* and states superposition of quantum mechanics. Unlike the classical representation of chromosomes (binary string for instance), here they are represented by vectors of *qubits* (quantum register). Thus, a chromosome can represent the superposition of all possible states [19]. The *qubit* is smallest unit of information stored in a two-states quantum computer. A *qubit* may have the state of 1, the 0 state or in any superposition of the two states [20].

The state of a *qubit* can be represented as following equation:

$$|\emptyset\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \quad (1)$$

Where $|\alpha|^2+|\beta|^2=1$, α and β are complex numbers that specify the probability amplitudes of the corresponding state.A *qubit* representation of *m-qubit* chromosome is shown in figure 1.

$$\begin{bmatrix}\alpha_1 \\ \beta_1\end{bmatrix}\begin{matrix}\alpha_2 \\ \beta_2\end{matrix}\begin{vmatrix}\ldots \\ \ldots\end{vmatrix}\begin{matrix}\alpha_m \\ \beta_m\end{matrix}\Big]$$

Figure (1): Q*ubit* Representation.

Where *m* represents the length of the chromosome. In this way, the *mqubit* chromosome can simultaneously represent the information of 2^m states. With the presence of the *qubit* representation, QGA has a better characteristic of population diversity than classical approaches, because it is able of representing a linear superposition of many states [21]. For example, the 2-bit binary expression (0,1) represents one state while 2-bit *qubit* expression $\begin{bmatrix}1/\sqrt{2} & 1/2 \\ 1/\sqrt{2} & \sqrt{3}/2\end{bmatrix}$ represents four states: $\frac{1}{8} < 0,0 >,\ \frac{3}{8} < 0,1 >, \frac{1}{8} < 1,0 >,\ and\ \frac{3}{8} < 1,1 >$

Where $\frac{1}{8}, \frac{3}{8}, \frac{1}{8}$ and, $\frac{3}{8}$ are probabilities.

In 1996, quantum genetic algorithm is first proposed by Narayanan and Moore to solve successfully traveling salesman problem (TSP). Based on presenting *qubit* representation and quantum logic gate operation the QGA are highly effective and robust over the TSP [22].

3. The Quantum Genetic Algorithm for VRPTW

Based on the fact that the standard QGA is successful in tackling various combinatorial optimization problems, the study hypothesize that QGA would be successful in tackling VRPTW. To solve the VRPTW which is highly constrained problem using QGA, the QGA components have to be designed carefully. In this section, the procedure of QGA for solving VRPTW is described as shown in figure (2). The procedure of QGA is described in the following steps:

Step1: Generate an initial population *Q(t)*

At the beginning, the QGA initializes a quantum population *Q(t),* which consists of ***n*** chromosomes. A quantum chromosome is defined as a string of ***m qubits***. A quantum chromosome is initialized randomly. That is, randomly generated any value in [0,1] for αi and βi according this formula:

$$|\boldsymbol{\alpha}|^2 + |\boldsymbol{\beta}|^2 = \mathbf{1} \quad (2)$$

Consequently a quantum population is defined by a set of quantum chromosomes as shown in figure (3)

Step2: Generate a binary population *P(t)*

After initialization, a binary population $P(t)$ is generated by measuring $Q(t)$. P represented as vector $P(t)=\{ x_1^t\ , x_2^t\ , \ldots, x_j^t\ \}$, where x_j^t is a binary value which is generated using process shown in figure (4).

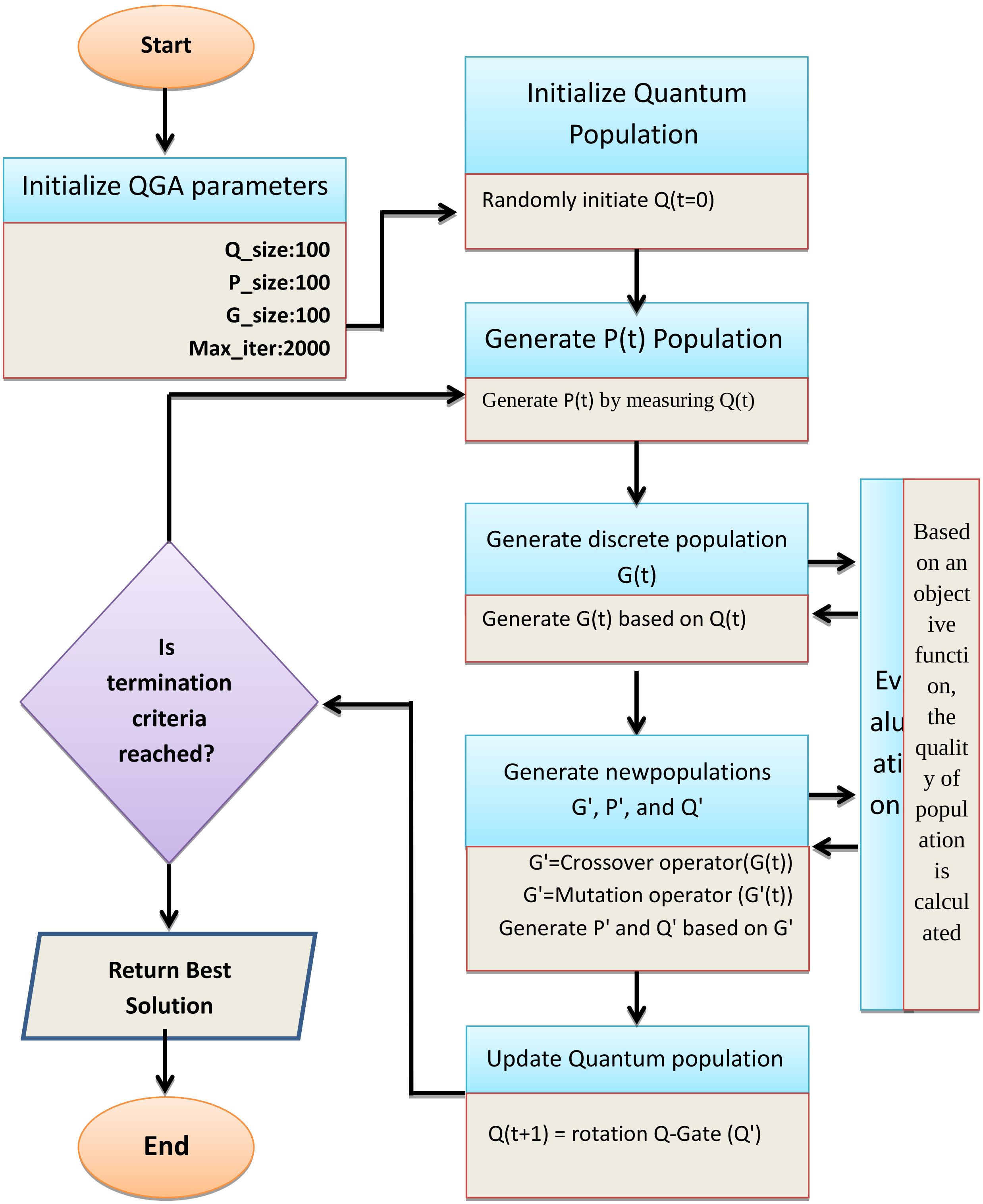

Figure (2): QGA for VRPTW.

$$q = \left[\begin{array}{c|c|c|c} \alpha_1 & \alpha_2 & \dots & \alpha_m \\ \beta_1 & \beta_2 & \dots & \beta_m \end{array}\right]$$

Figure (3): Quantum Chromosome Structure

$\in$

Figure (4): Measuring Process

Step3: Generate discrete population *G(t)*

In this step the VRPTW population is generated based on *Q(t).* The G(t) consists of *n* VRPTW solutions (chromosomes) and each solution involves *m* gene (customer ID). The random key algorithm [23] is utilized to generate VRPTW solutions as follows: For each quantum chromosome in Q(t) do the following, see figure (5):

1. Keep the index of *qubits i*n quantum chromosome based on the values of α.
2. Sort the quantum chromosome in ascending order based on the values of α.
3. Replace each *qubit* with its index in origin quantum chromosome.

qubit	**α**	**0.828**	**0.993**	**0.869**	**0.269**	**Origin**
	β	**0.559**	**0.166**	**0.493**	**0.962**	

Index	**1**	**2**	**3**	**4**	**Step 1**

qubit	**α**	**0.269**	**0.828**	**0.869**	**0.993**	**Step 2**
	β	**0.962**	**0.559**	**0.493**	**0.166**	

VRPTW solution	**4**	**1**	**3**	**2**	**Step 3**

Figure (5): VRPTW solution generation.

Step4: Evaluation

After generating the G(t) population, each VRPTW solution is associated with its fitness values according to the equation (3).

$$\boldsymbol{f(s) = min \sum_{i=0}^{n} \sum_{j=0}^{n} \sum_{k=0}^{v} t_{ij} \times x_{ij}^{k}} \qquad (3)$$

Step5: Generate new populations, as follows:

1. Applying crossover operator on the G(t) to generate a new population G'(t). In this process, the recombination is performed upon the selected parents as follows, see figure (6):

   a) Each two parents (p1 and p2) will pass through the crossover operator to generate new two children (c1 and c2).
   b) A certain position (*pos*) is randomly determined over the length of p1 and p2.
   c) Regarding to the first children:
      i. Create an empty solution c1.
      ii. From the parent *p1*, the left segment side of the *pos* are inserted into the c1.

iii. Others missing customers are selected from p2 and inserted into the c1 without violate the hard VRPTW constraints.

d) Regarding to the second children:

i. Create an empty solution c2.

ii. From the parent *p*2, the left segment side of the *pos* are inserted into the c2.

iii. Others missing customers are selected from p1 and inserted into the c2 without violate the hard VRPTW constraints.

▼

pos	1	2	3	4	5	6	7
P1	**6**	**5**	**7**	**2**	**3**	**1**	**4**
P2	**5**	**6**	**3**	**2**	**1**	**7**	**4**
Crossover							
C1	**6**	**5**	**7**	**3**	**2**	**1**	**4**
C2	**5**	**6**	**3**	**7**	**2**	**1**	**4**

Figure (6): Example of Crossover.

2. Mutation operator: In order to apply small modification on the generated children c1 and c2, the Mutation operator is utilized as follows: randomly select two deferent customers in each child and exchange their positions as shown in figure (7). is adopted in this step on the G'(t).

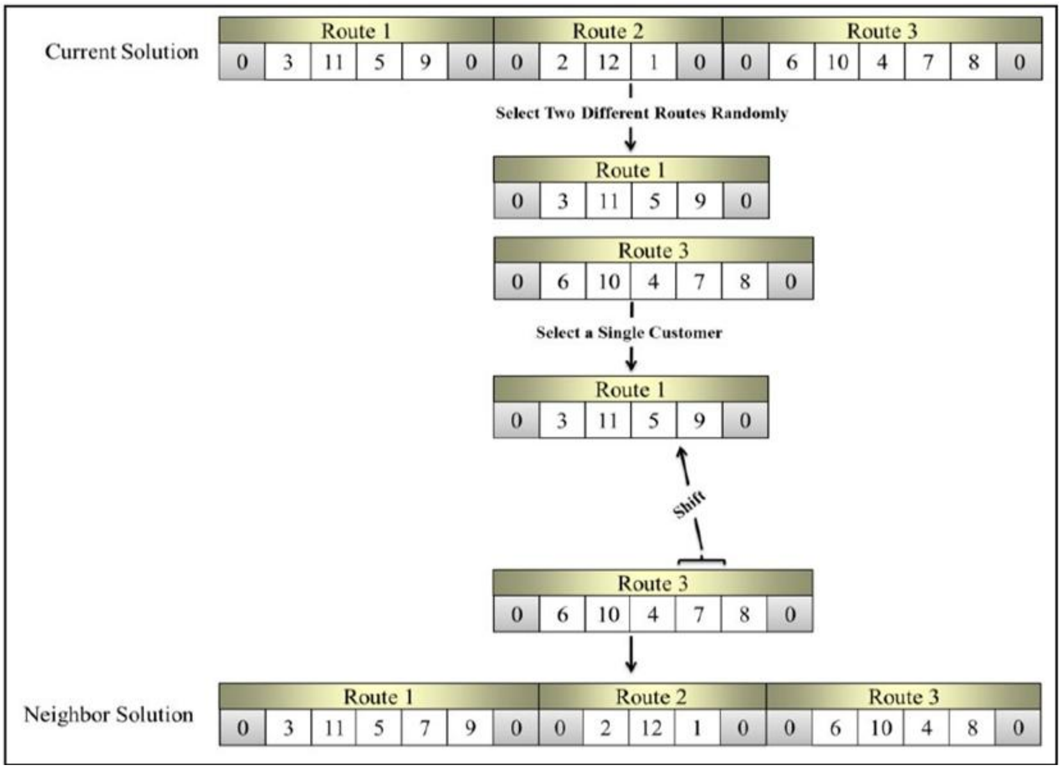

Figure (7): Example of Mutation.

3. Based on G'(t), generate Q'(t) and P'(t).

Step6: Update quantum population Q(t+1)

As in standard QGA, the quantum population Q(t+1) is updated with a quantum gates rotation of *qubits* constituting individuals based on Q'(t). The rotation of individual's amplitudes is performed by quantum gates. This rotation strategy is given by equation (4) :

$$\begin{pmatrix} \alpha_j^{t+1} \\ \beta_j^{t+1} \end{pmatrix} = \begin{pmatrix} cos(\delta\theta_j) & -sin(\delta\theta_j) \\ sin(\delta\theta_j) & cos(\delta\theta_j) \end{pmatrix} \begin{pmatrix} \alpha_j^{t} \\ \beta_j^{t} \end{pmatrix} \qquad (4)$$

Being α_j^t , β_i^t the amplitudes of the j^{th} *qubit* before the updating and α_j^{t+1} , β_i^{t+1} the amplitudes of the j^{th} *qubit* after the updating. In general, the rotation angle is obtained according to the following expression:

$$\delta\theta = sg(\alpha_j , \beta_j)\Delta\theta_j (5)$$

Where $sg(\alpha_j, \beta_j)$ represent the direction and $\Delta\theta$j represent rotation value. It worth noting an updating process is represented by the rotation. Consequently, too high or too low values should be avoided. The values of these parameters are summarized in Look up Table (Table 1),comparing the fitness of the best individual $f(best_j)$ with the fitness of the current chromosome $f(x_j)$.if $f(x_j)$ is of a less quality than $f(best_j)$, then adjust the corresponding qubits of q_j^t, making the probability amplitude (α_j,β_j) evolves toward the direction that is propitious to the emergence of *best*$_j$. Otherwise, making the probability amplitude ($\alpha_{j,j}$) evolves toward the direction that is propitious to the emergence of x_j.

Table (1):Lookup Table of the Rotation Angle ($\theta$is the angle step).

x_j	b_j	$f(x_j) \geq f(b_j)$	$\Delta\theta_j$	$sg(\alpha_j \beta_j)$			
				$\alpha_j\beta_j > 0$	$\alpha_j\beta_j < 0$	$\alpha_j = 0$	$\beta_j = 0$
0	0	False	0	-	-	-	-
0	0	True	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	-	-	-	-
1	1	True	0	-	-	-	-

To determine the rotation value by the following equation (6):

$$\Delta\theta_j = 0.005\pi + (0.05\pi - 0.005\pi)\frac{|f(x_j) - f(b_j)|}{\max(f(x_j), f(b_j))} \quad (6)$$

Finally, the new quantum population Q(t+1) is generated.

Step7: Termination process

This step checks the termination criterion of QGA. If the termination criterion is satisfied, QGA will stop and return the best solution found. Otherwise, go to the step 2.

4. Hybrid QGA

The results of standard QGA indicated that QGA suffers from slow convergence which prevents it from obtaining better results. This is because QGA is good in exploration but bad in exploitation. So, in order to compensating the weakness of QGA exploitation, the QGA is combined with one single-based meta-heuristic that is characterized with its effective ability to exploit the search space. As a result of this combination (hybridization) process, the HQGA emerged. In this work, the Hill-Climbing (HC) [9] is combined with QGA to improve its exploitation ability. The HC can be described as follows:

Given an initial solution *S*, HC generates a neighbor solution *S`*. *S* is replaced with *S'* if the quality of *S'* is better than *S;* otherwise, *S'* is rejected, and HC begins a new iteration. The search process will be repeated as long as the stopping criterion is not satisfied, see algorithm (1).

Algorithm 1: Hill Climbing Algorithm

```
s = s0 ; /∗Generate an initial solution s0∗/
While not Termination Criterion Do
        Generate (N(s)) ; /∗Generation of candidate neighbors∗/
        If there is no better neighbor Then Stop;
        s = s′; /∗Select a better neighbor s′∈ N(s)∗/
End while
Output Final solution found (local optima).
```

In this work, the neighborhood solution is generated using the Two-opt star neighborhood operator [24,25,26,27,28]. The Two -opt star operator randomly selects two routes from the current solution and swap the customers located at the end sections of selected routes, see example in figure (8).

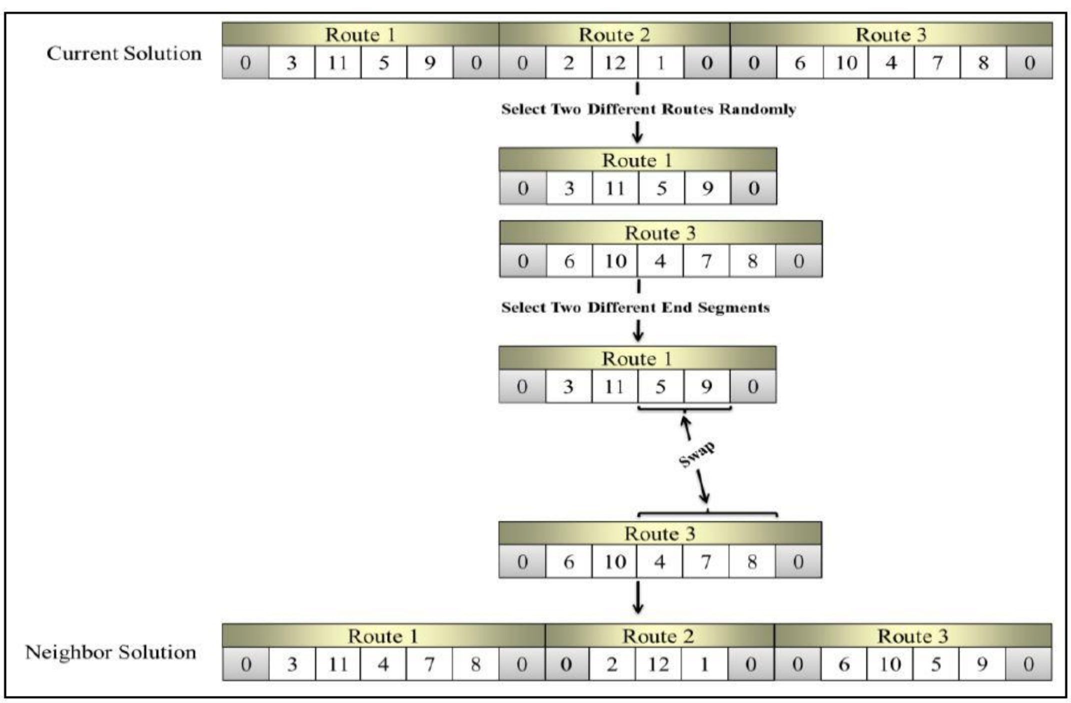

Figure (8): Two-opt Star Strategy.

In this hybridization, the HC is applied before the quantum population updating step. That is, the best individual in the G(t) population is used as an initial solution for the HC as shown in figure (9).

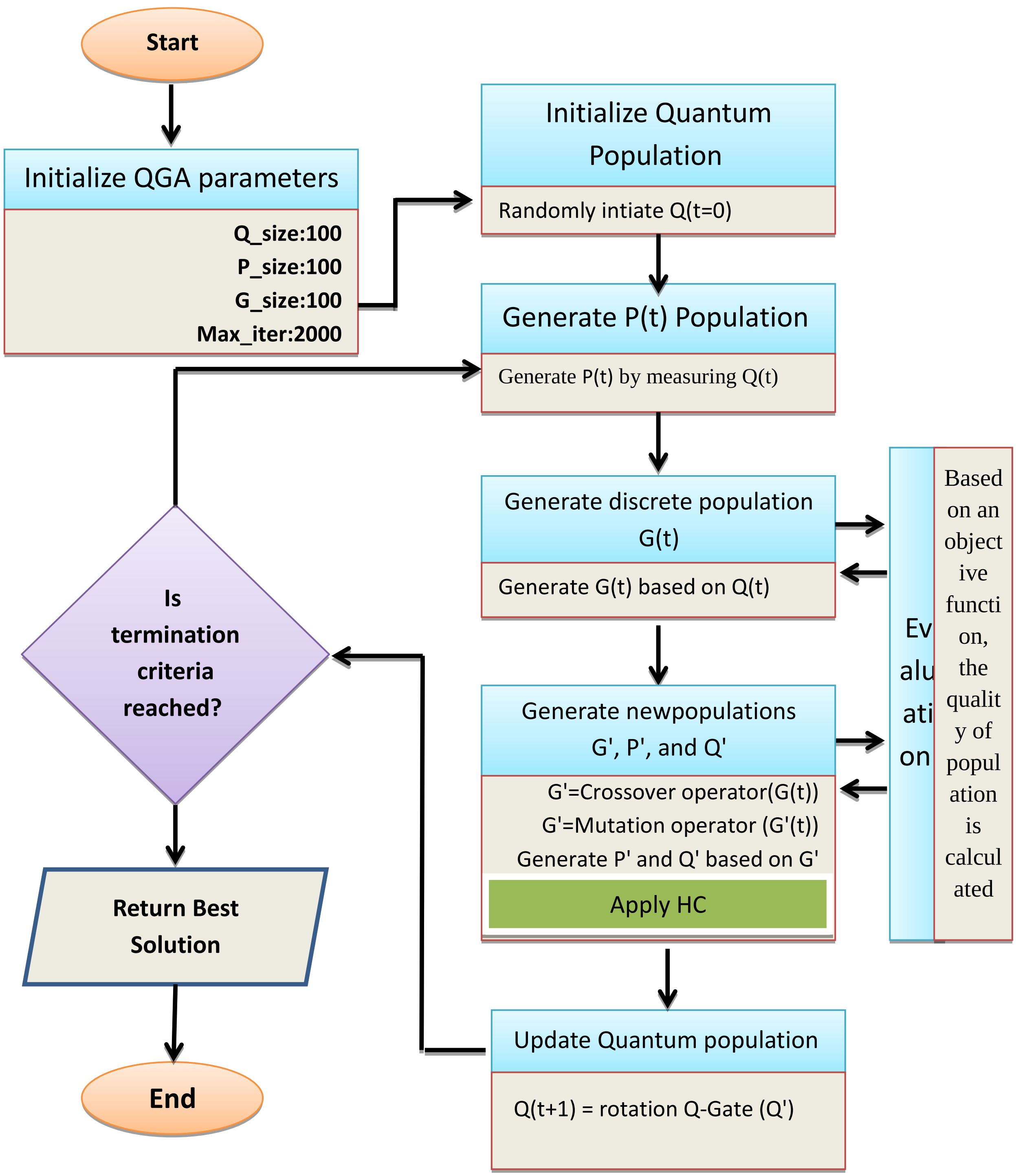

Figure (9): HQGA for VRPTW.

5. Experimental Design

In order to assess the effectiveness of the proposed algorithms, QGA and HQGA, this work utilizes VRPTW benchmark instances [6]. In this section, we first present the characteristics of VRPTW benchmark. Then, we discuss the adopted parameter values of the proposed algorithms.

5.1 The Solomon's VRPTW Benchmark

Solomon's VRPTW benchmark [2] is used in this work because it is able to take into consideration different constraints related to vehicle capacity, customers' demands and time windows density. Hence, this benchmark is very realistic. Moreover, this benchmark has been adopted by most researchers to verify the performance of their heuristics [4,5]. Consequently, the effectiveness of the proposed QGA improvements is evaluated using the Solomon's VRPTW benchmark which is available at (http://w.cba.neu.edu/~msolomon/problems.htm). This benchmark involves 56 instances. Each of these has 100 customers (the number of customers represents the problem size). Each customer is associated with its details: customer identity, customer location (x, y), customer demand, start of time window, end of time window and service duration.

According to the distribution of customers, these instances are categorized into three main classes called *R*, *C* and *RC*. Class *C* contains clustered customers in which the traveling distances and times between customers are short. Class *R* contains randomly distributed customers in which the traveling distances and times between customers are comparatively longer than class *C*. Class *RC* contains mixed distribution customers (clustered and randomized). The Euclidean distance (d_{ij}) between any two customers $C_i$and $C_j$is considered as the traveling time. Furthermore, every class is divided into two subclasses called 1, i.e., (R1, C1 and RC1) and 2, i.e., (R2, C2 and RC2) depending on the customers' time windows. R1, C1 and RC1 have customers with short time windows; while R2, C2 and RC2 have customers with long time windows [4], Table (2) summarizes the characteristics of these types.

Table (2): The Characteristics of Solomon's VRPTW Datasets.

Dataset	No. of Instances	No. of Customers	No. of Vehicles	Capacity of Vehicle	Distribution of Customers	Width of Time Window
R1	12	100	25	200	Random	Small Time Windows
R2	11	100	25	1000	Random	Large Time Windows
C1	9	100	25	200	Cluster	Small Time Windows
C2	8	100	25	700	Cluster	Large Time Windows
RC1	8	100	25	200	Random /Cluster	Small Time Windows
RC2	8	100	25	1000	Random /Cluster	Large Time Windows

5.2Parameters Settings

Since the GA, QGA and HQGA have many parameters that need to be determined before the implementation. The performance of the proposed algorithms is affected by these parameters; a series of experiments was conducted to see the effects of those parameters on the algorithm's performance [24].

The proposed algorithms have four parameters: GA, QGA and HQGA population size (***n***), solution size (***m***) and maximum number of iterations (***MaxItr***). HC has only one parameter, the maximum number of iteration (***HC_maxItr***). According to the results of the preliminary testing conducted on different values [25], these parameters are fixed. Table (3) summarizes the parameter settings [26] of all the algorithms used in this study [27].

Table (3): The Parameter Settings of All Algorithms

Parameter	**Value**
n	**100**
m	**100**
MaxItr	**2000**
HC_maxItr	**150**

6. Experimental Results

To investigate the performance of the proposed QGA for VRPWT, two sets of Experiments were conducted. The first one is to evaluate the efficiency of standard QGA in solving VRPTW. To do so, we compared its results with these of GA. In the second experimental test, we analyze the effect of HC on the performance of QGA. So, the results of HQGA are compared with the results of HC and QGA [28].

To gain sufficient experimental results for these experimental tests, we executed the proposed algorithms for 11 independent runs. Then, the best, average, standard deviation are reported. All these algorithms are tested on twelve Solomon VRPTW instances that were introduced by [2]. These instances are: (R101, R102, R201, R202, C101, C109, C201, C206, RC101, RC102, RC201, and RC202). The mathematical and statistical evaluation for the obtained results of the proposed algorithms is described [29].

6.1 The Standard QGA Results

This experiment is designed to investigate the effectiveness of the standard QGA in solving the VRPTW. In order to investigate the performance the QGA, its results [30] are compared with these of standard GA. Table (4) presents the Best, Av and Std for QGA and GA over 11 runs. From Table (4), we can observe the following: in terms of the Best quality solution [31], QGA obtained a better quality solution for six out of 12 tested instances compared to GA. In term of Avr, QGA is better than GA on seven instances out of 12. According to the Std results reported in Table (3), QGA is better than GA on 10 tested instances out of 12 instances [32].

Table (4): Results Comparison between GA and QGA.

Instance	GA			QGA		
	Best	Avr	Std	Best	Avr	Std
R101	**1891.87**	**2004.89**	66.81681	1944.457	2007.679	**33.87196**
R102	1827.64	1911.17	71.12597	**1812.658**	**1858.869**	**27.84494**
R201	**1657.30**	1733.24	53.02217	1674.251	**1730.709**	**30.59686**
R202	**1495.28**	1672.26	144.4957	1574.553	**1633.393**	**34.66816**
C101	1544.85	1672.85	110.7533	**1504.754**	**1591.165**	**65.50775**
C109	**1524.32**	**1648.82**	85.34133	1573.030	1765.409	115.95354
C201	1336.94	**1434.88**	58.74583	**1282.458**	1451.143	82.62737
C206	**1234.60**	**1442.75**	104.0266	1305.478	1519.675	**94.63431**
RC101	2089.71	2208.77	107.6252	**2059.855**	**2169.315**	**51.09241**
RC102	1962.61	2055.55	65.43556	**1850.957**	**2002.997**	**58.29527**
RC201	1885.33	2044.68	78.37532	**1851.003**	**2008.146**	**58.29527**
RC202	**1679.71**	**1908.13**	128.6558	1872.72	1991.255	**57.42184**

Figure (10) shows the behavior of QGA during the search. This figure illustrates that at the beginning periods of the search, the QGA succeeds in tackling the VRPTW via enhancing the solution quality. However, after 47% of the search process, at iteration 937 onward, the QGA capability of enhancing the solution quality decreases gradually. That's mean that the QGA stuck in local optima [33]. This is because the suggested QGA is efficient in exploration but not in exploitation [34].

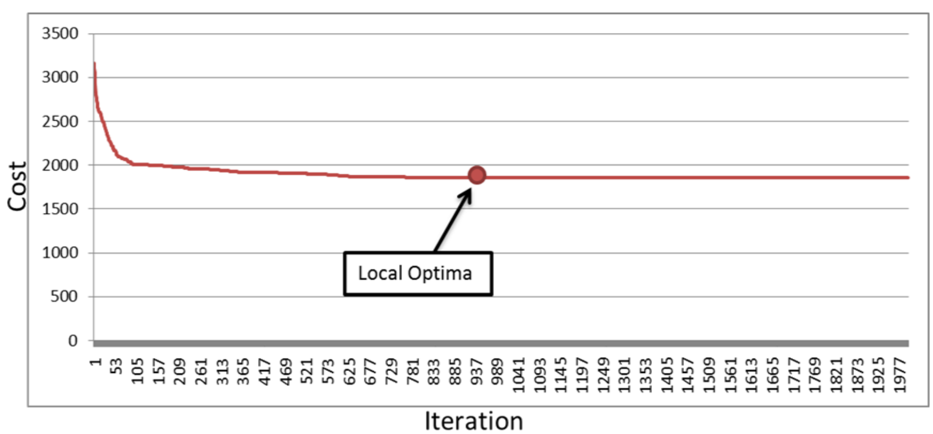

Figure (10)**:**TheBehavior of QGA during the Search (R201 instance).

6.2 The HQGA Results

In order to improve the QGA exploitation process and the quality of generated solution, a hybrid QGA (HQGA) will be proposed. In this hybridization, a single-based meta-heuristic (HC) was integrated with the QGA. This integration made use of QGA to explore the search space and the HC to exploit the search space.The results of QGAand HC are compared with these of the HQGA. This comparison are reported in tables (5), (6) and (7) in terms of Best, Average, and Std, respectively [35].

With regard to the Best, table (5) shows that HQGA obtained the better results in eight out of 12 instances compared to HC and QGA [36].

Table (5): Comparison HQGA among Different Heuristics Regarding to the Best.

Instance	HC	QGA	HQGA
R101	2395.051	1944.457	**1943.044**
R102	2156.006	1812.658	**1658.37**
R201	2578.919	1674.251	**1587.468**
R202	2538.584	1574.553	**1539.373**
C101	2397.752	**1504.754**	1527.395
C109	2233.954	1573.030	**1498.653**
C201	2362.825	**1282.458**	1289.09
C206	2140.252	1305.478	**1257.492**
RC101	2628.809	2059.855	**2034.47**
RC102	2679.113	**1850.957**	1891.193
RC201	2741.994	**1851.003**	1852.478
RC202	2872.913	1872.72	**1838.094**

According to the Avr, table (6) shows that HQGA obtained the better results in all instances compared to HC and QGA.

Table (6): Comparison HQGA among Different Heuristics Regarding to the Avr.

Instance	HC	QGA	HQGA
R101	2474.569	2007.679	**1986.49**
R102	2388.04	1858.869	**1812.942**
R201	2773.307	1730.709	**1693.767**
R202	2469.116	1633.393	**1589.562**
C101	2469.116	1591.165	**1613.024**
C109	2325.352	1765.409	**1556.302**
C201	2581.287	1451.143	**1436.065**
C206	2465.58	1519.675	**1433.143**
RC101	2836.463	2169.315	**2141.26**

RC102	2819.433	2002.997	**1982.964**
RC201	3102.524	2008.146	**1983.094**
RC202	3077.579	1991.255	**1942.449**

In terms of the Std, table (7) shows that HQGA obtained the better results in 8 out of 12 instances compared to HC and QGA, respectively.

Table (7):

Instance	HC	QGA	HQGA
R101	65.91005	33.87196	**25.69852**
R102	122.2763	**27.84494**	62.65536
R201	136.8781	**30.59686**	54.02967
R202	185.3814	34.66816	**32.74791**
C101	51.98635	65.50775	**44.70781**
C109	57.48637	115.95354	**32.30996**
C201	122.0615	82.62737	**68.41679**
C206	170.9263	**94.63431**	98.41659
RC101	94.04313	51.09241	**50.97785**
RC102	134.4966	58.29527	**48.12922**
RC201	144.5415	**58.29527**	68.64602
RC202	140.3834	57.42184	**47.21259**

Comparison HQGA among Different Heuristics Regarding to the Std.

Finally, the obtained results have shown that the hybrid QGA outperformed the standard QGA. This is due to the fact that the hybrid QGA integrates the abilities of HC exploitation and the standard QGA exploration.

7. Conclusion

The present study contributes to the field of transportation and distribution systems via addressing VRPTW aiming to reduce the distance required to serve customers. This was achieved by adopting the QGA for solving VRPTW via utilizing its strong abilities to improve the quality of obtained solutions. Regarding to the experiment which conducted in this study, at the early periods of the search process, the ability of QGA gradually decreased. This means that QGA suffers from slow convergence which prevents it from obtaining better results. So, in order to consolidate the performance of QGA, a hybrid QGA (HQGA) is proposed. In this hybridization, QGA combined with one single-based meta-heuristic (HC) that is characterized with its effective ability to exploit the search space. The obtained results had shown that the HQGA outperformed the standard QGA. This is due to the fact that the HQGA integrates the abilities of HC exploitation and the standard QGA exploration.

References

[1] Dantzig, G. B. & J. H. Ramser 1959. The truck dispatching problem. *Management science*: 80-91.

[2] Solomon, M. M. 1987. Algorithms for the vehicle routing and scheduling problems with time window constraints. Operations research: 254-265.

[3] Dethloff, J. 2001. Vehicle routing and reverse logistics: the vehicle routing problem with simultaneous delivery and pick-up. *OR spectrum* 23(1): 79-96.

[4] P. Mohamed Shakeel; Tarek E. El. Tobely; Haytham Al-Feel; Gunasekaran Manogaran; S. Baskar., "Neural Network Based Brain Tumor Detection Using Wireless Infrared Imaging Sensor", IEEE Access, 2019, vol 7, issue 1, Page(s): 1

[5] Bräysy, O. & M. Gendreau 2005b. Vehicle routing problem with time windows, part II: Metaheuristics. *Transportation science* 39(1): 119-139.

[6] Desrochers, M., J. Desrosiers & M. Solomon 1992. A new optimization algorithm for the vehicle routing problem with time windows. *Operations research* 40(2): 342-354.

[7] Mohammed, M.A., Ghani, M.K.A., Hamed, R.I., Mostafa, S.A., Ahmad, M.S. and Ibrahim, D.A., 2017. Solving vehicle routing problem by using improved genetic algorithm for optimal solution. Journal of Computational Science, 21, pp.255-262.

[8] Gong, Y. J., J. Zhang, O. Liu, R. Z. Huang, H. S. H. Chung & Y. H. Shi 2012b. Optimizing the Vehicle Routing Problem With Time Windows: A Discrete Particle Swarm Optimization Approach. *IEEE Transactions on Systems Man and Cybernetics-Part C-Applications Reviews* 42(2): 254.

[9] Talbi, E. G. 2009. Metaheuristics from design to implementation Ed.: Wiley Online Library.

[10] Garcia, B.-L., J.-Y. Potvin & J.-M. Rousseau 1994. A parallel implementation of the tabu search heuristic for vehicle routing problems with time window constraints. *Computers & Operations Research* 21(9): 1025-1033.

[11] Shakeel PM. Neural Networks Based Prediction Of Wind Energy Using Pitch Angle Control. International Journal of Innovations in Scientific and Engineering Research (IJISER). 2014;1(1):33-7.

[12] Feo, T. A. & M. G. Resende 1989. A probabilistic heuristic for a computationally difficult set covering problem. *Operations Research Letters* 8(2): 67-71.

[13] Kirkpatrick, S. & M. Vecchi 1983. Optimization by simmulated annealing. *science* 220(4598): 671-680.

[14] Dorigo, M. 1992. Optimization, learning and natural algorithms. *Ph. D. Thesis, Politecnico di Milano, Italy*.

[15] Bräysy, O. & M. Gendreau 2001. Genetic algorithms for the vehicle routing problem with time windows. *Arpakannus,(1)*: 33-38.

[16] Russell, R. A. & W. C. Chiang 2006. Scatter search for the vehicle routing problem with time windows. *European Journal of Operational Research* 169(2): 606-622.

[17] Esam Taha Yassen, Masri Ayob, Mohd Zakree Ahmad Nazri and Zulkifli Ahmad, 2013. Harmony Search Algorithm for Vehicle Routing Problem with Time Windows. Journal of Applied Sciences, 13: 633-638.

[18] E.T. Yassen, M. Ayob, M.Z.A. Nazri, N.R. Sabar, A hybrid meta-heuristic algorithm for vehicle routing problem with time windows. International Journal on Artificial Intelligence Tools, 24 (6) (2015), p. 1550021.

[19] Lahoz-Beltra, R. Quantum genetic algorithms for computer scientists. Computers 2016, 5, 24.

[20] K.Bommaraju, A. Manikandan, S.Ramalingam, "Aided System for Visually Impaired People in Bus Transport using Intel Galileo Gen-2" International Journal of Vehicle Structures & Systems, Online ISSN: 0975-3540, Volume 9, Issue 2,pp 110-112,2017.

[21] Kuk hyun Han and Jong-Hwan Kim. Genetic quantum algorithm and its application to combinatorial optimization problem. In in Proc. 2000 Congress on Evolutionary Computation. Piscataway, NJ: IEEE, pages 1354–1360. Press, 2000.

[22] A. Narayanan, M. Moore, Quantum inspired genetic algorithm, in: IEEE International Conference on Evolutionary Computation, Piscataway, January, 1996, pp. 61-66.

[23] James C. Bean, (1994) Genetic Algorithms and Random Keys for Sequencing and Optimization.

[24] Mostafa, S.A., Ahmad, M.S., Mustapha, A. and Mohammed, M.A., 2017. Formulating layered adjustable autonomy for unmanned aerial vehicles. International Journal of Intelligent Computing and Cybernetics, 10(4), pp.430-450.ORSA Journal on Computing 6(2):154-160.

[25] Ghani, M.K.A., Mohammed, M.A., Ibrahim, M.S., Mostafa, S.A. And Ibrahim, D.A., 2017. Implementing An Efficient Expert System For Services Center Management By Fuzzy Logic Controller. Journal of Theoretical & Applied Information Technology, 95(13).

[26] Mostafa, S.A., Mustapha, A., Hazeem, A.A., Khaleefah, S.H. and Mohammed, M.A., 2018. An Agent-Based Inference Engine for Efficient and Reliable Automated Car Failure Diagnosis Assistance. IEEE Access, 6, pp.8322-8331.

[27] Mohammed, M.A., Ghani, M.K.A., Hamed, R.I., Mostafa, S.A., Ibrahim, D.A., Jameel, H.K. and Alallah, A.H., 2017. Solving vehicle routing problem by using improved K-nearest neighbor algorithm for best solution. Journal of Computational Science, 21, pp.232-240.

[28] Potvin, J.-Y. & J.-M. Rousseau 1995. An exchange heuristic for routeing problems with time windows. *Journal of the Operational Research Society* 46(12): 1433-1446.

[29] Mohammed, M.A., Ghani, M.K.A., Mostafa, S.A., Ibrahim, D.A., 2017. Using Scatter Search Algorithm in Implementing Examination Timetabling Problem. Journal of Engineering and Applied Sciences, 12: 4792-4800.

[30] A. Meri *et al.*, “Modelling the Utilization of Cloud Health Information Systems in the Iraqi Public Healthcare Sector,” *Telemat. Informatics*, 2018.

[31] P. M. Shakeel, A. Tolba, Z. Al-Makhadmeh, and M. M. Jaber, “Automatic detection of lung cancer from biomedical data set using discrete AdaBoost optimized ensemble learning generalized neural networks,” *Neural Comput. Appl.*, pp. 1–14, 2019.

[32] P. M. Shakeel, S. Baskar, V. R. S. Dhulipala, and M. M. Jaber, “Cloud based framework for diagnosis of diabetes mellitus using K-means clustering,” *Heal. Inf. Sci. Syst.*, vol. 6, no. 1, p. 16, 2018.

[33] P. M. Shakeel, S. Baskar, V. R. S. Dhulipala, S. Mishra, and M. M. Jaber, “Maintaining security and privacy in health care system using learning based Deep-Q-Networks,” *J. Med. Syst.*, vol. 42, no. 10, p. 186, 2018.

[34] M. Jarrar, M. S. Minai, M. Al-Bsheish, A. Meri, and M. Jaber, “Hospital nurse shift length, patient-centered care, and the perceived quality and patient safety,” *International Journal of Health Planning and Management*, 2018.

[35] Bräysy, O. & M. Gendreau 2005a. Vehicle routing problem with time windows, Part I: Route construction and local search algorithms. *Transportation science* 39(1): 104-118.

[36] Anderson, D., E. Anderson, N. Lesh, J. Marks, B. Mirtich, D. Ratajczak & K. Ryall 2000. Human-guided simple search. *AAAI/IAAI*. 209-216.