Republic of Iraq
Ministry of Higher Education and Scientific Research
University of Anbar
College of Computer Science and Information Technology
Department of Computer Science

A Hybrid Approach to Solve Vehicle Routing Problem with Time Window Based on Quantum and Evolutionary Computing

A Thesis
Submitted to the Department of Computer Science, University of Anbar College of Computer Science and Information Technology as Partial Fulfillment of the Requirement for the Degree of Master of Computer Science

By
Ahmed Muhaned Fyaidh

Supervised By
Prof. Dr. Belal Al-Khateeb
Assist. Prof. Dr. Essam Taha Yassen

2019 A.D. *1440 A.H.*

بسم الله الرحمن الرحيم

﴿يَرْفَعِ اللَّهُ الَّذِينَ آمَنُوا
مِنكُمْ وَالَّذِينَ أُوتُوا الْعِلْمَ
دَرَجَاتٍ﴾

صدق الله العظيم

المجادلة (١١)

Supervisor’s Certification

*I certify that I read this thesis entitled “***A Hybrid Approach to Solve Vehicle Routing Problem with Time Window Based on Quantum and Evolutionary Computing***” that was done under my supervision at the Department of Computer Science, College of Computer Science and Information Technology of the University of Anbar, by student “***Ahmed Muhaned Fyaidh***” and that in my opinion it meets the standard of a thesis for the degree of Master of Science in Computer Science.*

Signature :

Name : *Prof. Dr. Belal Ismail Khalil Al-Khateeb*

Date : / / 2019

Signature :

Name : *Assist. Prof. Dr. Essam Taha Yassen*

Date : / / 2019

Certification of the Examination Committee

We the examination committee certifies that we have read this thesis entitled ***"A Hybrid Approach to Solve Vehicle Routing Problem with Time Window Based on Quantum and Evolutionary Computing"*** *and have examined the student* ***"Ahmed Muhaned Fyaidh"*** *in its contents. In our opinion it meets the standards of a thesis for the degree of Master of Science in Computer Science.*

Signature:
Name: Assist. Prof. Dr.Salah Sleibi Al-Rawi ***(Chairman)***
Date: / / 2019

Signature:
Name: Assist. Prof. Dr.Hasanen S. Abdulla ***(member)***
Date: / / 2019

Signature:
Name: Assist. Prof. Dr.Wesam M. Jasim Al-Rawi ***(member)***
Date: / / 2019

Signature:
Name: Prof. Dr. Belal Ismail Khalil Al-Khateeb ***(Supervisor)***
Date: / / 2019

Signature:
Name: Assist. Prof. Dr. Essam Taha Yassen ***(Supervisor)***
Date: / / 2019

Approved by the Dean of the College of Computer Science and Information Technology, University of Al Anbar.
Signature:
Name: Prof. Dr. Belal Ismail Khalil Al-Khateeb
Title: ***Dean of the College*** *Date: / / 2019*

Dedication

I would like to dedicate this work

To him who remove thorns from my way to pave the way of science. To the light, who illuminates the path of my success. To him who I carry his name proudly. My dear father.

To my angel in life. To the meaning of love, compassion and dedication. To the smile of my life and the mystery of my existence. To the one who her prayer was the secret of my success and her affection was balm to my surgeons. My beloved mother.

To those who supported me and waived their rights to satisfy me and make me live in blissful. My brothers.

To the one who shared my worry and carried with me the difficulties. To the one who supported me in all time. To the one who believed in my abilities. To my wife.

To my best friends.

Acknowledgements

I would like to thank and praise Allah, Lord of the worlds, who helped me to complete this thesis.

I would like to express my appreciation and gratitude to my supervisors "Dr.Belal Al-Khateeb" and "Dr.Essam Taha Yassen" for guidance, encouragement, advice and support during this work.

Also, I would like to thank the staff of Computer Science and Information Technology collage of Anbar University for their help.

I would like to extend my thanks and deep appreciation to my family who has sacrificed so much in order to achieve my goal.

Finally, I would like to thank everyone who supported me.

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**.** VRPTW is a problem to find the least distance for a range of ways to deliver goods using a combination 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 are starting from one depot and ending with the same depot, and all customers' demands per particular route must not exceed the vehicle's capacity**.** The customer service must start within the specified time windows.

Due to the importance of VRPTW, many algorithms have been proposed to address it, these algorithms can be classified into exact (exhaustive), heuristic and meta-heuristic algorithms. But, in one hand, none of these algorithms have succeeded in working efficiently in all instances of the problem. Consequently, more efficient algorithm that can significantly work well to improve the quality of obtained solution is highly required. In other hand, Quantum Genetic algorithm (QGA) has been presented as a powerful method to handle many real difficult problems and it has not been applied to the VRPTW.

This thesis aims to investigate the performance of Quantum Genetic Algorithm (QGA) and enhance its ability in tackling the VRPTW via conducting several modifications. These modifications are concerned with QGA designing and hybridization. QGA is a product of the combination of quantum computation and genetic algorithms. The obtained results show that the behavior of QGA during the search that at the early periods of the search process, succeeds in tackling the VRPTW via enhancing the solution quality. However, the QGA capability of enhancing the solution quality decreases gradually. That’s mean the QGA stuck in local optima. This problem often occurred because of the QGA is effective in exploration but not in exploitation. In order to improve the QGA exploitation process and the quality of generated solution, a hybrid QGA (HQGA) is proposed. In this hybridization a single-based meta-heuristic Hill-Climbing (HC)

was 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, this is due to the fact that the hybrid QGA integrates the abilities of HC exploitation and the standard QGA exploration.

Table of Contents	Page No.

Table of Figures

Table of Tables

Table of Algorithm

Table of Abbreviations

Abbreviation	***Meaning***
ACO	Ant Colony Optimization
AI	Artificial Intelligence
Avr	Average
CA	Cellular Automata
CGA	Conventional Genetic Algorithm
COPs	Combinatorial Optimization Problems
CVRP	Capacitated Vehicle Routing Problem
EU	European Union
GA	Genetic Algorithm

GRASP	Greedy Randomized Adaptive Search Procedure
HC	Hill-Climbing
HS	Harmony Search
HQGA	Hybrid Quantum Genetic Algorithm
IQGA	Improved QGA
MACS	Multiple Ant Colony System
BP	Binary Population
PQGA	Parallel Quantum Genetic Algorithm
QP	Quantum Population
QGA	Quantum Genetic Algorithm
SA	Simulated Annealing
SDVRP	Split Delivery Vehicle Routing Problem
Std	Standard Deviation
SVRP	Stochastic Vehicle Routing Problem
TS	Tabu Search
TSP	Traveling Salesman Problem
VRP	Vehicle Routing Problem
VRPB	Vehicle Routing Problem with Backhauls
VRPPD	Vehicle Routing Problem with Pickup and Delivery
VRPTW	Vehicle Routing Problem with Time Windows

Chapter One

General Introduction

CHAPTER ONE

1.1 Over View

An important factor in real life applications is balancing between increasing profits and reducing resources. All companies aspire to achieve this balance that ensures their success and sustainability. However, it is difficult to achieve this balance because all these applications have limited resources and are also very complex. This problems are known as combinatorial optimization problems (COPs) [1].COP is the most important category of improvement problems that concern finding the best solution, from a separate set of all available ones, for a specified problem instance [2].Transport and distribution systems are one of the examples of these problems.

Essentially, the use of effective transport and distribution systems will have three major positive impacts. Clearly, the first is cost reduction, as companies that adopt ineffective transport and distribution systems spend a large amount of their income on the distribution of their goods. Distribution costs may constitute 45% of total logistics costs. According to the Institute of Logistics and Distribution Management 1985, these costs may increase in some companies to 70% of the total cost of goods, such as in the soft drinks industry [3].The second is the environmental issue. There is no doubt that transport accounts for a significant proportion of total greenhouse gas emissions. [4] assessed that these emissions represent 19% of total EU-27 gas emissions, the second largest pollutant after energy industries by 31% [5].The third is linked to huge traffic congestion caused by millions of vehicles required to transport billions of tons of cargo .According to statistical office of the European Community in 2006, the transport of 1904 billion tons of cargo per kilometer was required 32.2 million vehicles [6]. One can imagine massive traffic congestion caused by hundreds of companies distributing thousands of products in a certain area. The vehicle routing problem (VRP) is considered as optimal problems to challenge in distribution systems and transmission [7].

VRP aims to design a range of vehicle routes that serve a group of customers geographically distributed at the lowest cost (minimum travel distance or time) while respecting all restrictions[8]. VRP is an active field of research and has attracted the researcher's attention from various societies such as scientific research and artificial intelligence. Because VRP solution ends efficiently with transportation and distribution systems that can greatly reduce cost, pollution and traffic congestion .There are many basic VRP extensions, such as the vehicle routing problem with pickup and delivery (VRPPD) [9], capacitated vehicle routing problem (CVRP) [7], stochastic vehicle routing problem (SVRP) [10], split delivery vehicle routing problem (SDVRP) [11] and vehicle routing problem with time windows (VRPTW) [12]. We focus in our study on VRPTW which is considered to be the most popular and most widespread widely studied, because it includes the time windows constraint the VRPTW which represents factual life situations [13]. 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 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 [14,15]. Modern studies of VRPTW's solution techniques can be found in [16] focus on exact techniques. 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 [17]. 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, one of the recent algorithms which could to solve several combinatorial problems is Quantum Genetic algorithm (QGA) [18], but it has not been applied to the VRPTW. Therefore, in this thesis, the Quantum Genetic algorithm is proposed for solving the VRPTW.

1.2 Literature Review

Due to the importance of VRPTW, many algorithms have been proposed to address it [15]. These algorithms can be classified into exact (exhaustive), heuristic and meta-heuristic algorithms .Many exact methods had handled with VRPTW such as 2-path cuts for VRPTW, branch-and-bound method for the TCVRP, Lagrangean relaxation and column generation [19]. Exact algorithms are not suitable for large VRPTW problems regardless of the fact that can obtain optimum solutions with guaranteed optimize for small VRPTW problems 25 to 50 instances and a few 100 customer cases, Due to the desired arithmetic time which rises exponentially with the problem [20]. Accordingly, researchers promote to use heuristic algorithms to deal with large-size VRPTW. But no one of them can successfully account for all COPs, due to each one of heuristic algorithms is designed to address a particular problem [2,21]. So they introduced meta-heuristic for exceed this impediment [22]. There are two types of meta-heuristic, single-solution based meta-heuristics and population-based meta-heuristics [2]. Single-solution based meta-heuristics such as tabu search (TS), hill-climbing algorithm (HC), Greedy randomized adaptive search procedure (GRASP), and simulated annealing (SA), population-based meta-heuristics such as Ant Colony Optimization Algorithm (ACO), particle swarm optimization, genetic algorithm and scatter search.

1) **WANPRACHA CHAOVALI TWONGSE, DUKWON KIM and PANOS M.PARDALOS (2003)** adopted A Greedy Randomized Adaptive Search Procedure (GRASP) for vehicle routing problem with time windows (VRPTW) and decreased the travel distance and the number of vehicles. There are two stages at each GRASP iteration local search phase and construction phase. The initial solution is calculated by applying the random greedy adaptation function in the construction phase. The initial solution that was created and obtained from the construction phase for improvement is applied to the search process at the local search phase [23].

2) **B. Ombuki, B.J.Ross, and F.Hanshar (2006)** provided the VRPTW as a multi-objective problem and present a genetic algorithm solution using the Pareto ranking technique. They use a direct interpretation of the VRPTW as a multi-objective problem, in which the two objective dimensions are number of vehicles and total cost (distance). An advantage of this approach is that it is unnecessary to derive weights for a weighted sum scoring formula. This prevents the introduction of solution bias towards either of the problem dimensions. The solution quality of our GA is competitive with the best solutions reported for the VRPTW by other researchers. However, the most significant contribution of this paper is our interpretation of the VRP as a MOP. Our simple translation of the VRPTW into a MOP was surprisingly effective. Firstly, its performance was very good. Our results are competitive with other vehicle-biased results in the literature. Secondly, the Pareto scoring procedure precludes the need to experiment with weights as required in a weighted-sum approach. Poorly chosen weights result in unsatisfactory solutions, and only after considerably experimentation can effective weights be obtained for a specie instance of a VRPTW [24].

3) **Yaw Chang and Lin Chen (2007)** suggested effective and simple genetic algorithm GA for VRP. In way of the average solution, it was considered better than most tabu search published results at that time. They implement this hybrid Genetic algorithm to handle VRPTW. A specific time windows is given by each customer to complete the transfer. When the customer size is small, the results are encouraging. In the future with large number of customers, they will have more comparable experiments [25].

4) **H. Nazif and L.S. Lee (2010)** looked at the application of the genetic algorithm to the vehicle routing problem with time windows where a range of vehicles are available with limits on travel time and capacity to serve a group of customers with demands and the closest and latest time for serving. The aim is to find ways for vehicles to serve all customers at the

lowest possible cost without violating the capacity and time constraints of the vehicle and the time window restrictions determined by the customer. The researchers suggested a genetic algorithm using an optimized crossover engine designed by a complete incomplete binary diagram that finds an optimal set of delivery methods to meet the requirements and give a minimum total cost. To further enhance the quality of solutions, different techniques have been introduced in the proposed algorithm. The results showed that the suggested algorithm is competitive in terms of the quality of existing solutions after testing against the best known solutions reported in the literature and comparing them with some other heuristics in literatures, using 56 Solomon's problems with 100 customers [26].

5) **Filip Taner, Ante Galić and Tonči Carić (2012)** discussed the Vehicle Routing Problem with Time Windows (VRPTW) and showed that it can dramatically reduce the transport costs that occur during the delivery process by implementing algorithms to solve different cases of VRPs. Two meta-heuristic algorithms Simulated Annealing and Iterated Local Search were developed for solving VRPTW. These algorithms generate practical solution using of different operators and strategies and use constructive heuristics for repeated improvement for effectively solve VRPTW, it is necessary to get a practical initial solution in which all conditions are satisfied. Constructive heuristic algorithms usually product solutions of bad quality which help only as a beginning point for further optimize. Solomon I1 heuristic is better-known constructive algorithm for VRPTW. The results obtained showed that the same distribution function could be achieved with savings of up to 40% in the total distance traveled and that routes were built manually are very ineffective [27].

6) **Petr.Kalina and Jiri.Vokrinek (2012)** proposed an efficient algorithm dependent on agent negotiation for the vehicle routing problem with time windows (VRPTW). The algorithm is dependent on state-of-the-art insertion heuristics and a set of general negotiation methods. The

appreciation of generic applicability of agent based to oncoming routing problems in general providing a tough rule for future search in this scope is the major contribution of this work. Experimental results offer that the algorithm is can agree the best-known solutions finished by the centralized solvers in 48.6% of the instances with an average relative error of 3.2% in all tested cases with VRPTW primary optimization standard [28].

7) **Esam Yassen, Masri Ayob, Mohd Zakree, Zulkifli Ahmad (2013)** investigate the performance of HSA in solving the vehicle VRPTW. The performance of HSA is tested using Solomon VRPTW benchmarks. The obtained results demonstrate that the HSA matching the best known result in one instance and obtained promising results in other tested instances. This proves that HSA is a promising method for solving VRPTW [29].

8) **Vaira, G. & Kurasova, O. (2014)** propose a genetic algorithm based on insertion heuristics for the vehicle routing problem with constraints. A random insertion heuristic is used to construct initial solutions and to reconstruct the existing ones. The location where a randomly chosen node will be inserted is selected by calculating an objective function. The process of random insertion preserves stochastic characteristics of the genetic algorithm and preserves feasibility of generated individuals. The defined crossover and mutation operators incorporate random insertion heuristics, analyse individuals and select which parts should be reinserted. Additionally, the second population is used in the mutation process. The second population increases the probability that the solution, obtained in the mutation process, will survive in the first population and increase the probability to find the global optimum. The result comparison shows that the solutions, found by the proposed algorithm, are similar to the optimal solutions obtained by other genetic algorithms. However, in most cases the proposed algorithm finds the solution in a shorter time and it makes this algorithm competitive with others [30].

9) **Koch, H., Henke, T. & Wascher, G. (2016)** introduced a variant of the multi-compartment vehicle routing problem. It includes compartments of flexible sizes, allowing for the number of compartments being smaller than the number of product types, which have to be transported separately. Because of this aspect, an additional question arises concerning the assignment of product types to vehicles. The results of the experiments have shown that only problem instances of limited size can be solved to optimality in reasonable computing time. Based on instances from practice, the benefits of using vehicles with multiple compartments of flexible sizes over using vehicles with a single compartment were investigated. The results for this specific application have shown that – if multiple, flexible-size compartments can be introduced – the total costs of the tours necessary for collecting all supplies can be reduced drastically. These results indicate the necessity of dealing with and developing effective solutions approaches for multi-compartment vehicle routing problems [31].

10) **Yassen, E.Taha (2016)** discussed the effectiveness of selection mechanism on the efficiency of multi-parent crossover. The performance of the proposed algorithm is tested using Solomon VRPTW benchmark.To test these mechanism they selected seven selection mechanisms such as roulette wheel mechanism, random selection mechanism, tournament selection mechanism, best selection mechanism, single best-couple random selection mechanism, stochastic universal sampling mechanism and couple best- single random selection mechanism. The experimental results show the superiority of multi-parent crossover that employs the selection mechanism which selects the outstanding individuals to form most of parents over multi-parent crossover that employ other selection mechanisms. This indicates the efficiency of employing the best parents in the crossover process that can help the search process to reach a better solution [32].

1.3 Problem Statement

Despite the fact that the VRPTW represents the most popular and most widespread widely studied, because it includes the time windows restricting the VRPTW which represents factual life situations, very relevant methods have been suggested a large number of algorithms for the treatment of VRPTW. This indicates their major importance. But, in one hand, none of these algorithms have succeeded in working efficiently in all instances of the problem. Consequently, more efficient algorithm that can significantly work well to improve the quality of obtained solution is highly required. In other hand, Quantum Genetic algorithm (QGA) has been presented as a powerful method to handle many real difficult problems and it has not been applied to the VRPTW. Therefore, in this thesis, the performance of QGA for solving the VRPTW is investigated , in spite of the positive characteristics of QGA and its successful application to address various problems, it still suffers from slow convergence problem when addressing constraint problems especially VRPTW. This problem is attributed to the fact that QGA is effective in exploring the search space but ineffective in exploiting it. In order to compensating the weakness of QGA exploitation, the QGA is combined with one single-based meta-heuristic Hill-Climbing (HC) that is characterized with its effective ability to exploit the search space. As a result of this combination (hybridization) process, the HQGA emerged.

1.4 Research Objectives

This thesis aims to investigate the performance of Quantum Genetic Algorithm and enhance its ability in tackling the Vehicle Routing Problem with Time Window via conducting several modifications. These modifications are concerned with QGA designing and hybridization.

1.5 Thesis Outline

This thesis contains five chapters:

Chapter One: represents the introduction of this thesis, which provides a general view of this thesis, the literature review, problem statement and research objectives.

Chapter Two: this chapter focuses on the combinatorial optimization problem, i.e. the VRPTW which is the focus of this thesis, and the Genetic (GA) and Quantum Genetic (QGA) algorithms to address it.

Chapter Three: presents a detailed explanation of the standard GA, QGA and HQGA about how they are applied for solving VRPTW.

Chapter Four: shows the analysis and the discussion of the results of the proposed GA, QGA and HQGA, which are presented in this thesis.

Chapter Five: conclusions and list of suggestions for future work are given.

Chapter Two

Theoretical Background

Chapter Two

2.1 Introduction

Over the last decade, Quantum mechanical computers were proposed in the early 1980s [33] and the description of quantum mechanical computers was formalized in the late 1980s [34]. Many efforts on quantum computers have progressed actively since the early 1990s because these computers were shown to be more powerful than classical computers on various specialized problems. Quantum Genetic Algorithm (QGA) emerged as a new class of genetic algorithms as it is possible to emulate a quantum computer [18]. QGA is a modern and favorable genetic algorithm which is developed and proposed in the last years, as the product of the combination of quantum computation and genetic algorithms [35]. It is a new evolutional algorithm of eventuality [36]. Wide attention has been attracted by the quantum computation, where it became a very promising research field [18]. The first quantum algorithm made for number factorization was proposed by Shor [36]. Also, a quantum algorithm for random search in databases was proposed by Grover, the complexity of its algorithm was minimized to be of the order of O ($n^{1/2}$) [37]. 2010 year witnessed the proposal of a quantum computing that was used to achieve specific goals in Artificial Intelligence (AI) by Ying [38]. In 1996, QGA is first proposed by Narayanan and Moore and it is successfully used to solve the TSP problem as a probabilistic searching algorithm which exploits the power of quantum computation in order to accelerate genetic procedures [39]. parallel quantum genetic algorithm (PQGA) and quantum genetic algorithm (QGA) are presented by Han and Kim [40] and Han et al [41] to solve an NP-hard combination optimization problem (knapsack problem). In order to enhance performances of QGA [41] in [33], both quantum mutation and quantum crossover were used for that. An enhanced QGA based on multi-qubit encoding and dynamically adjusting the rotation angle mechanism was presented to separate the blind sources [42]. Zhang et. al. proposed a novel PQGA [43] by using a novel evolutionary strategy. They improved QGA by introducing violent vibration and population catastrophe operation [44]. A parallel quantum evolutionary algorithm based on QGA [33] and parallel algorithm was proposed by Li and Jiao [45]. According to the results obtained by [43], the QGA is greatly superior to conventional genetic algorithm (CGA).

2.2 Vehicle Routing Problems

The Vehicle Routing Problem is a class of NP-hard combinatorial optimization problems. It was first proposed by Dantzig who named it the truck dispatching problem [7]. VRP was formulated as a complex extension of the traveling salesman problem with salesmen [7], which have different routes and each vehicle has a specific route. It is one of the most commonly researched problems in the distribution and transportation systems. VRP role is searching for a number of vehicle routes that can be used by a number of customers with the smallest amount of cost (minimum traveling distances). The capacitated vehicle routing problem (CVRP) represents the traditional and common extension of VRP, in which there is one restriction that is the vehicle capacity. Consequently, in every route, the overall demands of all customers should not be beyond the capacity of the vehicle. In CVRP, the fleet of vehicles is homogeneous, i.e. the capacities of all vehicles are similar [46].

Due to the rapid growth in real world applications and the increase of their requirements, quite a lot of extensions have been made to the basic VRP. The rest of this section presents an overview of the most renowned VRP variants:

- **Split Delivery Vehicle Routing Problem**

  The split delivery VRP represents the extension of VRP in which every customer can have a demand more than the vehicle capacity. In this variant, the constraint that a customer must be visited only once should be canceled, to permit the split of the customer demand, and consequently the customer can be served multi times till fulfilling his demand [11].

- **Site-Dependent Vehicle Routing Problem**

  In the site-dependent VRP, the site dependent constraint must be taken into account when the vehicle routes are planned. In this variant, the fleet composes different vehicle types (each vehicle has a specific capacity and size) and there is a relation between the vehicle and customer types. For example, the big vehicle cannot serve a customer who lives in a narrow street, and a customer whose demand cannot be served by a small vehicle. This means that the vehicle type and customer demand highly affect each other [47].

- **Heterogeneous Vehicle Routing Problem**

The heterogeneous VRP is a variant of CVRP with the depot of different vehicle types, i.e. the vehicles of each type have a specific capacity. So, the HVRP solution composes of multi routes and each one is associated with the type of vehicle [48].

- **Period Vehicle Routing Problem**

The period VRP deals with planning for vehicle routes over a multi-day period. In this extension, any customer may be visited multi times and these visits are organized based on allowable combinations set of distribution days [49].

- **Vehicle Routing Problem with Pickup and Delivery**

There is a VRP with pickup and delivery (VRPPD) where customers may return some of the goods on condition that the delivering vehicle has the capacity to pick up these goods [9]. Some other variants, similar to this variant, have been suggested such as VRP with backhauls (VRPB). The VRPB does not permit returning the goods until all demanded goods are delivered [50].

- **Multiple Trips Vehicle Routing Problem**

In multiple trips, VRP the vehicle can be up to the depot multi times in order to load and unload the goods w.r.t. the maximum duration restriction [51].

- **Three-dimensional Loading Capacitated Vehicle Routing Problem**

  The three-dimensional loading capacitated VRP resulted from the combination of two problems: VRP and three dimensional loading problem. The solution of this VRP variant must guarantee the feasible item packaging feasibility in addition to meeting all customers' demands and reducing the overall distance costs. Unlike the CVRP where the overall weight of items must not violate the vehicle capacity, in this extension, the three dimensions of each item should be taken into consideration in the process of packaging besides the weight of that item [52].

- **Vehicle Routing Problem with Time Windows**

  The VRP with time windows (VRPTW) is one of the most widely studied. VRP studied variants that include capacity and time windows constraints. The time windows constraint is the most importance factor in the VRPTW because it reflects the real life situations. The VRPTW, which is the problem under discussion in this thesis, it is described in details in the following section [13].

- **Time Dependent Vehicle Routing Problem**

  Unlike most of VRP extensions which identify the travel time between any two vertices based on the distance between them only, the time dependent VRP variant takes into account the vehicle departure time as well as the distance. In real word applications, many factors can have a significant impact on the travel speeds and times, such as traffic jams and routing problems. Thus, the vehicle departure time must be respected when the travel time for each vehicle is identified [53].

- **Stochastic Vehicle Routing Problem**

  The stochastic VRP includes all extensions of VRP where one or more VRP factors are undefined. Factors such as customer's availability and demands, and service time could be random and not known in advance [52].

2.2.1 Vehicle Routing Problem with Time Window

VRPTW is the most popular and most widely studied extension because the time window restriction of VRPTW represents actual life situations, such as school bus, emergency (police, fire) and logistics. Therefore, this work focuses on VRPTW in which a given set of customers are geographically distributed. Each customer requires a certain amount of goods to be loaded/unloaded (demand). a customer is served during a certain time window (the time requested by the customer) by a set of vehicles of specified capacities [13], as shown in figure (2.1).

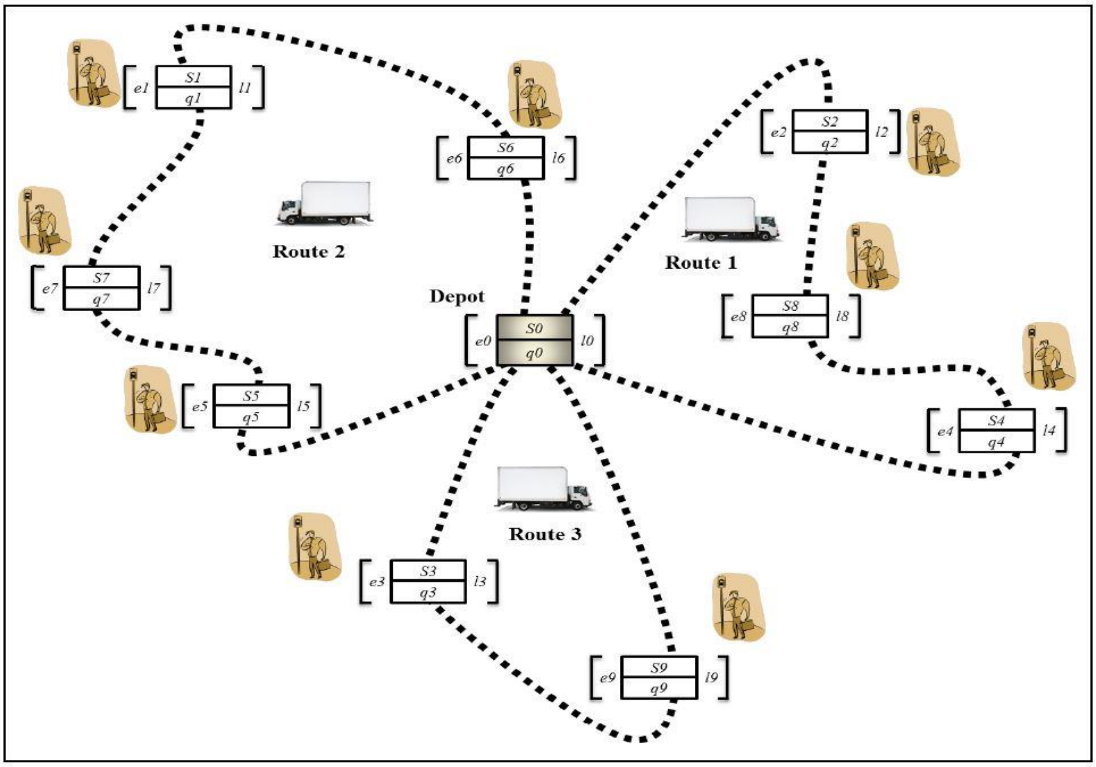

Figure (2.1): A Solution for a VRPTW with 9 Customers and 3 Vehicles.

Main components of vehicle routing problem are:

- **Customers:** The main aim of the VRPTW is to meet its customers' demands. The locations of those customers are around the depot.

- **Vehicles:** In each VRPTW, there is a particular number of vehicles that can have a maximum traveling time, a cost, a specified maximum time, or capacity.
- **Routes:** the various paths that are taken by a vehicle in order to serve the customers. These routes may vary in travel time and cost.
- **Depots:** A depot is considered as the starting and finishing points of any VRPTW.

The goal is to design a set of vehicle routes to serve all customers at the minimal cost (minimum traveling distance) while respecting the following conditions (the VRPTW hard constraints):

1) Each vehicle should begin and end at the depot.
2) Each customer should be visited exactly once during its time window.
3) Split deliveries should be avoided.
4) The total demands of all customers in each route must not exceed the capacity of the vehicle.

While, the only one VRPTW soft constraint is reducing the total traveling distance. Although the soft constraint can be violated, the performance of the heuristics is measured by their ability to achieve these soft constraints [54].

2.2.2 Vehicle Routing Problem with Time Windows Formulation

The VRPTW is defined as follows: assume that G (V, E) is an undirected graph, where V represents a group of nodes $V = \{0,1, 2,\ldots, n\}$, node 0 is the depot, and customers are represented by nodes $1,2,\ldots, n$. E is a set of edges, $E= \{(i, j): i \neq j$ and i, j €$V\}$ and each edge is comes along with a travel distance cost $c_{ij} = c_{ji}$ and $c_{ji} > 0$. Each customer, except the depot, has a specific demand and service time that must be known in advance. In VRPTW, there is a fixed number of vehicles (v). Each vehicle has a capacity, which is known in advance. If the vehicle arrives at the customer before the beginning of its start time window, the vehicle must wait until its time window begins. On the other hand, the vehicle cannot serve the customer if it arrives after the end of the customer's time window. The goal of solving VRPTW is to generate feasible routes to serve all customers with minimal cost. Assume the following variables:

$v \rightarrow$ number of serving vehicles

$n \rightarrow$ number of customer nodes (excluding the depot node)

$q_i \rightarrow$ demand of costumer c_i

$Q_k \rightarrow$ capacity of the k^{th} vehicle

$s_i \rightarrow$ service duration of costumer c_i

$t_i \rightarrow$ arrival time at customer c_i

$t_{ij} \rightarrow$ travel time (travel distance) from customer c_i to customer c_j

$W_i \rightarrow$ waiting time at customer c_i

$e_i \rightarrow$ start of time window for customer c_i

$l_i \rightarrow$ end of time window for customer c_i

The VRPTW can be described mathematically as follows [20]:

$$x_{ij}^k = \begin{cases} 1 & \text{if vehicle k travelled directly from customer } c_i \text{ to } c_j \\ 0 & \text{Otherwise} \end{cases}$$

$$y_i^k = \begin{cases} 1 & \text{if customer } c_i \text{ is served by vehicle k} \\ 0 & \text{Otherwise} \end{cases}$$

The quality of the solution *S* is measured using an objective function, as in eq. (2.1).

$$\boldsymbol{f(s) = min \sum_{i=0}^{n} \sum_{j=0}^{n} \sum_{k=0}^{v} t_{ij} \times x_{ij}^k} \tag{2.1}$$

$$\boldsymbol{\sum_{i=0}^{n} x_{ij}^k = y_j^k, \forall k = 1, \dots, v, \forall i = 1, \dots, n} \tag{2.2}$$

$$\boldsymbol{\sum_{j=0}^{n} x_{ij}^k = y_i^k, \forall k = 1, \dots, v, \forall i = 1, \dots, n} \tag{2.3}$$

$$\boldsymbol{\sum_{i=0}^{n} y_i^k \times q_i \leq Q_k, \forall k = 1, \dots, v} \tag{2.4}$$

$$\sum_{k=1}^{v} \boldsymbol{y_i^k} = \mathbf{1}, \forall \boldsymbol{i} = \mathbf{1}, \dots, \boldsymbol{n} \tag{2.5}$$

$$\sum_{k=1}^{v} \boldsymbol{y_0^k} = \boldsymbol{v} \tag{2.6}$$

$$\boldsymbol{t_i} \ + \boldsymbol{w_i} \ + \boldsymbol{s_i} \ + \boldsymbol{t_{ij}} \ = \boldsymbol{t_i} \ , \forall \boldsymbol{ij} = \mathbf{0}, \mathbf{1}, \mathbf{2} \dots .. \boldsymbol{n}, \boldsymbol{i} \neq \boldsymbol{j} \tag{2.7}$$

$$\boldsymbol{e_i} \ \leq \ \boldsymbol{t_i} \ \leq \boldsymbol{l_i} \ , \forall \boldsymbol{i} = \mathbf{0}, \mathbf{1}, \mathbf{2} \dots . \boldsymbol{n} \tag{2.8}$$

$$\boldsymbol{w_i} \ = \max\{\boldsymbol{e_i} \ - \boldsymbol{t_i} \ , \boldsymbol{O}\} \ \forall \boldsymbol{i} = \mathbf{0}, \mathbf{1}, \mathbf{2}, \dots . \boldsymbol{n} \tag{2.9}$$

Constraints (2.2) and (2.3) guarantee that each vehicle can enter and leave from any customer, provided that it serves that particular customer. Constraint (2.4) checks that the capacity of the vehicle is not exceeded. Constraint (2.5) confirms that each customer is served only once. Constraint (2.6) ensures that each vehicle must start from the depot. Constraints (2.7), (2.9) verify the time constraints to make sure that no time window is violated. More details about VRPTW can be found in [14,15].

2.3 Quantum Computing

As mentioned previously Quantum mechanical computers were proposed in the early 1980s [33] and the description of quantum mechanical computers was formalized in the late 1980s [34]. Quantum computation is contrary to classical computation.

It uses the superposition, coherence, and the entanglement of different *qubits* of quantum state to realize quantum computation [54]. Quantum computation is the produced by quantum mechanics applying in the field of algorithm. Parallelism is the essential difference between classical computation and quantum computation. In the calculations of probability, the system is not in a constant state. On the contrary, it has a particular probability and the state probability vector corresponds to various possible states. Similarly, the probability amplitudes of quantum states is used in the quantum computation and probability amplitudes of quantum states are squared normalized, so the calculation speed of quantum computation is faster than that of classic calculation by $\sqrt{N}$ times. Quantum rotating gates are used to realize quantum transformation. After Quantum Computing was first proposed in the 1980s, its combining with the subjects in other areas is an important forefront topic recently. There are well-known quantum algorithms such as Shor's quantum factoring algorithm [36] and Grover's database search algorithm [37]. Research on merging evolutionary computing and quantum computing has been started since late 1990s.

2.3.1 Quantum Computing Basics

The basics of quantum computing are addressed briefly in this subsection. The smallest unit of information stored in a two-state quantum computer is called a quantum bit or *qubit* [55]. A *qubit* may be in the "1" state, in the "0" state, or in any superposition of the two. The state of a *qubit* can be represented as:

$$|\Psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \qquad (2.10)$$

Where $|0\rangle$ and $|1\rangle$ represent the values of classical bits *0* and *1* respectively, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are complex numbers satisfying:

$$|\alpha|^2 + |\beta|^2 = 1 \qquad (2.11)$$

Where α and β are complex numbers that specify the probability amplitudes of the corresponding states. α gives the probability that the *qubit* will be found in the "0" state and β gives the probability that the *qubit* will be found in the "1" state [18].

2.3.2 Properties of Quantum Computation

Quantum computation and classical computation share a number of properties such as:

(1) Superposition of quantum computation: *qubit* is the most basic unit of quantum computers for information storage [56]. n-bits quantum register can found be in the state $|\phi\rangle$, which is the coherent superposition state of 2^n ground states. An act on a n-bits quantum register is equal to the act on 2^n numbers. The relationship between superposition state and ground states is

$$|\emptyset\rangle = \sum_{\mathrm{i}} C_i |\emptyset \mathrm{i}\rangle \qquad (2.12)$$

Where C_i represents the probability amplitude of the state $|\phi i\rangle$; $|C_i|^2$ is the probability to collapse to the ground state $|\phi \mathrm{i}\rangle$ when the state $|\phi\rangle$ is measured; C_i should satisfy the condition [57]:

$$\sum_{\mathrm{i}} |C_i|^2 = 1 \qquad (2.13)$$

(2) Coherence of quantum computation: it differs from the classical computation; coherence is an important property of quantum computation. The relative phase of the respective ground states alters with the interferences that occur to each ground state by the action of quantum rotating gates. For example, a quantum system of a single *qubit* is $|\phi\rangle = C_1|0\rangle + C_2|1\rangle$ and $|C_1|^2+|C_2|^2=1$. After process of few of quantum gate, the values of C_1 and C_2 correspondingly occur a change, and the new C_1 and C_2 still give the result $|C_1|^2 + |C_2|^2 = 1$. So, if the probability amplitude of the ground state $|0\rangle$ increased, the probability amplitude of the ground state $|1\rangle$ will be reduced. When the quantum system is measured, coherence will disappear and collapse to some ground state, while the probability is determined by $|C_i|^2$ [18].

(3) Entanglement of the quantum state: the quantum state which cannot be divided into the form of the direct product of two subsystems is referred as entanglement state .Once an operation is performed on one or more specific *qubits* both the operated state of the *qubit* and the state of the other *qubit* entangled occur changes [58].

(4) Parallelism of the quantum computation: the same quantum circuit achieves the parallelism of quantum computation rather than implementing it by multiple hardware calculated simultaneously. Parallelism of quantum computation uses the superimposed ability of various states which the quantum computer lies in; therefore, using a single (x) circuit can calculate multiple values of x.

Generally, the complexity of quantum algorithm is less than its classic equivalent algorithms through the concept of quantum superposition [18].

2.4 Genetic Algorithm

Genetic algorithm (GA) was proposed by Holland (1975) [59]. Since then GA has been popular, because it can contribute to detect good solutions for complex mathematical problems in a reasonable amount of time [2]. Genetic Algorithms (GA) are a representative example of a set of methods known as evolutionary algorithms. The principles of GA are familiar. A population of solutions (chromosomes in the GA) is kept along with a crossover process allowing the parent solutions to be chosen from the population. Offspring solutions are produced which exhibit some of the characteristics of each parent. The fitness of each solution can be related to the objective function value [59]. Thangiah et al. (1991) have used GA firstly to tackle the VRPTW [60]. The main principle here is that only the fittest entities survive [61]. The major procedures within GA include the selection, crossover, mutation and updating processes as shown in algorithm (2.1) [62].

Algorithm 2.1: A Typical Genetic Algorithm

Start

P = initial population;

Evaluate (P);

While termination criterion not satisfied **Do**

P' = recombines (selected (P));

Mutate (P');

Evaluate (P');

replace (P');

End while

End

A general formwork of GA can be described as follow:

- **Initial population**

  An initial population of chromosomes will be generated randomly and arranged according to their corresponding objective function values.

- **Evaluation**

  Once the offspring is created or a population is initialized, an evaluation of the fitness values of the candidate solutions is performed.

- **Selection**

  Selection allocates more copies of those solutions with higher fitness values and thus imposes the survival-of-the-fittest mechanism on the candidate solutions. The essential idea of selection is to prioritize better solutions on worse ones, and in order to achieve the idea many selection procedures have been proposed, including tournament selection, ranking selection roulette-wheel selection [63] and stochastic universal selection.

- **Crossover**

  Recombination is the process of combining parts of two or more parental solutions to create a new and possibly better solution (i.e. offspring). There are lots of methods to accomplish this, and the competent performance depends on a good design of recombination mechanism. The offspring under recombination will not be similar to any particular parent, it will instead combine parental traits in an unfamiliar manner [64].

- **Mutation**

  While recombination operates on two or more parental chromosomes, mutation locally but randomly preforms a modification on a solution. Again, there are a large number of mutation types, but it usually includes a single or multiple changes being made to an individual's trait or traits. This means that candidate solution will pass through a random walk performed by mutation [61].

- **Replacement**

  The offspring population generated by selection, recombination, and mutation replaces the original parental population. GAs use a big number of replacement techniques such as elitist replacement, generation wise replacement and steady-state replacement methods [62].

2.5 Quantum Genetic Algorithm (QGA)

Based on the concept and principles of quantum computing the Quantum genetic algorithm is emerged. In 1996, quantum genetic algorithm is first proposed by Narayanan and Moore to solve successfully traveling salesman problem (TSP), By Presenting *qubit* representation and quantum logic gate operation the QGA are highly effective and robust over a traveling salesman problem TSP [39]. QGA combines some characteristics of quantum computation with the genetic algorithm, 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 [18]. QGA is essentially a kind of genetic algorithm and can be applied in the field that the conventional genetic algorithm can be applied. However, the concepts of *qubits* and superposition of states of quantum mechanics represent the base of QGA. The smallest unit of information stored in a two-state quantum computer is called a quantum or *qubit*. A Q-bit may have the state of 1, the 0 state or in any superposition of the two [65]. QGA is a probabilistic searching algorithm which exploits the power of quantum computation in order to accelerate genetic procedures [18]. The state of a *qubit* can be represented by eq. (2.10). A *qubit* representation of *m-qubit* chromosome as shown in figure (2.2).

$$\boxed{\begin{bmatrix} \alpha_1 \\ \beta_1 \end{bmatrix} \begin{vmatrix} \alpha_2 \\ \beta_2 \end{vmatrix} \begin{matrix} \dots \\ \dots \end{matrix} \begin{vmatrix} \alpha_m \\ \beta_m \end{bmatrix}}$$

Figure (2.2): Q*ubit* Representation.

Where *m* represents the length of the chromosome. In this way, the *m qubit* 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. 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: $\frac{\mathbf{1}}{\mathbf{8}} < \mathbf{0}, \mathbf{0} >,\ \frac{\mathbf{3}}{\mathbf{8}} < \mathbf{0}, \mathbf{1} >,\ \frac{\mathbf{1}}{\mathbf{8}} < \mathbf{1}, \mathbf{0} >,\ \mathbf{and}\ \frac{\mathbf{3}}{\mathbf{8}} < \mathbf{1}, \mathbf{1} >$

Where $\frac{\mathbf{1}}{\mathbf{8}}, \frac{\mathbf{3}}{\mathbf{8}}, \frac{\mathbf{1}}{\mathbf{8}}$ and, $\frac{\mathbf{3}}{\mathbf{8}}$ are probabilities [40].

2.5.1 Quantum Genetic Algorithm Structure

The structure of a QGA is illustrated in figure (2.3) [44].

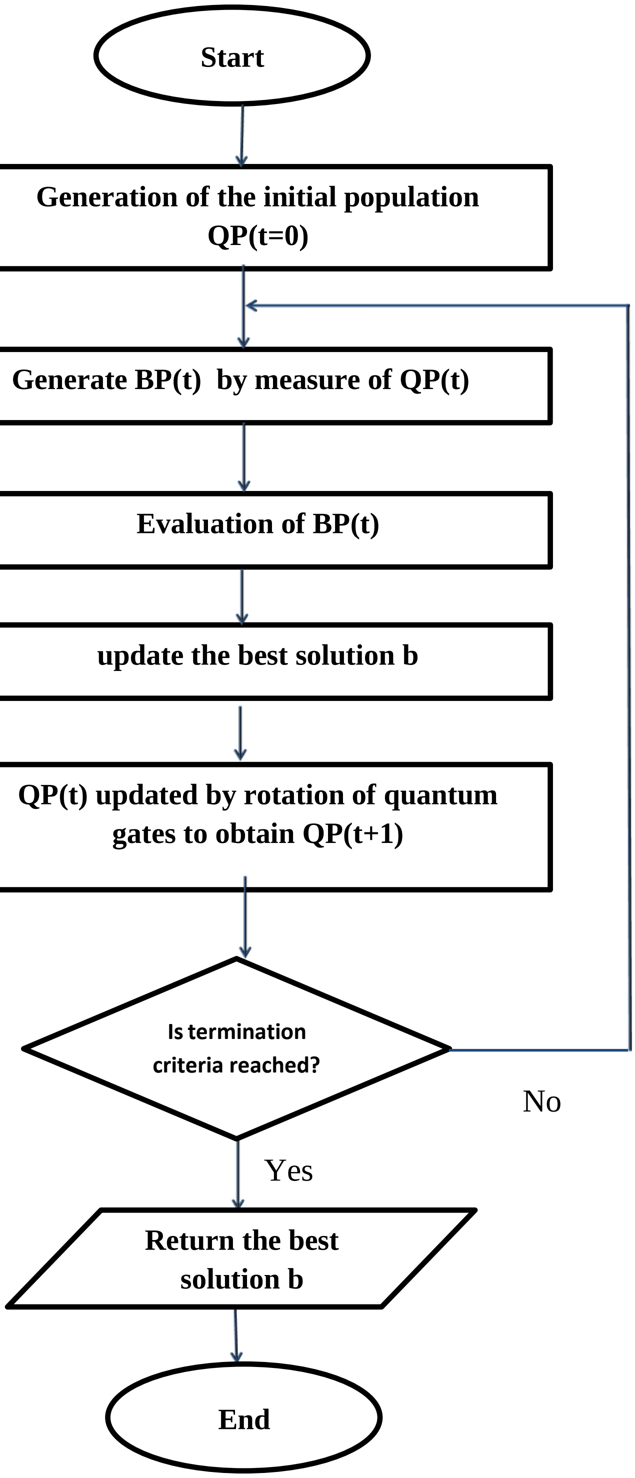

Figure (2.3): QGA Structure.

The basic QGA consist of the following steps [44]:

Step 1: Generation of the initial population

In this stage, an initial quantum population (QP) is generated. The most easy method of creating the initial population is initializing all the amplitudes of *qubits* by the value ($1/\sqrt{2}$). This means that a chromosome represents all quantum superposition states with equal probability.

A chromosome is simply represented as a string of *m qubits* by which a quantum register is formed. Figure (2.4) shows the structure of a quantum chromosome. In each quantum chromosome (q), the gene is represented by a *qubit* in QGA.

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

Figure (2.4): Quantum Chromosome Structure.

Step 2: Generate BP(t) population

In this step, a population BP(t), which composed of classical chromosomes, is obtained by measuring or observation of *qubits* states in the quantum chromosomes of the QP(t). BP(t)={ x_1^t , x_2^t , …,x_j^t }, where x_j^t is a binary string and is formed by selecting each bit using the probability amplitude of *qubit* in a quantum chromosome [65]. This leads to the extraction of a classic chromosome as illustrated in figure (2.5).

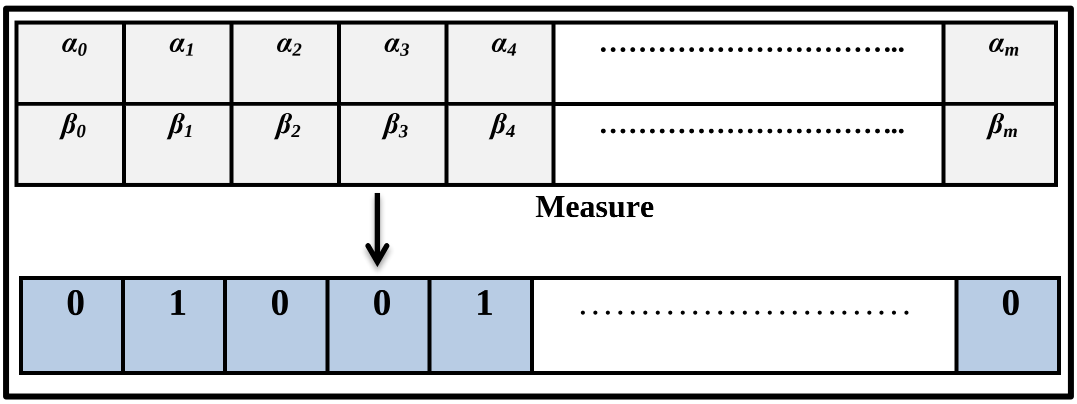

Figure (2.5): Binary Chromosome Representation.

A simple way to implement this function is given by the following pseudo code, see figure (2.6).

```
/* Generate random number */
        R = rand ∈ ( 0,1)
        if (R > α^2)
                return 1
        else
                return 0
end if
```

Figure (2.6): Measuring Process [38].

Step 3: Evaluation of individuals

This phase role is quantifying the quality of each quantum chromosome in the population to form a reproduction (crossover). The evaluation depends on an objective function that corresponds to each individual, after measuring, an adaptation value. It allows marking the individuals in the population.

Step 4: Update the best solution

Based on the quality of each chromosome in BP(t), the best solution *best* is updating as shown in figure (2.7).

```
b ← the best chromosome within BP(t)
if (b better than best)
        best ← b
```

Figure (2.7): Update the Best Solution Pseudo Code.

Step 5: Update quantum population

In this step, the quantum population QP(t+1) is updated according to a quantum rotation gate U(t), as follows [18]:

The rotation of individual's amplitudes is performed by quantum gates. Quantum gates can be designed in accordance with the present problem. This rotation strategy is given by the following equation:

$$\begin{pmatrix} \alpha_j^{t+1} \\ \beta_j^{t+1} \end{pmatrix} = \begin{pmatrix} cos(\delta\theta_j) & -sin(\delta\theta_j) \\ sin(\delta\theta_j) & cos(\delta\theta_j) \end{pmatrix} \begin{pmatrix} \alpha_j^t \\ \beta_j^t \end{pmatrix} \tag{2.15}$$

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:

$$\delta\theta = sg(\alpha_j , \beta_j)\Delta\theta_j \tag{2.16}$$

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 2.1), comparing the fitness of the best individual $f(best_j)$ with the fitness of the current chromosome $f(x_j)$. According to the problem type (minimization or maximization) 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 2.1: Lookup Table of the Rotation Angle (θ 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:

$$\Delta\theta_{\mathrm{j}} = 0.005\pi + (0.05\pi - 0.005\pi)\frac{|f(x_j) - f(b_j)|}{max(f(x_j), f(b_j))} \qquad (2.17)$$

Finally, the new quantum population QP(t+1) is generated [44].

Step 6: Termination process

This step examines the QGA termination criterion. If the termination criterion is fulfilled, QGA will halt and give back the best obtained solution. If not, the steps from step 2-5 are reiterated.

2.5.2 Comparison between Quantum Genetic Algorithm and Conventional Genetic Algorithm.

QGA is essentially a kind of genetic algorithm and can be applied in the field that the conventional genetic algorithm can be applied. Quantum genetic algorithm is a more wonderful optimization process than the conventional genetic algorithm, and its encoding mode is more complex and each generation of the evolution can cover a wider area [66]. The quantum genetic algorithm

(QGA) results are greatly superior to conventional genetic algorithm (CGA) [43] leads QGA to behave better than CGA and this in all problem solution variants.

The quantum state vector is introduced in the Genetic Algorithm to express genetic code, and quantum logic gates are used to realize the chromosome evolution. By these means, better results are achieved [36].

2.5.3 QGA Applications

QGA has been shown as an effective approach for tackling numerous real world difficult problems [56], such as the personnel scheduling problem shows that compared with GA, QGA can get its smallest value in a short time; however, QGA can hardly get the optimal result. Though the evolution time of the IQGA is longer and the evolution generation of the IQGA is more, the IQGA can get the optimal result in a bigger probability [57]. Cryptanalysis shows that QGAs were utilized in the cryptanalysis of TEA. They not only significantly enhanced the results in [67] and [68] in terms of both weight and bitmask chi-square statistic, but also had the ability of breaking TEA of cycles greater than or equal to four that formed an unsolved challenge for previous studies. With these enhanced bitmasks, efficient distinguishers for TEA can be built. A small number of inputs are required by these distinguishers require to obtain a high distinguishing probability [67]. Where a quantum-inspired differential evolution algorithm was proposed to solve the N-queens problem that show the hybrid algorithm has given better results in term either of computation time or in term of fitness evolution. To conclude, the quantum-inspired differential evolution algorithm is well situated to be among the good alternatives to solve combinatorial optimization problems, especially in term of efficiency and algorithmic complexity [69]. A QGA was also used to solve the binary decision diagram ordering problem. The obtained results are encouraging and attest the feasibility and the effectiveness of the approach. QGABDD is distinguished by a reduced population size and a reasonable number of iterations to find the best order, thanks to the principles of quantum computing [70].

2.6 The Solomon's VRPTW Benchmark

There are some sets of benchmarks which have been defined for VRPTW. According to the number of customers, these benchmarks are categorized into small and large, where the small benchmarks contain 25 and 50 customers, while large benchmarks contain 100 customers [71]. Solomon's VRPTW benchmark [12] is used in this study 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 [14,15]. 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. Three types of information are provided on each instance, which are instance name, vehicle specifications and customer specifications. 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, Equations (2.1). 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 [14]. Table 3.2 shows the best-known result for each individual instance, regarding both the number of routes and the travel distance were collected from [72].

Table (2.2): Best-Known Results for the Solomon's Benchmark Set.

Instances	*BK*	Instances	*BK*	Instances	*BK*
R1-01	1642.87	R2-08	723.61	C2-07	588.29
R1-02	1472.62	R2-09	879.53	C2-08	588.32
R1-03	1213.62	R2-10	932.89	RC1-01	1623.58
R1-04	982.01	R2-11	787.51	RC1-02	1466.84
R1-05	1360.78	C1-01	828.94	RC1-03	1261.67
R1-06	1241.51	C1-02	828.94	RC1-04	1135.48
R1-07	1076.13	C1-03	828.07	RC1-05	1518.6
R1-08	948.57	C1-04	824.78	RC1-06	1377.35
R1-09	1151.84	C1-05	828.94	RC1-07	1212.83
R1-10	1080.39	C1-06	828.94	RC1-08	1117.52
R1-11	1053.49	C1-07	828.94	RC2-01	1274.54
R1-12	953.63	C1-08	828.94	RC2-02	1113.53
R2-01	1148.48	C1-09	828.94	RC2-03	945.96
R2-02	1049.74	C2-01	591.56	RC2-04	798.41
R2-03	900.08	C2-02	591.56	RC2-05	1161.81
R2-04	772.33	C2-03	591.17	RC2-06	1059.89
R2-05	970.89	C2-04	590.6	RC2-07	976.4
R2-06	898.91	C2-05	588.88	RC2-08	795.39
R2-07	814.78	C2-06	588.49		

Chapter Three

The proposed System

Chapter Three

3.1 Introduction

Due to the fact that the QGA has been successfully applied to solve many combinatorial optimization problems, but has not been applied to solve VRPTW, this thesis investigates the performance of QGA in solving the VRPTW and Hybrid Quantum Genetic algorithm (HQGA).

In this chapter, the designing of constructive and improvement algorithms is done. The constructive algorithm deals with how a population is initially constructed, while the improvement algorithms; Genetic algorithm (GA), Quantum Genetic algorithm (QGA) and Hybrid Quantum Genetic algorithm (HQGA); deals with how the quality of that population is iteratively improved. This chapter is organized as follows: The standard GA and QGA for VRPTW in Sections 3.2 and 3.3 respectively. Finally, the proposed hybrid QGA is presented in Section 3.4.

3.2 Genetic Algorithm for VRPTW

In order to investigate the performance of QGA in solving the VRPTW, its obtained results are compared with these of GA. So, the GA is designed and implemented to solve the VRPTW. The GA consists of six steps as shown in algorithm (2.1).

Step1: Initial population

In this work, GA population is randomly initialized. In VRPTW, each GA chromosome represents the VRPTW solution, which is represented as a vector to store the set of routes. Each route includes the customers who are served by the vehicle of that route. Each route starts and ends at the depot (where depot is labeled as ‘0’ as shown in figure 3.1).

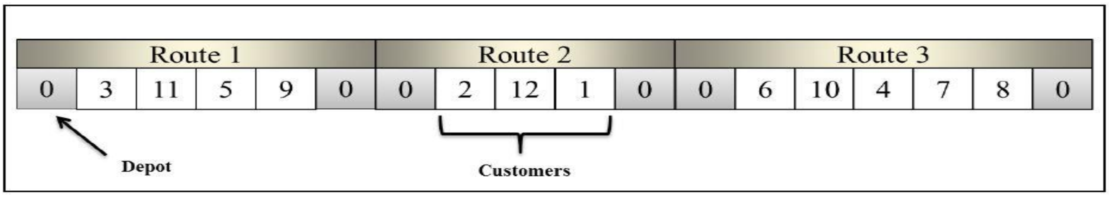

Figure (3.1): The Representation of Solution with 12 Customers and 3 Routes.

In this work, the GA population is initialized according to the following steps:

1. Generate an empty route *r*.

2. Randomly chooses a seed customer, and inserts it into the route *r*.

3. As long as the VRPTW constraints are not violated, randomly select other un-routed customer and insert it into *r* after the last inserted one.

4. Repeat the third step until no more customers can be inserted.

5. The process of creating new routes and inserting customers is repeated until all customers have been routed.

Step2: Evaluation

After initializing the population, each VRPTW solution is associated with its fitness values according to the equation (2.1). In VRPTW, the goal is to generate feasible routes to serve all customers with minimal cost (Distance) According this equation: $\boldsymbol{d}(\mathrm{i},\mathrm{j}) = \sqrt{(\mathrm{x_j} - \mathrm{x_i})^2 + (\mathrm{y_j} - \mathrm{y_i})^2}$.

Step3: Selection

After that, according to the fitness value, the process of selection is employed to choose best 50% of the candidate solutions (population) known as the parents. These selected parents are adopted in a recombination process.

Step4: Crossover

In this process, the order crossover (OX2) is performed upon the selected parents as follows, see figure (3.2):

1. Each two parents (p1 and p2) will pass through the crossover operator to generate new two children (c1 and c2).
2. A certain position (*pos*) is randomly determined over the length of p1 and p2.
3. Regarding to the first children:
    - 3.1 Create an empty solution c1.

3.2 From the parent *p1*, the left segment side of the *pos* are inserted into the c1.

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

4. Regarding to the second children:

4.1 Create an empty solution c2.

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

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

▼ *Pos*

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 (3.2): Example of Crossover.

Step5: Mutation

In case minor modifications on the generated children c1 and c2 are required, the Mutation operator is utilized as follows: randomly select two diferent customers in each child and exchange their positions as shown in figure (3.3).

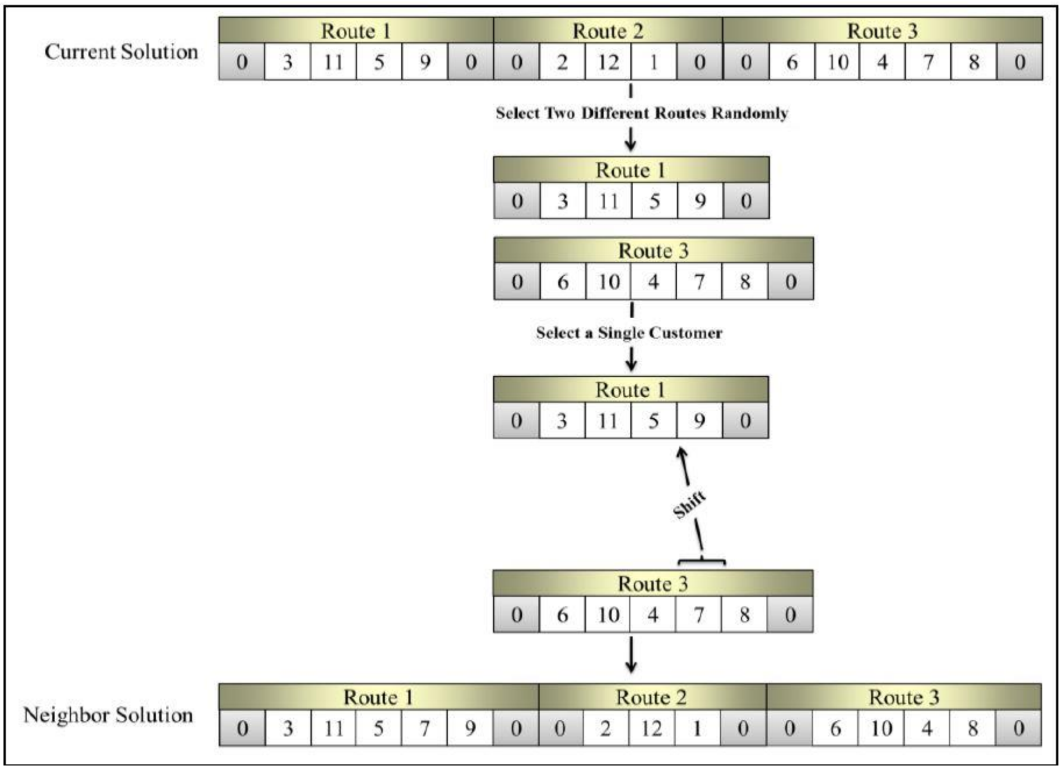

Figure (3.3): Example of Mutation.

Step6: Termination process

This step checks the termination criterion of GA. If the termination criterion is satisfied, GA will stop and return the best solution found. Otherwise, the steps 2-5 are reiterated.

3.3 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 (3.4)

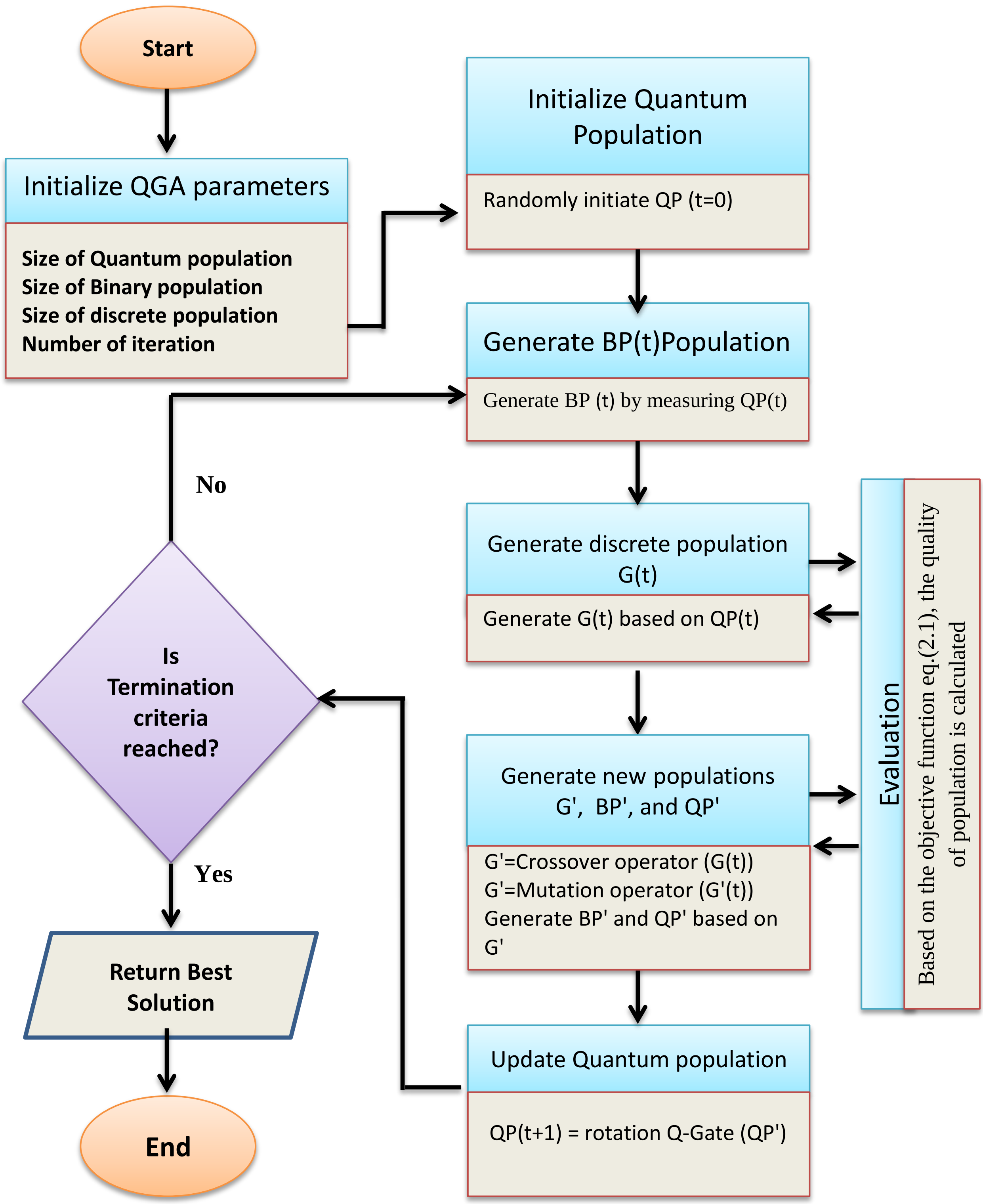

Figure (3.4): QGA for VRPTW.

The procedure of QGA is described in the following steps:

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

At the beginning, the QGA initializes a quantum population *QP(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:

$$|\boldsymbol{\alpha}|^2 + |\boldsymbol{\beta}|^2 = 1 \tag{3.1}$$

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

After initialization, a binary population *BP(t)* is generated by measuring *QP(t).* *BP* represented as vector *BP(t)*={ x_1^t , x_2^t , …,x_j^t }, where x_j^t is a binary value which is generated using process shown in figure (2.6). Figure (3.5) shows the structure of generated binary chromosome.

0	**1**	**0**	**0**	**1**	**0**	**1**	**0**

Figure (3.5): Binary Chromosome.

Step3: Generate discrete population *G(t)*

In this step the VRPTW population is generated based on *QP(t).* The G(t) consists of *n* VRPTW solutions (chromosomes) and each solution involves *m* gene (customer ID). The random key algorithm [73] is utilized to generate VRPTW solutions as follows: For each quantum chromosome in QP(t) do the following: see figure (3.6)

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 (3.6): VRPTW Solution.

Step4: Evaluation

After generating the G(t) population, The feasibility of each VRPTW solution is checked based on the problem constraints (2,2) (2,3),(2,4),(2.5),(2.6),(2.7),(2.8) and (2.9). And the repair mechanism is utilized to rectify the infeasible solution (section 3.5). After that all solutions in G(t) are associated with its fitness values according to the equation (2.1).

Step5: Generate new populations, as follows:

1. Applying crossover operator on the G(t) to generate a new population G'(t). This crossover operator was explained in details in section (3.2).
2. Mutation operator: the same mutation process which was explained in section (3.2) is adopted in this step on the G'(t).
3. Based on G'(t), generate QP'(t) and BP'(t).

Step6: Update quantum population QP(t+1)

As in standard QGA, the quantum population QP(t+1) is updated with a quantum gates rotation of *qubits* constituting individuals based on QP'(t). This process explained in details in chapter two, section (2.5.1).

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, the steps 2-5 are reiterated.

3.4 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) 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 (3.1) [74].

Algorithm 3.1: 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 2-opt star neighborhood operator [75]. The 2-opt star operator randomly selects two routes from the current solution and swap the customers located at the end sections of the selected routes, see example in figure (3.7).

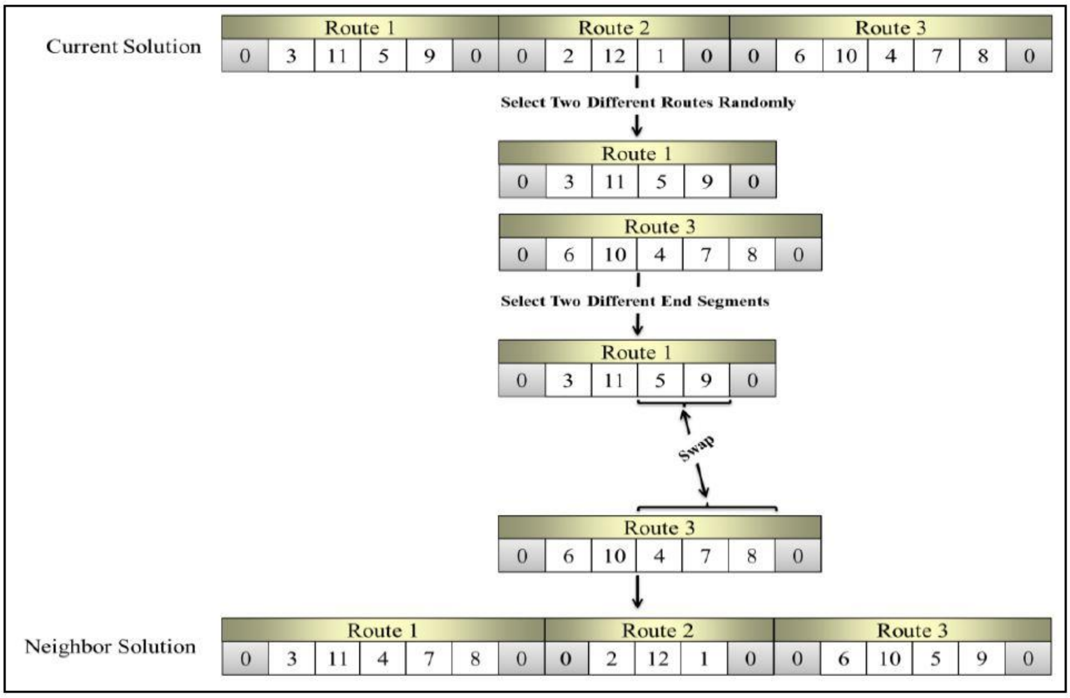

Figure (3.7): Two-opt Star Strategy.

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 (3.8).

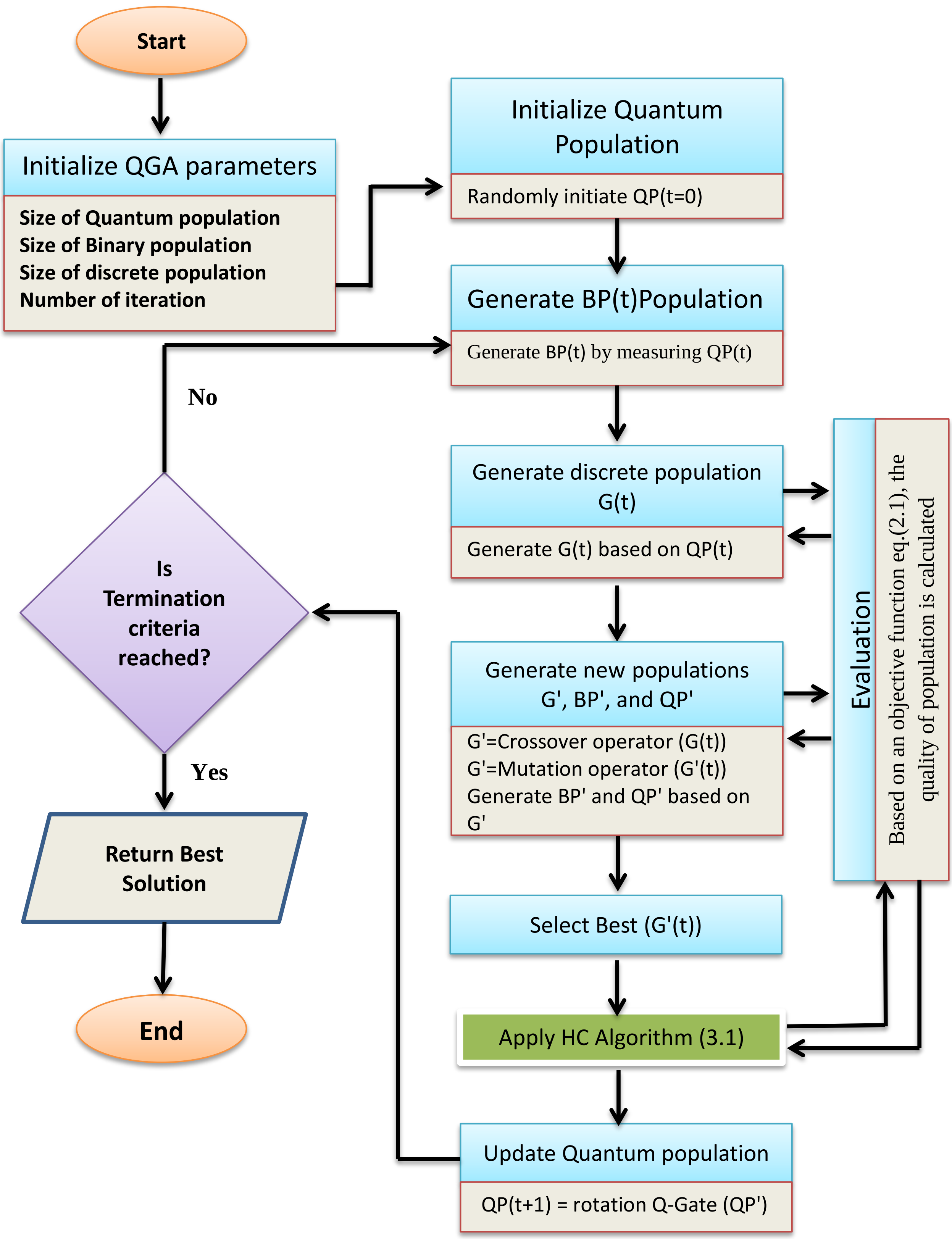

Figure (3.8): HQGA for VRPTW.

3.5 Validation

Obtaining good solutions for a specific NP-hard problem without violating all its constraints is a very complicated task, particularly when dealing with problems of large size and various constraints such as timeframes constraint. Since VRPWT is a constrained problem, the generated solutions are usually infeasible because some customers are either missed or duplicated. In this work, to rectify the infeasible solution, a repair mechanism is applied. Figure (3.9) illustrates the adopted repair mechanism.

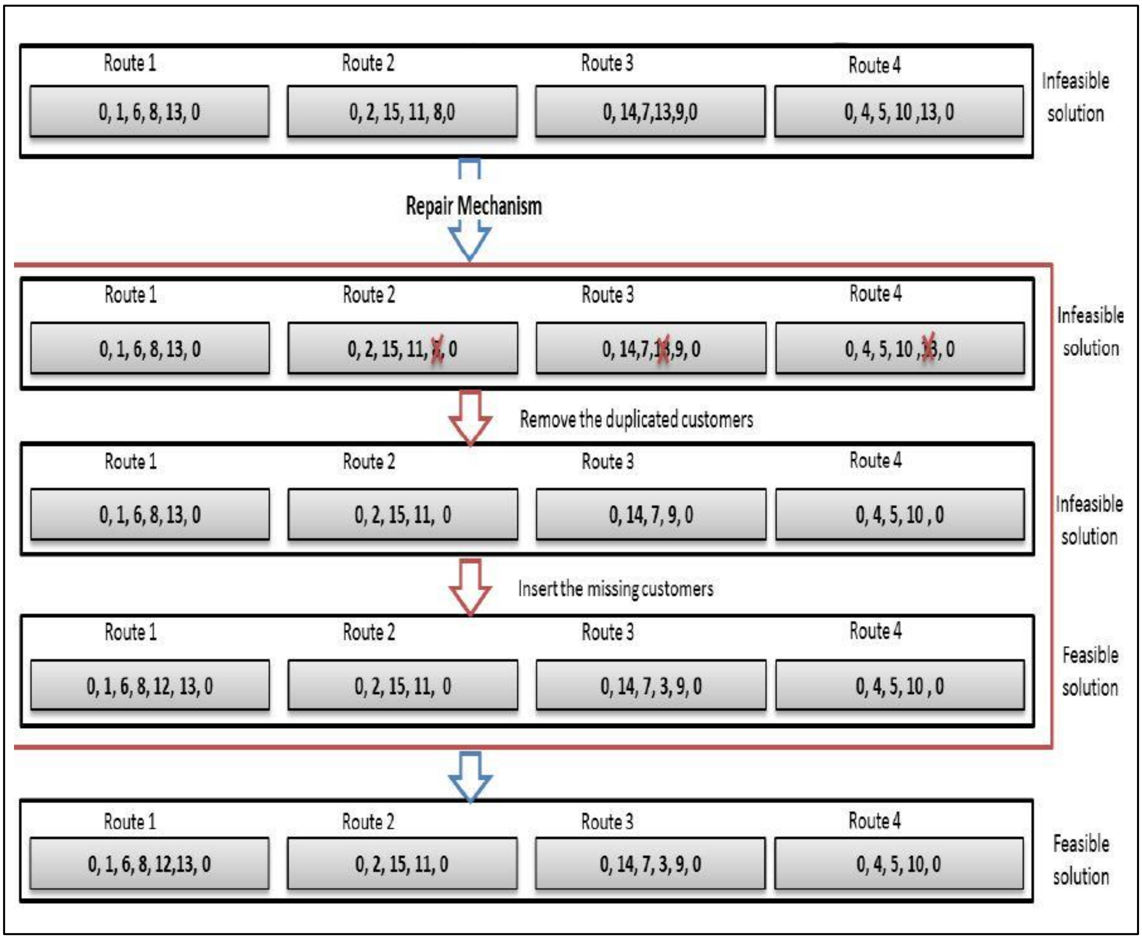

Figure (3.9): Repair Mechanism [76].

In this mechanism, initially, the duplicated and missed customers are determined. Then, all duplicated customers are removed from the routes. Finally, for each missed customer, we try to allocate it to any possible location. If the customer cannot be inserted to any route, a new route will be created for it.

3.6 VRPTW Benchmark

Solomon's VRPTW benchmark is used in this study because it is able to take into consideration different constraints related to vehicle capacity, customers' demands and time windows density. 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. (Figure 3.10) shows the format of each instance in the Solomon's VRPTW benchmark.

RC102

VEHICLE

NUMBER	CAPACITY
25	200

CUSTOMER

CUST NO.	XCOORD.	YCOORD.	DEMAND	READY TIME	DUE DATE	SERVICE TIME
0	40	50	0	0	240	0
1	25	85	20	0	191	10
2	22	75	30	0	199	10
3	22	85	10	0	190	10
4	20	80	40	141	171	10
5	20	85	20	0	189	10
6	18	75	20	95	125	10
7	15	75	20	0	194	10
8	15	80	10	91	121	10
9	10	35	20	91	121	10

Figure (3.10): The Format of the Solomon's VRPTW Instances.

Three types of information are provided on each instance, which are instance name, vehicle specifications and customer specifications. The first row gives the name of the instance and the fourth row gives the number of vehicles and their capacity. Other rows provide detailed information on all customers, where each row describes an individual customer.

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, Equations (2.1). 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. Figure (3.11) and table (3.1) summarizes the characteristics of these types.

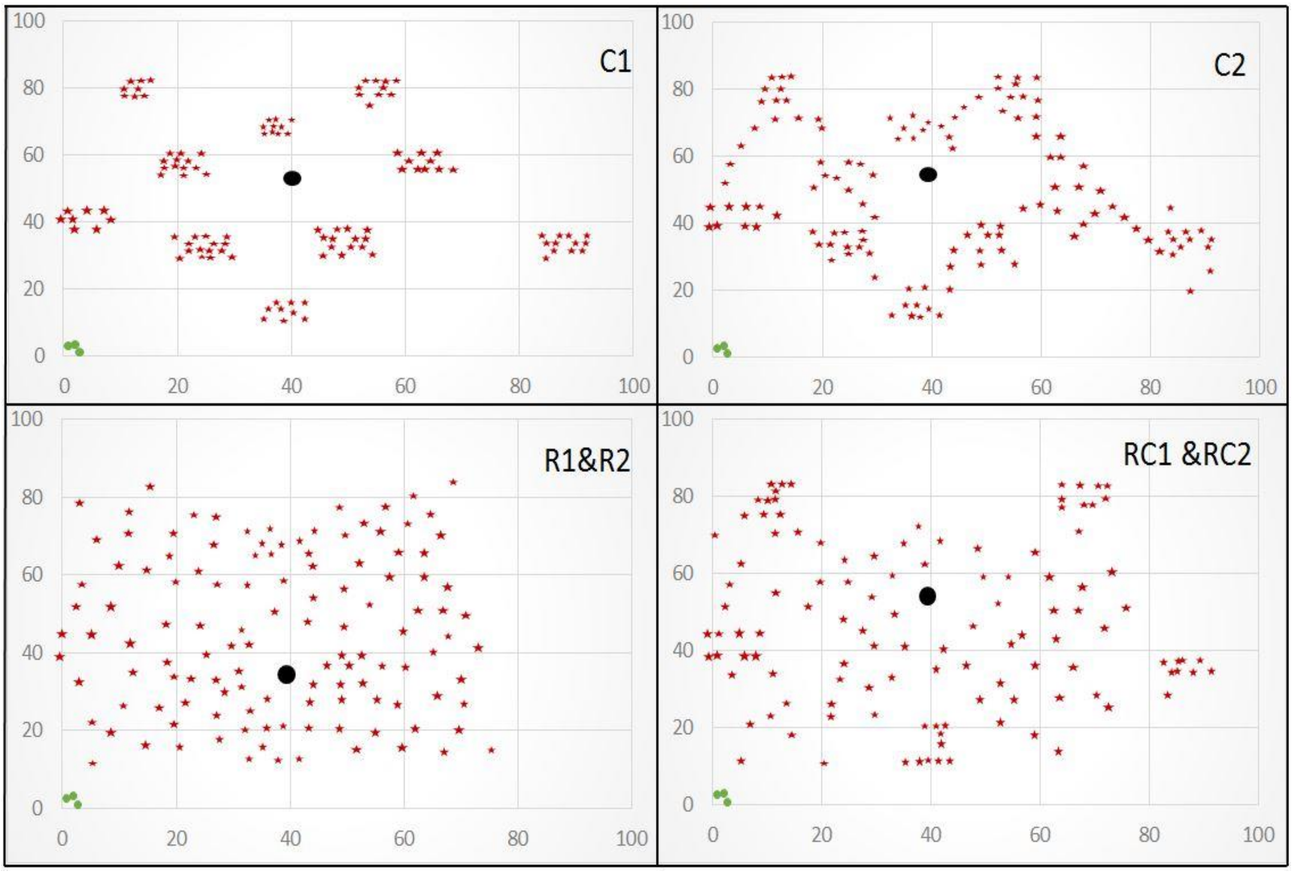

Figure (3.11): The Solomon's VRPTW Benchmark Classes Location with a point black represents the Depot and Stars represents customers.

Table (3.1): 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

Instances with long distances require a great effort to get the best plan to queue customers correctly in each route and eventually decrease the routing cost. With regard to instances with short time windows, the vehicle cannot serve many customers due to the too tight time available. Consequently, there is high possibility of generating infeasible solutions to these instances compared to long time windows instances.

Chapter Four

Experimental Results and Discussion

Chapter Four

4.1 Introduction

The experimental test to verify the effectiveness of proposed algorithms (QGA and HQGA) appears in this chapter. This experimental test is designed to test QGA and HQGA performance and compare them to state-of-the-art techniques. The comparison is conducted in terms of best (Best), average (Avr) and standard deviation (Std) results across the VRPTW. QGA and HQGA have been implemented using Visual studio C#.Net 2010 programming language that is applying under the environment of Microsoft Windows 7 Professional operating system with 64-bits, using PC with processor of 2.80 GHz Core i5 CPU speed and 8 GB RAM

4.2 Parameters 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.

The proposed algorithms have four parameters: GA, QGA and HQGA population size (***n***), solution size (***m***) and maximum number of iterations (***Max_Itr***). HC has only one parameter, the maximum number of iteration (***HC_maxItr***). According to the results of the preliminary testing conducted on different values, these parameters are fixed. Table (4.1) summarizes the parameter settings of all the algorithms used in this study.

Table (4.1): The Parameter Settings of All Algorithms

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

4.3 Experimental Results

To investigate the performance of the proposed QGA for VRPWT, three 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 GA and QGA. To gain sufficient experimental results for these experimental tests, we executed the proposed algorithms for 11 independent runs. Then, the best, average, and standard deviation are reported. In the third experimental test, we compared the performance of HQGA with the results in the literatures. All these algorithms are tested on twelve Solomon VRPTW instances that were introduced by (Solomon 1987). 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.

4.3.1 The Standard QGA Results

This experiment is designed to investigate the effectiveness of the standard QGA in solving the VRPTW. For each tested instance, the best (Best), the average (Avr) and the standard deviation (Std) are shown in table (4.2).

Table (4.2): The QGA Results.

Instance	**QGA**		
	(Avr)	**(Best)**	**(Std)**
R101	2007.679	1944.457	33.87196
R102	1858.869	1812.658	27.84494
R201	1730.709	1674.251	30.59686
R202	1633.393	1574.553	34.66816
C101	1591.165	1504.754	65.50775
C109	1765.409	1573.030	115.95354
C201	1451.143	1282.458	82.62737
C206	1519.675	1305.478	94.63431

RC101	2169.315	2059.855	51.09241
RC102	2002.997	1850.957	58.29527
RC201	2008.146	1851.003	58.29527
RC202	1991.255	1872.72	57.42184

In order to investigate the performance of the QGA, Its results are compared with these of standard GA. Table (4.3) presents the Best, Av and Std for QGA and GA over 11 runs. From Table (4.3), we can observe the following: in terms of the best quality solution, 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 (4.3), QGA is better than GA on Ten tested instances out of 12 instances.

Table (4.3): 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 (4.1) 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 the QGA stuck in local optima. This means that the suggested QGA is efficient in exploration but not in exploitation.

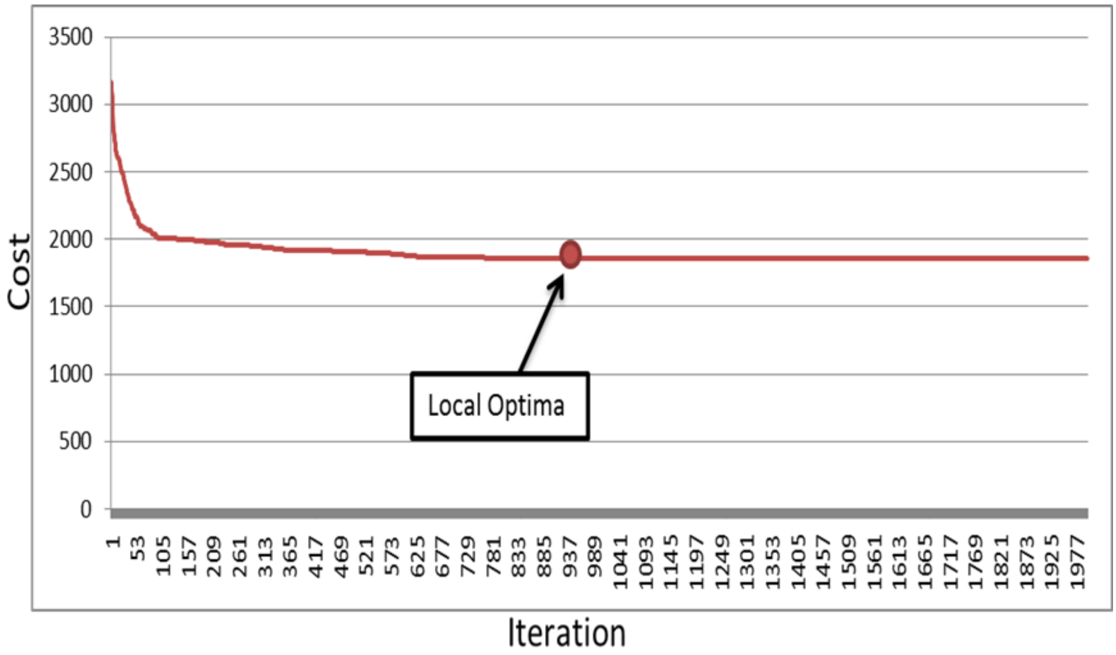

Figure (4.1): The Behavior of QGA during the Search (R201 instance).

4.3.2 The HQGA Results

In order to improve the QGA exploitation process and the quality of generated solution, a hybrid QGA (HQGA) is be proposed in this section. In this hybridization, a single-based meta-heuristic (HC) was integrated with the QGA. This integration made the use of QGA to explore the search space and the HC to exploit the search space.

The results of the basic QGA and the HC are compared with these of the HQGA are reported in tables (4.4), (4.5) and (4.6) in terms of Best, Average, and Std, respectively.

With regard to the Best, table (4.4) shows that HQGA obtained the better results in all and 8 out of 12 instances compared to HC and QGA, respectively.

Table (4.4): Comparison HQGA among Different Heuristics Regarding to the Best.

Instance	HC	QGA	HQGA
	(Best)	(Best)	(Best)
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 (4.5) shows that HQGA obtained the better results in all and 11 out of 12 instances compared to HC and QGA, respectively.

Table (4.5): Comparison HQGA among Different Heuristics Regarding to the Avr.

Instance	HC (Avr)	QGA (Avr)	HQGA (Avr)
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 (4.6) shows that HQGA obtained the better results in all and 8 out of 12 instances compared to HC and QGA, respectively.

Table (4.6): Comparison HQGA among Different Heuristics Regarding to the Std.

Instance	HC	QGA	HQGA
	(Std)	(Std)	(Std)
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**

Finally, Table (4.7) summarizes the obtained results (Best, Avg and Std) of all algorithms.

Table (4.7): obtained results (Best, Avg and Std) of all algorithms.

Instance	GA			QGA			HQGA		
	(Best)	(Avr)	(Std)	(Best)	(Avr)	(Std)	(Best)	(Avr)	(Std)
R101	1891.87	2004.89	66.81681	1944.457	2007.679	33.87196	1943.044	1986.49	25.69852
R102	1827.64	1911.17	71.12597	1812.658	1858.869	27.84494	1658.37	1812.942	62.65536
R201	1657.30	1733.24	53.02217	1674.251	1730.709	30.59686	1587.468	1693.767	54.02967

R202	1495.28	1672.26	144.4957	1574.553	1633.393	34.66816	1539.373	1589.562	32.74791
C101	1544.85	1672.85	110.7533	1504.754	1591.165	65.50775	1527.395	1613.024	44.70781
C109	1524.32	1648.82	85.34133	1573.030	1765.409	115.95354	1498.653	1556.302	32.30996
C201	1336.94	1434.88	58.74583	1282.458	1451.143	82.62737	1289.09	1436.065	68.41679
C206	1234.60	1442.75	104.0266	1305.478	1519.675	94.63431	1257.492	1433.143	98.41659
RC101	2089.71	2208.77	107.6252	2059.855	2169.315	51.09241	2034.47	2141.26	50.97785
RC102	1962.61	2055.55	65.43556	1850.957	2002.997	58.29527	1891.193	1982.964	48.12922
RC201	1885.33	2044.68	78.37532	1851.003	2008.146	58.29527	1852.478	1983.094	68.64602
RC202	1679.71	1908.13	128.6558	1872.72	1991.255	57.42184	1838.094	1942.449	47.21259

4.4 Performance Comparison with Other Algorithms

In this section, the results obtained by the other algorithms in literatures are compared with those of HQGA. Those algorithms are as follows:

- HS: This work verifies the performance of the harmony search algorithm with the VRPTW [29].
- Multi-GA: this work proposed the genetic algorithm with multi parent crossover for solving VRPTW [32].

The best results are displayed from HQGA along with the algorithms compared in Table (4.8) best results are displayed in bold. Table (4.8) illustrate that HQGA obtained the best results in five and eight instances, respectively compared to HS and Multi-GA according to the Best.

Table (4.8): Comparison HQGA among Different Heuristics Regarding to the Best.

Instance	HQGA (Best)	HS (Best)	Multi-GA (Best)
R101	**1943.044**	2061.79	2149.25
R102	**1658.37**	1803.25	2000.92
R201	1587.468	1532.24	1740.88
R202	1539.373	1329.84	1461.81
C101	**1527.395**	1640.07	1646.96
C109	1498.653	1406.84	1993.47
C201	1289.09	581.45	591.56
C206	1257.492	800.13	889.31
RC101	**2034.47**	2154.70	2212.72
RC102	**1891.193**	2033.93	1961.68
RC201	1852.478	1749.43	2039.41
RC202	1838.094	1564.13	1691.68

In terms of Avr, Table (4.9) show that HQGA gets better results in six and ten instances out of 12 instances, respectively Compared to HS and Multi-GA.

Table (4.9): Comparison HQGA among Different Heuristics Regarding to the Avr

Instance	HQGA	HS	Multi-GA
	(Avr)	(Avr)	(Avr)
R101	**1986.49**	2137.62	2304.20
R102	**1812.942**	1962.83	2113.86
R201	1693.767	1584.98	1919.16
R202	1589.562	1460.78	1697.44
C101	**1613.024**	1761.54	1966.10
C109	**1556.302**	1586.32	2217.38
C201	1436.065	780.19	765.64
C206	1433.143	894.26	1407.83
RC101	**2141.26**	2303.16	2367.91
RC102	**1982.964**	2136.89	2169.00
RC201	1983.094	1885.26	2214.62
RC202	1942.449	1734.14	1935.87

Regarding to Std, Table (4.10) show that HQGA gets better results in seven and all instances out of 12, respectively.

Table (4.10): Comparison HQGA among Different Heuristics Regarding to the Std

Instance	HQGA	HS	Multi-GA
	(Std)	**(Std)**	**(Std)**
R101	**25.69**	33.85	94.92
R102	62.65	42.43	63.01
R201	54.02	29.61	98.00
R202	**32.74**	34.30	88.25
C101	**44.70**	62.33	158.77
C109	**32.30**	55.53	121.80
C201	**68.41**	149.47	277.96
C206	98.41	43.35	232.86
RC101	**50.97**	52.36	114.53
RC102	48.12	40.67	120.22
RC201	68.64	52.82	94.02
RC202	**47.21**	49.66	101.26

Provided that the inadequacy of each population-based meta-heuristic exploitation can be compensated via hybridizing it with one of the SBHs that is characterized with its effective ability to exploit the search space, this thesis hypothesized that QGA exploitation ability would be enhanced through combining it with HC. The obtained results have shown that the hybrid QGA outperformed the standard QGA. Therefore, the hypothesis raised above is accepted and proved to be true, this is due to the fact that the hybrid QGA integrates the abilities of HC exploitation and the standard QGA exploration.

Chapter Five

Conclusion and Future Work

Chapter Five

5.1 Conclusions

The present search contributes to the field of transportation and distribution systems via addressing VRPTW aiming to reduce the distance/time required to serve customers. This ends up with more efficient systems that can significantly reduce cost, traffic jam and pollution. This was achieved by enhancing the standard QGA to be able to take into account status of the search during the search process. These enhancements focused on the following aspects:

1) **Carefully design the QGA for VRPTW.** To solve the VRPTW which is highly constrained problem using QGA, the QGA components have to be designed carefully. The main components that are different from one problem to others are discrete population G(t) initialization and crossover operator.
2) **Rectify the infeasible solution**. Since VRPWT is a constrained problem, solutions generated by the improvisation process are usually infeasible because some customers are either missed or duplicated. To rectify the infeasible solution, a repair mechanism is applied.
3) **Consolidate the performance of QGA (exploitation ability) in solving VRPTW.** 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, it combines with one single-based meta-heuristic (HC) that is characterized with its effective ability to exploit the search space. In this combination, 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.

5.2 Future Works

Some ideas for future extensions of the work developed in this thesis are listed:

1) Implementing other different algorithms, such as PSO and ACO with principles of quantum computing.
2) Implementing the proposed algorithm with other combinatorial optimization problems, such as scheduling problem.
3) Implementing the proposed algorithm with another single-based meta-heuristic such as Simulated annealing (SA), Tabu search algorithm.

Reference

References

[1]. Michalewicz, Zbigniew, Fogel, David.B. "How to Solve It: Modern Heuristics", Springer-Verlag Berlin Heidelberg , 978-3-662-07807-5 (2004).

[2]. Talbi, E. G. " Metaheuristics from design to implementation ",Ed.: Wiley Online Library (2009).

[3]. Golden, B. L. & E. A. Wasil " OR Practice—Computerized Vehicle Routing in the Soft Drink Industry", *Operations research* 35(1): 6-17 (1987).

[4]. Eurostat " Panorama of Transport", *Publications Office of the European Union* (2009).

[5]. Roberti, R." Exact algorithms for different classes of vehicle routing problems", (2012).

[6]. Communities, S. O. o. t. E. " Energy, transport and environment indicators" Ed.: Publications Office of the European Union, (2011).

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

[8]. Bräysy, O. & M. Gendreau. "Tabu search heuristics for the vehicle routing problem with time windows", *Top* 10(2): 211-237 (2002).

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

[10]. Stewart, W. & B. Golden. "A chance-constrained approach to the stochastic vehicle routing problem", *1980 Northeast AIDS Conference*. 33-35 (1980).

[11]. Archetti, C., M. W. P. Savelsbergh & M. G. Speranza. "Worst-case analysis for split delivery vehicle routing problems", *Transportation science*: 226-234 (2006).

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

[13]. Yang, C., Z.-x. Guo & L.-y. Liu. "Comparison Study on Algorithms for Vehicle Routing Problem With Time Windows", *Proceedings of the 21st*

International Conference on Industrial Engineering and Engineering Management 2014, Springer: 257-260 (2015).

[14]. O. Braysy, and M. Gendreau . "Vehicle Routing Problem with Time Windows, Part 1: Route Construction and Local Search Algorithms", SINTEF Applied Mathematics Report, Department of Optimization, Norway (2005).

[15]. O. Braysy and M. Gendreau. "Vehicle Routing Problem with Time Windows, Part II: Metaheuristics", SINTEF Applied Mathematics Report, Department of Optimization, Norway (2005).

[16] Desrochers, M., J. K. LENSTRA, M.W. P. SAVELSBERGH AND F. SOUMIS. "Vehicle Routing With Time Windows: Optimization and Approximation. In Vehicle Routing: Methods and Studies, B. L. Golden and A. A. Assad (eds.). North-Holland, Amsterdam, 65-84 (1988).

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

[18] Lahoz-Beltra, R. "Quantum genetic algorithms for computer scientists", Computers, 5, 24 (2016).

[19] Kohl, N., J. Desrosiers, O. B. Madsen, M. M. Solomon & F. Soumis. "2-path cuts for the vehicle routing problem with time windows", *Transportation Science* 33(1): 101-116 (1999).

[20]. Gong, Y. J., J. Zhang, O. Liu, R. Z. Huang, H. S. H. Chung & Y. H. Shi. "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 (2012).

[21]. Burke, E. K., M. Hyde, G. Kendall, G. Ochoa, E. Ozcan & R. Qu. "Hyper-heuristics: A survey of the state of the art", *Journal of the Operational Research Society*, 1(1): 1-40 (2013).

[22]. Burke, E. K. & G. Kendall.. "Search methodologies: introductory tutorials in optimization and decision support techniques" Ed.: Springer (2005).

[23] Thomas A. FeoMauricio G. C. Resende. "Greedy Randomized Adaptive Search Procedures", Journal of Global Optimization (2003).

[24] B.Ombuki, B.J.Ross, and F. Hanshar. "Multi-objective genetic algorithm for vehicle routing problem with time windows", Applied Intelligence, 24:17—30, (2006).

[25] Yaw Chang, Lin Chen. "Solve the vehicle routing problem with time windows via a genetic algorithm", *Conference Publications*, (Special): 240-249 doi, (2007).

[26] H. Nazif and L.S. Lee. "Optimized Crossover Genetic Algorithm for Vehicle Routing Problem with Time Windows", American Journal of Applied Sciences 7 (1): pg. 95-101. (2010).

[27] Filip Taner, Ante Galić, Tonči Carić, "Solving Practical Vehicle Routing Problem with Time Windows Using Metaheuristic Algorithms" , Prompt - Traffic & Transportation: Vol 24 No 4 (2012).

[28] Kalina P., Vokřínek J., Mařík V. "An Efficient Route Minimization Algorithm for the Vehicle Routing Problem with Time Windows Based on Agent Negotiation", IEEE International Conference on Systems (2012).

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

[30] Vaira, G. & Kurasova, O. "Genetic algorithm for VRP with constraints based on feasible insertion", Informatica, 25, 155‑184 (2014).

[31] Koch, H., Henke, T. & Wascher, G. "WORKING PAPER SERIES A Gentic Algorithm for the Multi-Compartment Vehicle Routing Problem with Flexible Compartment Sizes", (2016).

[32] Yassen, E. Taha. "Effectiveness of Selection Mechanisms on the efficiency of Multi Parent Crossover Operator", Journal of university of Anbar for Pure science, 2016, 10 (35-46), (2016).

[33] P.Benioff, "The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines," J. Statist. Phys., vol. 22, pp. 563–591 (1980).

[34] D. Deutsch, "Quantum theory, the Church–Turing principle and the universal quantum computer," *Proc. Roy. Soc. London*,vol. A 400, pp. 97–117 (1985).

[35] K.H. Han, J.H. Kim, "Quantum inspired evolutionary algorithm for a class of combinatorial optimization", IEEE Transactions on Evolutionary Computation 6 (6) 580–593 (2002).

[36] Shor P, "Algorithms for Quantum Computation: Discrete Logarithms and Factoring," *in Proceedings of the 35th Annual Symposium on the Foundation of Computer Sciences*, NM, pp. 20-22 (1994).

[37] Grover L., “A Fast Quantum Mechanical Algorithm for Database Search,” in Proceedings of 28th Annual ACM Symposium on the Theory of Computing, USA, pp. 212-221 (1996).

[38] Ying, M. "Quantum computation, quantum theory and AI. Artif. Intell", , 174, 162–176 (2010).

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

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

[41] Kuk hyun Han, Kui hong Park, Chi ho Lee, and Jong hwan Kim. "Parallel quantum-inspired genetic algorithm for combinatorial optimization problem", In in Proc. 2001 Congress on Evolutionary Computation. Piscataway, NJ: IEEE, pages 1422–1429. Press, (2001).

[42] Zhenquan Zhuang Junan Yang, Bin Li. "Research of quantum genetic algorithm and its application in blind source separation", Journal of Electronics, 20(1):62–68, (2003).

[43] Gexiang Zhang, Weidong Jin, and Laizhao Hu. "A novel parallel quantum genetic algorithm. In Parallel and Distributed Computing", Applications and Technologies, PDCAT’2003. Proceedings of the Fourth International Conference on, pages 693–697. IEEE, (2003).

[44] Gexiang Zhang, Weidong Jin, and Na Li. "An improved quantum genetic algorithm and its application", In Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, pages 449–452. Springer, (2003).

[45] Li Ying Jiao Licheng. "An effective method of image edge detection based on parallel quantum evolutionary algorithm" ,Signal Processing, 1:016, (2003).

[46] Vidal, T., T. G. Crainic, M. Gendreau & C. Prins. "Time-Window Relaxations in Vehicle Routing Heuristics", Ed.: Tech. rep., CIRRELT, Montréal (2013).

[47] Cordeau, J.-F. & M. Maischberger. "A parallel iterated tabu search heuristic for vehicle routing problems", *Computers & Operations Research* 39(9): 2033-2050 (2012).

[48] Takes, F. W. & W. A. Kosters ". Applying Monte Carlo Techniques to the Capacitated Vehicle Routing Problem", in *22th Benelux Conference On Artificial Intelligence (BNAIC)*, (2010).

[49] Francis, P. M., K. R. Smilowitz & M. Tzur. "The period vehicle routing problem and its extensions. *The vehicle routing problem: latest advances and new challenges"*, Springer: 73-102 (2008).

[50] Toth, P. & D. Vigo. "An exact algorithm for the vehicle routing problem with backhauls", *Transportation Science* 31(4): 372-385 (1997).

[51] Vidal, T., T. G. Crainic, M. Gendreau & C. Prins. "Heuristics for multi-attribute vehicle routing problems: a survey and synthesis", *European Journal of Operational Research* 231(1): 1-21 (2013).

[52] Gendreau, M., G. Laporte & R. Séguin. "Stochastic vehicle routing", *European Journal of Operational Research* 88(1): 3-12 (1996).

[53] Malandraki, C. & M. S. Daskin. "Time dependent vehicle routing problems: Formulations, properties and heuristic algorithms", *Transportation Science* 26(3): 185-200 (1992).

[54]. Garcia-Najera, A. & J. A. Bullinaria. "An improved multi-objective evolutionary algorithm for the vehicle routing problem with time windows", *Computers & Operations Research* 38(1): 287-300 (2011).

[54] S. Y. Yang, F. Liu, and L. C. Jiao, “Novel genetic algorithm based on the quantum chromosome,” *Journal of XidianUniversity*, vol. 31, no. 1, pp. 76–81, (2004).

[55] T. Hey, “Quantum computing: An introduction,” in *Computing & Control Engineering Journal*. Piscataway, NJ: IEEE Press, vol. 10, no. 3, pp. 105–112 (1999).

[56] Zhifeng, Z.; Hongjian, Q. "A New Real-Coded Quantum Evolutionary Algorithm", In Proceedings of the 8th WSEAS International Conference on Applied Computer and Applied Computational Science, Hangzhou, China, pp. 426–429 (2009).

[57] Wang, H.; Li, L.; Liu, J.; Wang, Y.; Fu, C. "Improved quantum genetic algorithm in application of scheduling engineering personnel", Abstr. Appl. Anal, 1–10 (2014).

[58] C. J. Rieser, "Biologically Inspired Cognitive Radio Engine Model Utilizing Distributed Genetic Algorithms for Secure and Robust Wireless

Communications and Networking", Virginia Tech, Blacksburg, Va, USA, (2004).

[59] Holland, J. "Adaptation in Natural and Artificial Systems: An Introductory Analysis with applications to biology, Control and Artificial Intelligence", The University of Michigan Press (1975).

[60] Thangiah, S. R., K. E. Nygard & P. L. Juell. Gideon." A genetic algorithm system for vehicle routing with time windows", *Artificial Intelligence Applications, 1991. Proceedings., Seventh IEEE Conference on*, IEEE. 322-328 (1991).

[61] Reid, D. J. "Feasibility and Genetic Algorithms: the Behaviour of Crossover and Mutation", DSTO Electronics and Surveillance Research Laboratory (2000).

[62] Zapfel, G., R. Braune & M. Bogl. "Metaheuristic search concepts", Ed.: Springer (2010).

[63] Baker, J. E. "Reducing bias and inefficiency in the selection algorithm", *Proceedings of the second international conference on genetic algorithms,* 14-21 (1987).

[64] Goldberg, D. E. "*Design of Innovation: Lessons From and For Competent Genetic Algorithm"s,* Kluwer, Boston, MA (2002).

[65] Laboudi, Z.; Chikhi, S. "Comparison of genetic algorithm and quantum genetic algorithm", Int. Arab J. Inf. Technol, 9, 243–249 (2012).

[66] F. Aminifar, C. Lucas, A. Khodaei, and M. Fotuhi Firuzabad, "Optimal placement of phasormeasurement units using immunity genetic algorithm," *IEEE Transactions on Power Delivery,* vol. 24, no. 3, pp. 1014–1020,(2009).

[67] J. C. Hernandez and P. Isasi, "Finding efficient distin-guishers for cryptographic mappings, with an application to the block cipher TEA," Proceedings of the Congress on Evolutionary Computation, pp. 341–348, IEEE Press, (2003).

[68] A. Garrett, J. Hamilton, and G. Dozier, "Genetic algorithm techniques for the cryptanalysis of TEA," International Journal on Intelligent Control and Systems Special Session on Information Assurance. Vol. 12, pp. 325–330, (2007).

[69] Draa A., Meshoul S., Talbi H., and Batouche M.,"A Quantum-Inspired Differential Evolution Algorithm for Solving the N-Queens Problem," *The*

International Arab Journal of Information Technology, vol. 7, no. 1, pp. 21-27, (2010).

[70] Layeb A. and Saidouni D., “Quantum Genetic Algorithm for Binary Decision Diagram Ordering Problem,” *International Journal of Computer Science and Network Security*, vol. 7 no. 9, pp. 130-135, (2007).

[71] Alvarenga, G. B., G. R. Mateus & G. De Tomi. "A genetic and set partitioning two-phase approach for the vehicle routing problem with time windows", *Computers & Operations Research* 34(6): 1561-1584 (2007).

[72] Muter, İ., S. I. Birbil & G. Sahin. "Combination of metaheuristic and exact algorithms for solving set covering-type optimization problems", *INFORMS Journal on computing* 22(4): 603-619 (2010).

[73] James C. Bean, "Genetic Algorithms and Random Keys for Sequencing and Optimization", ORSA Journal on Computing 6(2):154-160 (1994).

[74] Talbi, E. G. "Metaheuristics from design to implementation", Ed.: Wiley Online Library (2009).

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

[76] Koch, P., S. Bagheri, W. Konen, C. Foussette, P. Krause & T. Bäck. "A New Repair Method for Constrained Optimization", *Proceedings of the 2015 on Genetic and Evolutionary Computation Conference*, ACM. 273-280 (2015).

الخلاصة

اعتبرت مشكلة توجيه مسار المركبات مع الوقت (VRPTW) الاكثر شويعاً وانتشاراً على نطاق واسع وذلك لأحتوائها على عامل الوقت الذي يمثل عامل اساسي في حياتنا الواقعية .تبحث مشكلة توجيه مسار المركبات في العثور على أقل مسافة لعدد من المسارات لتسليم البضائع باستخدام مجموعة من المركبات ذات السعة المحدودة وفترة زمنية محددة لخدمة كل عميل. يجب تصميم المسارات بحيث يتم زيارة كل زبون مرة واحدة بواسطة مركبة واحدة فقط خلال فترة زمنية معينة ، كل المسارات تبدأ من مستودع واحد وتنتهي بنفس المستودع. ويجب أن لا تتجاوز جميع طلبات الزبائن في المسار الواحد سعة المركبة. يجب أن تبدأ خدمة الزبائن ضمن الفترة الزمنية المحددة لكل زبون .

تهدف هذه الرسالة إلى التحقق من أداء الخوارزمية الجينية الكمومية (QGA) وتعزيز قدرتها على معالجة (VRPTW) من خلال إجراء العديد من التعديلات. وتهتم هذه التعديلات بتصاميم (QGA) و الخوارزمية المهجنة (HQGA) .QGA هي خوارزمية وراثية حديثة تم تطويرها واقتراحها في السنوات القليلة الماضية ،الناتجة من مزيج الحساب الكمومي والخوارزميات الجينية . تظهر النتائج التي تم الحصول عليها أثناء البحث ، أن سلوك QGA في الفترات الأولى من عملية البحث ، نجح في معالجة VRPTW من خلال تعزيز جودة الحل. ومع ذلك ، فإن قدرة QGA على تحسين جودة الحل تنخفض تدريجياً . وهذا يعني أن QGA تقع في (local optima) . غالباً ما تحدث هذه المشكلة بسبب فعالية QGA في الاستكشاف ولكن ليس في الاستغلال. من أجل تحسين عملية استغلال QGA ونوعية الحل المتولد ، يتم اقتراح خوارزمية مهجنة (HQGA) . في هذا التهجين يتم دمج خوارزمية تسلق الجبل Hill-climbing (HC) مع خوارزمية (QGA) . يمكن QGA هذا التكامل من استكشاف مساحة البحث و HC لاستغلال مساحة البحث .

أظهرت نتائج مقاييس الأداء أن الخوارزمية المقترحة HQGA لها القدرة على إيجاد حلول منافسة مقارنة بالخوارزميات الأخرى. يرجع إلى حقيقة أن الخوارزمية المهجنة HQGA تدمج قدرات استغلال HC واستكشاف QGA.

جمهورية العراق

وزارة التعليم العالي والبحث العلمي

جامعة الانبار

كلية علوم الحاسوب وتكنولوجيا المعلومات

إستخدام خوارزمية هجينة لحل مشكلة توجيه المركبات مع الإطار الزمني إعتماداً على الحوسبة الكمية التطويرية

رسالة مقدمة الى

قسم علوم الحاسبات – كلية علوم الحاسبات وتكنولوجيا المعلومات –جامعة الانبار وهي جزء من متطلبات نيل درجة الماجستير في علوم الحاسبات

قدمت من قبل

أحمد مهند فياض

بـإشراف

أ.د. بلال أسماعيل الخطيب

أ.م.د.عصام طه ياسين

١٤٤٠هـ ٢٠١٩ م